Text Author Classification: A ModernBERT Approach with Gradient Loss Function

Notebook for the PAN at CLEF 2025

Zhankeng Liang[1], Kaiyin Sun[2], Haojie Cao[1], Jieren Luo[1] and Zhongyuan Han[1,*]

[1]*Foshan University, Foshan , China*
[2]*Foshan No.3 Middle School, Foshan , China*

Abstract

This paper focuses on the text author (human or AI) classification problem, aiming to enhance the model's ability to detect AI-generated texts. We propose an innovative method based on ModernBERT, which introduces a custom gradient loss function and optimizes model training by combining sample weighting strategies, effectively enhancing the model's classification capability for texts of varying difficulties. In implementation, we fine-tuned the ModernBERT model, optimized the training process using the custom gradient loss function, and constructed an efficient classification system through meticulous data preprocessing and rigorous testing evaluation. Experimental results show that this method achieved good performance metrics on the training set, but there was some overfitting on the test set, with performance metrics declining. Future work will be dedicated to further optimizing the model's generalization ability to improve its potential for application in multi-domain text classification.

Keywords

PAN 2025, Text Author Classification, ModernBERT, Gradient Loss Function, AI Detection

1. Introduction

With the tremendous success of large language models in recent years and the rapid development of AI content generation technology, text author classification (human or AI) has gradually become an important research area. In recent years, the gap between AI-generated texts and human texts in terms of language fluency and logical coherence has gradually narrowed, and in some scenarios, it is even difficult to distinguish between them. This makes text author classification crucial for content authenticity verification, copyright protection, and information security defense. [1] The research work in this paper is conducted for the Voight-Kampff Generative AI Authorship Verification task at the PAN 2025 workshop. [2]

This study aims to develop a classification model that can effectively distinguish whether the text author is human or AI. Based on the pre-trained ModernBERT model, we referred to a training method that combines gradient loss and sample weighting. During training, we not only focused on classification accuracy but also controlled the model's learning intensity for samples of different difficulties through the gradient loss function to prevent overfitting to difficult samples. We also assigned weights to the samples in the dataset, with higher weights for easy samples and lower weights for difficult samples, to adjust the loss calculation. Experiments have shown that this method can improve the model's accuracy on the validation set, verifying its effectiveness in the text author classification task.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

[*] Corresponding author.

oooo1390329007@163.com (Z. Liang); sunkaiyin123@163.com (K. Sun); caohaojie0322@163.com (H. Cao); wolike666@gmail.com (J. Luo); hanzhongyuan@gmail.com (Z. Han)

0009-0009-6124-5494 (Z. Liang); 0009-0001-7966-8390 (K. Sun); 0000-0002-8365-168X (H. Cao); 0009-0007-4867-4214 (J. Luo); 0000-0001-8960-9872 (Z. Han)

2. Related Work

Currently, research in this field mainly focuses on the application of text feature analysis and deep learning models. Traditional methods extract text features such as vocabulary, grammar, and syntax, and use machine learning algorithms such as Support Vector Machine (SVM), Naive Bayes, Logistic Regression, and Decision Tree for classification. Although these methods can distinguish between human and AI-generated texts to some extent, feature engineering is complex and relies on human experience, making it difficult to effectively extract and utilize complex text features. Moreover, with the development of deep learning technology, models based on Recurrent Neural Networks (RNN) and its variants (such as LSTM, GRU) have been applied to text author classification. These models can automatically learn the sequential features and semantic information of texts, resulting in improved classification performance compared to traditional methods. [3] In addition, Convolutional Neural Networks (CNN) have also been applied in this field, mainly for extracting local features and structural information of texts. However, these methods have limited ability to handle long texts and complex semantic relationships and are easily affected by noisy data.

