DeBERTa-v3 with R-Drop regularization for Multi-Author Writing Style Analysis

Notebook for the PAN Lab at CLEF 2024

Zhijian Huang[1], Leilei Kong[1,†]

[1]Foshan University, Foshan, Guangdong, China

Abstract

The Multi-Author Writing Style Analysis task aims to identify points within a multi-author document where the author changes, using variations in writing style as indicators. Existing approaches face challenges in achieving high robustness due to the complexity of distinguishing between different authors' styles. To address these challenges, we use a model based on base version of the DeBERTa-v3 model combined with R-Drop regularization. We trained the DeBERTa-v3 model independently on three different datasets representing varying difficulty levels, using R-Drop during training to enhance the model's performance by reducing uncertainty and improving generalization. In experiments, our method achieves F1 scores of 0.985, 0.815, and 0.826 on Task 1, Task 2, and Task 3 of the official test set for the PAN 2024 Multi-Author Writing Style Analysis, respectively.

Keywords

Multi-Author Writing Style Analysis, DeBERTa-v3, R-Drop

1. Introduction

Multi-Author Writing Style Analysis aims to identify points within a multi-author document where authorship changes occur. This task is based on the hypothesis that variations in writing style can serve as indicators of changes in authorship. PAN's evaluation focuses on distinguishing authorship changes at the paragraph level under varying conditions of topical similarity [1, 2].

Various methods have been proposed to tackle this task, ranging from traditional machine learning algorithms to advanced deep learning models. Earlier approaches predominantly relied on features extracted from the text, such as lexical and syntactic markers, to differentiate between authors. However, these methods often fall short in scenarios where stylistic differences are minute. More recent approaches employ pre-trained language models [3] like BERT [4] and its variants, which have shown promise in capturing deeper contextual information.

To further enhance the robustness of pre-trained language model approaches, we use the advanced pre-trained language model DeBERTa(Decoding-enhanced BERT with Disentangled Attention)-v3 [5] and combine it with the R-Drop regularization [6]. The DeBERTa-v3 model, known for its ability to capture complex language structures and patterns through its decoding-enhanced attention mechanism, serves as the foundation. By incorporating R-Drop, which introduces dual regularization during training, we enhance the model's performance by reducing uncertainty and improving generalization, thereby effectively mitigating overfitting and maintaining model stability.

2. Related work

Pre-trained language models have revolutionized the field of natural language processing (NLP), demonstrating significant improvements across various tasks, including multi-author writing style analysis. Models such as BERT [4], RoBERTa [7] and DeBERTa [5] have set new benchmarks by

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

† corresponding author

huangzhijian1024@163.com (Z. Huang); kongleilei@fosu.edu.cn (L. Kong)

0009-0002-5049-2093 (Z. Huang); 0000-0002-4636-3507 (L. Kong)

leveraging large-scale unsupervised pre-training followed by fine-tuning on specific tasks. These models capture contextual information bidirectionally, making them highly effective for tasks requiring nuanced understanding of both writing styles and semantic content.

In multi-author writing style analysis, pre-trained language models are utilized to encode textual features that are indicative of different authors' styles. For instance, Chen et al. [8] demonstrated the use of a pre-trained language model for generating sentence embeddings optimized through contrastive learning for detecting writing style changes in multi-author documents. Similarly, Huang et al. [9] proposed an encoded classifier using knowledge distillation, leveraging a large pre-trained model as the teacher to train a smaller student model for style change detection.

Regularization techniques are critical in enhancing the generalization capabilities of neural networks by preventing overfitting, especially on small datasets. Traditional methods such as L2 regularization, dropout, and early stopping have been widely used to improve model robustness. Dropout, in particular, randomly sets a fraction of the input units to zero during training, which helps in preventing co-adaptation of hidden units.

Building on the concept of dropout, R-Drop (Regularized Dropout) is a more recent technique proposed by Liang et al. [6]. R-Drop enhances regularization by applying dropout twice during training and minimizing the divergence between the two forward passes. This approach encourages the model to produce consistent outputs despite the dropout noise, thereby learning more robust representations.

In the context of multi-author writing style analysis, R-Drop can be particularly beneficial when combined with pre-trained language models. The regularization helps mitigate overfitting on training datasets, which is often a challenge in style change detection tasks. By ensuring that the model maintains consistent outputs despite the dropout noise, R-Drop enhances the robustness of the learned writing style representations.

3. Methods

In the Multi-Author Writing Style Analysis task, our goal is to identify points within a document where the author changes, using variations in writing style as indicators. We use a model based on the base version of the DeBERTa-v3 model combined with R-Drop regularization. We trained the base version of the DeBERTa-v3 model independently on three different datasets representing varying difficulty levels, using R-Drop during training to enhance the model's performance by reducing uncertainty and improving generalization.

3.1. Encoder and Classifier

We used the base version of the DeBERTa-v3 model as the encoder, which excels in capturing language structures and patterns. DeBERTa utilizes enhanced decoding and disentangled attention mechanisms to better understand contextual information. On top of DeBERTa-v3, we added a binary classification layer to detect style changes between paragraphs. This layer is trained using a loss function that combines cross-entropy loss and KL divergence for enhanced regularization, as described in the R-Drop regularization.

3.2. R-Drop regularization

The R-Drop method (Regularized Dropout) aims to reduce model uncertainty by introducing dual regularization during training. Specifically, for each training batch, we perform two forward and backward passes and calculate the KL divergence between the two forward pass results as a regularization term. The core formula is as follows:

$$L_{\text{total}} = L_{\text{ce}} + \alpha L_{\text{kl}}$$

where L_{ce} is the cross-entropy loss, L_{kl} is the KL divergence between the two forward pass results, and α is the weighting parameter.

Given the input data x_i at each training step, we feed x_i through the forward pass of the network twice, obtaining two distributions of the model predictions, denoted as $P_{w1}(y_i|x_i)$ and $P_{w2}(y_i|x_i)$. Since the dropout operator randomly drops units in a model, the two forward passes are based on two different sub models. The KL divergence between these two output distributions is then calculated as follows:

$$L_{\text{kl}} = \frac{1}{2}\left(D_{\text{KL}}(P_{w1}(y_i|x_i)\|P_{w2}(y_i|x_i)) + D_{\text{KL}}(P_{w2}(y_i|x_i)\|P_{w1}(y_i|x_i))\right)$$

where D_{KL} denotes the Kullback-Leibler divergence.

3.3. DeBERTa-v3 with R-Drop regularization

As Algorithm 1 demonstrates, the training process for DeBERTa-v3 with R-Drop regularization includes data input, model training, and model parameters output, ensuring a comprehensive understanding of the method. We start by loading and preprocessing the data, splitting it into training, validation, and test sets. The preprocessed paragraph pairs are then input into the model. We fine-tune the DeBERTa-v3-base model on the training data using the R-Drop method. This method involves performing two forward and backward passes for each training batch and calculating the KL divergence between the two forward pass results as a regularization term, helping to reduce model uncertainty and improve robustness.

During training, for each batch, we compute the forward pass twice with dropout, calculate the cross-entropy and KL divergence losses, and update the model parameters. Early stopping [10] is implemented based on the evaluation set to prevent overfitting, monitoring the validation loss and halting training when it ceases to improve. We assess the model's performance on the validation set using the F1-score to measure its effectiveness. Finally, we evaluate the final model on the test set to measure its overall performance.