In recent years, pre-trained language models based on the Transformer architecture (such as BERT and its variants, GPT series, etc.) have gradually become the mainstream method for text author classification. [1] These methods first use unsupervised pre-training on large-scale unannotated text data to enable the model to learn rich text features and semantic knowledge. Then, through fine-tuning strategies, they can effectively capture the differences between human and AI-generated texts, achieving higher accuracy in classification tasks. [4]

3. Method Introduction

This experiment is based on the ModernBERT model and employs an efficient text author classification method, aimed at overcoming the limitations of traditional models such as the original BERT in handling long texts, computational efficiency, and timeliness of data. [5]The ModernBERT model, introduced by Warner et al. (2024), provides a robust foundation for text classification tasks with its capability to process sequences up to 8192 tokens, optimized architectural design, and adaptability to new data. [1]

In the experiment, we first preprocessed the training and validation data, including tokenization, truncation or padding to a fixed length, and assigning weights based on sample difficulty. Then, we fine-tuned the ModernBERT model using this data by adding a classification layer that utilizes the CLS token as input to capture the global representation of the text for the binary classification task and optimizing model parameters to adapt to specific text classification tasks. Additionally, a custom gradient loss function was introduced, which effectively controlled the learning intensity of the model for samples of varying difficulties by adjusting the gradient of the loss function for model outputs, preventing overfitting. Ultimately, the model performed well on the training set, verifying its effectiveness and superiority in text author classification tasks.

Figure 1 below shows the basic workflow of the ModernBERT model in the experiments of this paper.

4. Experiment

4.1. Overview of Experimental Requirements

This experiment aims to construct a binary classification model and program that can accurately distinguish whether the text author is human or AI. The experimental dataset is provided by PAN lab 2025 and contains text samples generated by humans or machines in JSONL format. [2] Each sample includes the text content, a unique identifier, the author type (human or a specific AI model), a label (0 for human, 1 for AI), and the text genre (such as prose, news, or fiction). The test set only includes the text and identifier, used to evaluate the model's generalization ability. Evaluation metrics include ROC-AUC, Brier score, C@1, F1 score, F0.5u, the arithmetic mean of these metrics, and the confusion matrix used to calculate true/false positives and true/false negatives.

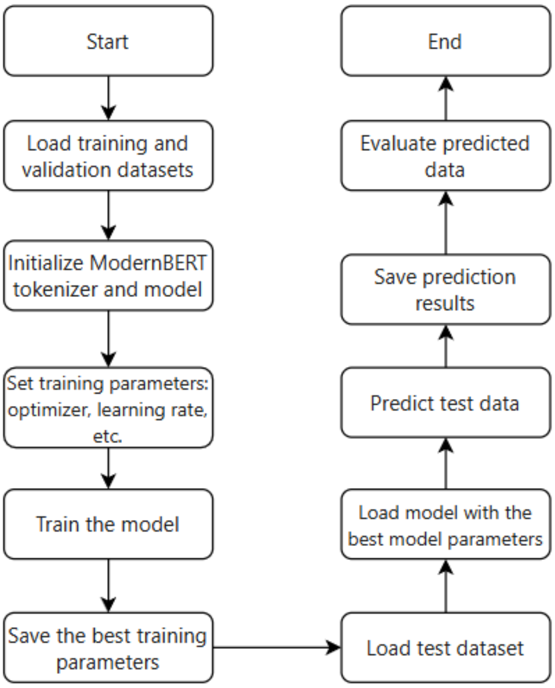
Start
Load training and validation datasets
Initialize ModernBERT tokenizer and model
Set training parameters: optimizer, learning rate, etc.
Train the model
Save the best training parameters
Load test dataset
Load model with the best model parameters
Predict test data
Save prediction results
Evaluate predicted data
End

Figure 1: Overall Experimental Workflow Diagram

4.2. Data Source

The fine-tuning training data for this model comes from the Zenodo platform, uploaded by PAN lab 2025. The dataset includes training and validation datasets, containing text samples generated by humans and multiple AI models to achieve training and prediction purposes. [2] Additionally, some of the test datasets are not publicly available due to competition reasons and are only allowed for online testing by uploading the program.