Algorithm 1 Training Process for DeBERTa-v3 with R-Drop regularization

Input: Number of training epochs $epochs$, Training data loader $train\_loader$, Validation data loader val_loader, Loss function $loss\_fct$, Weighting parameter α, Optimizer $optimizer$, Evaluation step $eval\_step$
Output: Trained model parameters θ
Initialize model parameters θ with initial values
for epoch e in range $epochs$ **do**
    for each batch (x, y) in $train\_loader$ **do**
        Set model to training mode
        Compute model output $output1$ for the current batch x
        Compute model output $output2$ for the current batch x
        Calculate the cross-entropy loss $L_{ce} = 0.5 \cdot (loss\_fct(output1, y) + loss\_fct(output2, y))$
        Compute the KL divergence loss $L_{kl} = compute\_kl\_loss(output1, output2)$
        Combine the losses: $L_{total} = L_{ce} + \alpha \cdot L_{kl}$
        Perform backpropagation and update model parameters using $optimizer$
        if current step % $eval\_step == 0$ **then**
            Evaluate the model on validation data
        end if
    end for
end for
Return: Trained model parameters θ

4. Experiments

4.1. Datasets

The datasets were provided by the PAN (Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection) initiative as part of the PAN 2024 lab at CLEF (Conference and Labs of the Evaluation Forum). The datasets used for this task are derived from Reddit comments, combined into documents that represent different levels of difficulty. These datasets are designed to test the models' ability to detect style changes under various conditions of topical similarity. Each dataset is split into three subsets: training, validation, and test sets, with respective proportions of 70%, 15%, and 15%. The difficulty levels of the task are as follows:

- **Easy**: This dataset consists of documents where the paragraphs cover a wide variety of topics. The diverse topics make it easier for models to leverage topic changes as signals for detecting authorship changes.
- **Medium**: The documents in this dataset have a limited number of topics, requiring models to focus more on subtle changes in writing style rather than topic shifts to detect changes in authorship.
- **Hard**: All paragraphs in the documents within this dataset are on the same topic. This scenario poses the greatest challenge as the models must rely entirely on stylistic differences to identify authorship changes

4.2. Experimental Setup

For our experiments, our preprocessing of the data involves several key steps. First, the dataset is loaded, and each document is read. The documents are then split into natural paragraphs. Following this, we generate pairs of consecutive paragraphs and label each pair to indicate whether there is a style change between them. This labeling transforms the task into a binary classification problem. Each labeled pair of paragraphs is then used as an input for the training of our model. These steps result in the creation of features and labels for the training, validation, and test sets, which are used for model training and evaluation.

We utilized the DeBERTa-v3-base(the base version of the DeBERTa-v3 mode) and DeBERTa-v3-base+R-Drop models. The DeBERTa-v3-base model served as our baseline, whereas the DeBERTa-v3-base+R-Drop model incorporated the R-Drop regularization technique to enhance performance by reducing variance between different forward passes. For all three datasets, the R-Drop hyperparameters were set as follows: kl_alpha, was set to 5 based on the experimental results from the original paper [6]. The Dropout rate was set to 0.1, which is the default value for DeBERTa-v3-Base.

The DeBERTa-v3-base+R-Drop model was fine-tuned on the training sets using the Adam optimizer [11] with a learning rate of 1×10^{-5}. The batch size was set to 16, and the models were trained for 10 epochs. To prevent overfitting, early stopping was implemented based on the validation loss. Additionally, we included results from two simple baselines: one where the prediction is always 1 and another where the prediction is always 0 [12], and compared these to the outcomes from our models on the test dataset.

Evaluation of the models was conducted using the F1-score on both the validation and test sets. The F1-score was chosen as the primary metric due to its balance between precision and recall, which is crucial for accurately detecting changes in writing style.

4.3. Results

In Table 1, we report the F1 scores on the validation set for the multi-author writing style analysis task. We present the scores for the validation and test sets, comparing the performance of DeBERTa-v3-base and DeBERTa-v3-base+R-Drop.The results show that the DeBERTa-v3-base+R-Drop model generally outperforms the baseline DeBERTa-v3-base model across all difficulty levels on the validation

set. However, this improvement is more noticeable in the Easy category compared to the Medium and Hard categories.

The Easy category shows a significant performance boost, which might be attributed to the R-Drop regularization reducing overfitting and enhancing model stability. On the other hand, the performance gains in the Medium and Hard categories are relatively modest. This could be due to the increased complexity and reduced topical variety in these categories, which pose a greater challenge for the model. The smaller performance gains suggest that while R-Drop helps, it might not fully address the intricacies involved in these more difficult tasks.

Table 1
F1 scores on validation set for multi-author writing style analysis task using DeBERTa-v3 and DeBERTa-v3 + R-Drop. The tasks included Task 1 (easy dataset), Task 2 (medium dataset), and Task 3 (hard dataset).

Approach	Task 1	Task 2	Task 3
DeBERTa-v3-base	96.9	83.8	83.5
DeBERTa-v3-base+R-Drop	98.7	84.1	84.1

In Table 2, we report the F1 scores on the test set for the multi-author writing style analysis task. The DeBERTa-v3-base+R-Drop model maintains its performance, but with noticeable variability in the Medium and Hard categories. This variability indicates that the model's generalization capability, while improved by R-Drop, still faces challenges with more complex and less distinct style changes.

Table 2
F1 scores on test set for multi-author writing style analysis task using DeBERTa-v3 and two sample baselines. The tasks included Task 1 (easy dataset), Task 2 (medium dataset), and Task 3 (hard dataset).

Approach	Task 1	Task 2	Task 3
DEBERTA+R-drop	0.985	0.815	0.826
Baseline Predict 1	0.466	0.343	0.320
Baseline Predict 0	0.112	0.323	0.346

5. Conclusion

Our study shows that combining the base version of the DeBERTa-v3 model with R-Drop regularization significantly improves the accuracy of detecting authorship changes across documents of varying difficulty. We trained the model on datasets with different levels of topic diversity, showing marked improvements particularly in documents with diverse topics. However, in documents with limited topical diversity, performance gains are modest, indicating the need for further refinement. Future work should focus on enhancing model performance in more complex scenarios and exploring complementary methods to address the identified challenges.

Acknowledgments

This work is supported by the National Social Science Foundation of China (22BTQ101)

References

[1] 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.

[2] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: M. A. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), Association for Computational Linguistics, 2018, pp. 2227–2237. URL: https://doi.org/10.18653/v1/n18-1202. doi:10.18653/V1/N18-1202.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[4] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/V1/N19-1423.

[5] P. He, X. Liu, J. Gao, W. Chen, Deberta: decoding-enhanced bert with disentangled attention, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021. URL: https://openreview.net/forum?id=XPZIaotutsD.

[6] X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T. Liu, R-drop: Regularized dropout for neural networks, in: M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, 2021, pp. 10890–10905. URL: https://proceedings.neurips.cc/paper/2021/hash/5a66b9200f29ac3fa0ae244cc2a51b39-Abstract.html.

[7] 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, volume abs/1907.11692, 2019. URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[8] H. Chen, Z. Han, Z. Li, Y. Han, A writing style embedding based on contrastive learning for multi-author writing style analysis, in: M. A. andf Guglielmo Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023, volume 3497 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 2562–2567. URL: https://ceur-ws.org/Vol-3497/paper-206.pdf.

[9] M. Huang, Z. Huang, L. Kong, Encoded classifier using knowledge distillation for multi-author writing style analysis, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023, volume 3497 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 2629–2634. URL: https://ceur-ws.org/Vol-3497/paper-214.pdf.

[10] R. Caruana, S. Lawrence, C. L. Giles, Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, in: T. K. Leen, T. G. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA, MIT Press, 2000, pp. 402–408. URL: https://proceedings.neurips.cc/paper/2000/hash/059fdcd96baeb75112f09fa1dcc740cc-Abstract.html.

[11] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6980.

[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. doi:10.1007/978-3-031-28241-6_20.