4.3. Program Design

4.3.1. Model Selection and Initialization

This program uses the pre-trained ModernBERT-base model from answerdotai company as the base model and fine-tunes it (fine-tuning) with the training dataset to adapt to the text author classification task. The ModernBERT model, based on the Transformer architecture, uses a self-attention mechanism to capture long-range dependencies in the text. The output layer of the model is adjusted for binary classification tasks, with an output category number of 2 (human or AI).

4.3.2. Data Preprocessing

The training and validation datasets are provided in JSONL format. The data preprocessing steps are as follows:

1. Tokenize the text using the ModernBERT tokenizer.

2. Truncate or pad the tokenized sequence to a fixed length (512 tokens). Although ModernBERT can handle sequences of up to 8192 tokens, we truncate or pad the text to 512 tokens here to align with the input requirements of the dataset and to ensure uniformity of model input, which can also improve the computational efficiency during training.

3. Assign weights to each sample based on the difficulty label. The difficulty of a sample is determined by its distinguishability from human-written text. Samples that are more easily distinguishable (i.e., those with more obvious AI characteristics) are labeled as easy and assigned higher weights, while samples that are harder to distinguish (i.e., those closely resembling human writing style) are labeled as difficult and assigned lower weights. This strategy allows the model to focus more on difficult samples during training, thereby improving its generalization ability.

4.3.3. Model Training

The model training process includes several key steps:

1. Define a custom dataset class: Create a custom dataset class MyDataset to load and preprocess training and validation data. This class is also responsible for reading JSONL files, tokenizing, truncating, and padding the text, and assigning weights to each sample based on the difficulty label.
2. Custom gradient loss function: A custom gradient loss function is defined in the experimental program, combining the standard cross-entropy loss and its gradient for model outputs, while adjusting the gradient size based on sample weights. The formula for the custom gradient loss function is as follows:

$$\mathcal{L}_{\text{grad}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\nabla_{\theta}\,\text{CE}(\text{logits}_i,\ \text{labels}_i)\cdot \text{weights}_i\right\|_2$$

   where CE is the cross-entropy loss function, logits represents the gradient of the model's output logits, weights are the sample weights, and N is the batch size. This formula calculates the gradient of the cross-entropy loss for the model's output, adjusts the gradient size based on sample weights, and ultimately derives the gradient loss to optimize model training and prevent overfitting.
3. Training process: During the training phase, the preprocessed data is loaded onto the GPU, and the AdamW optimizer is used to zero the model's gradients. In training, the model performs forward propagation to obtain output results, calculates the loss using the custom gradient loss function, and then updates the model parameters through backpropagation and the optimizer. The entire training process lasts for three epochs, and the best model parameters are saved by monitoring the validation set accuracy.

4.3.4. Prediction Data

The prediction process includes the following steps:

1. Define the test dataset class: Load the trained model parameters and define the test dataset class TestDataset to load and preprocess the test data, similar to the custom dataset class MyDataset used in the training model above.
2. Prediction function: The prediction function loads the test data, performs inference through the model, converts the predicted confidence scores into probabilities through the sigmoid function, and then converts them into binary labels based on a threshold of 0.5. Finally, the prediction results are saved in JSONL format.

4.4. Experimental Results and Analysis

4.4.1. Training Results

This experimental program has been uploaded to the official testing platform tria of PAN lab 2025 for official testing [6]. The program has undergone three online tests using both the training and test datasets. In the tests on the training dataset, the average ROC-AUC metric value was 0.874, the average Brier metric value was 0.878, the average C@1 metric value was 0.878, the average F1 metric value was 0.904, the average F0.5u value was 0.913, the average Mean value was 0.889. In the tests on the test dataset, the average ROC-AUC metric value was 0.844, the average Brier metric value was 0.853, the average C@1 metric value was 0.853, the average F1 metric value was 0.815, the average F0.5u value was 0.917, the average Mean value was 0.856.

Tables 4-1 and 4-2 below show the evaluation results of the test and training datasets on the tria online platform, respectively.

Table 1
Evaluation Results on Test Dataset

Method	ROC-AUC	Brier	C@1	F1	F0.5u	Mean
Gradient Loss	0.844	0.853	0.853	0.815	0.917	0.856
ppmd	0.723	0.794	0.731	0.796	0.780	0.765
tiny-llama	0.740	0.673	0.502	0.439	0.658	0.603

Table 2
Evaluation Results on Training Dataset

Method	ROC-AUC	Brier	C@1	F1	F0.5u	Mean
Gradient Loss	0.874	0.878	0.878	0.904	0.913	0.889
ppmd	0.786	0.799	0.757	0.812	0.778	0.786
tiny-llama	0.821	0.751	0.627	0.585	0.773	0.711

4.4.2. Results Analysis

Based on multiple tests of the test and training datasets, overall, the program using the pre-trained ModernBERT model and fine-tuning it to adapt to the text author classification task has achieved good results. However, there is still room for improvement in the model's generalization ability. The high metric values on the training set indicate that the model fits the training data well, but the performance drop on the test set suggests insufficient adaptability to new data. In the future, methods such as data augmentation, regularization, or adjusting the model architecture could be considered to enhance the model's generalization ability to cope with the diversity of test data.

5. Conclusion

This thesis innovatively proposes a text author classification method based on ModernBERT, conducting in-depth research on the detection of human and AI texts. For the first time, a gradi-ent loss function is applied to the fine-tuning process of ModernBERT, combined with a sample weighting strategy to optimize model training, effectively improving the classification perfor-mance for texts of different difficulties. In implementation, through meticulous data prepro-cessing, custom gradient loss calculation, model fine-tuning, and rigorous testing evaluation, an efficient classification system is constructed. Experiments have verified the advantages of this method over traditional models, providing new ideas for the identification of AI-generated con-tent and potentially expanding its application in multi-domain text classification in the future.

Acknowledgments

This work is supported by the Social Science Foundation of Guangdong Province, China (No.GD24CZY02)

Declaration on Generative AI

During the preparation of this work, the author utilized Kimi K2 and DeepSeek-V3 to accomplish drafting content, text translation, and content enhancement tasks. After employing these tools, the author reviewed and edited the content as necessary and took appropriate measures to assume full responsibility for the content of the publication.

References

[1] T. Wu, Y. Wang, N. Quach, Advancements in natural language processing: Exploring transformer-based architectures for text understanding, arXiv preprint arXiv:2503.20227 (2025).

[2] J. Bevendorff, Y. Wang, J. Karlgren, M. Wiegmann, M. Fröbe, A. Tsivgun, J. Su, Z. Xie, M. Abassy, J. Mansurov, R. Xing, M. Ta, K. Elozeiri, T. Gu, R. Vardhan Tomar, J. Geng, E. Artemova, A. Shelmanov, N. Habash, E. Stamatatos, I. Gurevych, P. Nakov, M. Potthast, B. Stein, Overview of the "Voight-Kampff" generative AI authorship verification task at PAN and ELOQUENT 2025, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR-WS.org, Madrid, Spain, 2025.

[3] B. Mahesh, Machine learning algorithms—a review, International Journal of Science and Research (IJSR) 9 (2020) 381–386.

[4] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, S. Feizi, Can AI-generated text be reliably detected?, arXiv preprint arXiv:2303.11156 (2023).

[5] B. Warner, A. Chaffin, B. Clavié, Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, arXiv preprint arXiv:2412.13663 (2024).

[6] 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: Advances in Information Retrieval: 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2023, pp. 236–241. URL: https://link.springer.com/chapter/10.1007/978-3-031-28241-6_20. doi:10.1007/978-3-031-28241-6_20.