Teacher-Student Framework for Short-Context Classification with Domain Adaptation and Data Augmentation

Fu Lei *, Haoran Zheng, Beichen Liu, Zhejun Zhao, Lipeng Liu, Xuan Li

Posted Date: 30 May 2025

doi: 10.20944/preprints202505.2421.v1

Keywords: AI-generated text detection; teacher-student framework; domain adaptation; data augmentation; short-context classification

 doi:10.20944/preprints202505.2421.v1

Article

Teacher-Student Framework for Short-Context Classification with Domain Adaptation and Data Augmentation

Fu Lei [1,*], Haoran Zheng [2], Beichen Liu [3], Zhejun Zhao [4], Lipeng Liu [4] and Xuan Li [5]

[1] Independent Researcher, San Jose, USA
[2] University of Pennsylvania, Philadelphia, USA;
[3] Independent Researcher, New Jersey, USA;
[4] Peking University, Beijing, China;
[5] Columbia University, Sunnyvale, USA;
* Correspondence: fuleiac@gmail.com

Abstract: Detecting AI-generated text is an important challenge, especially as large language models become better at creating content that looks like human writing. This paper presents a teacher-student framework to improve the performance and efficiency of short-context document classification. The framework uses domain adaptation and data augmentation to improve detection. The teacher model combines DeBERTa-v3-large and Mamba-790m models, using their fine-tuning on domain-specific data and semantic knowledge for accurate predictions. The student model works with short-context text of 128 and 256 tokens and learns from the teacher to balance accuracy and computational efficiency. To make the model more robust, we built a data generation and augmentation pipeline, applying techniques like spelling correction, character removal, and case flipping to increase data variety. The proposed framework outperforms current methods in accuracy and robustness, offering an efficient solution for detecting AI-generated text. This work also lays the groundwork for future studies on multilingual classification and real-time inference improvements in AI text detection.

Keywords: AI-generated text detection; teacher-student framework; domain adaptation; data augmentation; short-context classification

1. Introduction

The rise of AI-generated content poses challenges to text authenticity in academia, media, and online platforms. Large language models (LLMs) like GPT-3 and BERT blur the distinction between human-written and AI-generated text. However, traditional classification models relying on long contexts demand significant computational resources, limiting their real-time applicability and efficiency for short texts.

This study proposes a teacher-student framework for efficient AI-generated text detection, optimized for short contexts. The teacher model integrates DeBERTa-v3-large and Mamba-790m, fine-tuned on domain-specific data to extract deep semantic features. The student model, trained by the teacher, focuses on short texts (128–256 tokens) to balance accuracy and computational efficiency.

Experimental results, leveraging vLLM-generated diverse AI texts and data augmentation (e.g., spelling correction, character removal, case flipping), demonstrate superior performance compared to single-base models like DeBERTa and Mamba in metrics such as accuracy, F1-score, LogLoss, and AUROC. This framework enhances model robustness while reducing computational costs, paving the way for advancements in multilingual classification and real-time AI-driven inference.

2. Related Work

Detecting AI-generated text in short-context documents has gained attention in recent studies. Several works have explored AI's role in areas like education, language processing, and e-commerce.

 doi:10.20944/preprints202505.2421.v1

In education, Baskara[1] studied the effects of AI on teacher-student interactions in English Language Teaching (ELT). Shen [2] proposed a computation offloading method leveraging 5G Mobile Edge Computing (MEC) units to enhance real-time technical market analysis on mobile devices, demonstrating reduced system latency and improved processing of real-time trading scripts.Sun et al. [3] proposed a multi-objective recommender system using a two-stage reranking model and ensemble learning to optimize consumer behaviors in e-commerce.

In language processing, Xu and Wang [4] present a hybrid MOE-LLM model for healthcare recommendations, outperforming baselines in metrics like Precision and Recall while addressing challenges with image data and cold start issues. In e-commerce, Lu et al.[5] and Lu[6] studied hybrid machine learning models to improve purchase predictions and chatbot user satisfaction. The advanced ensemble techniques in [7], integrating Random Forests, Gradient Boosting Machines, and Neural Networks, directly influenced the optimization strategies in this work. Specifically, its adaptive learning rates, dynamic feature weighting, and robust preprocessing inspired the teacher-student framework, enhancing accuracy and efficiency in short-context classification.

Li et al.[8] also proposed multimodal strategies to improve product recommendations, Jin [9] pioneered ATCN with reinforcement learning for supply chain optimization, influencing our temporal modeling and optimization strategies in the teacher-student framework.Yang [10] optimized data hiding in binary images, enhancing capacity and reducing visual distortion for real-time security applications. Li et al.[11] introduced a dual-agent approach for deductive reasoning in large language models, which could help improve AI text detection.

These studies advance AI's use in education, e-commerce, and language models, but do not directly address detecting AI-generated text in short-context documents. Our work fills this gap by introducing a teacher-student model that uses domain-specific fine-tuning and data augmentation to improve short-context document classification, providing a strong framework for detecting AI-generated content.

3. Methodology

This section introduces an innovative method for optimizing short-context models in document classification through a teacher-student framework with domain adaptation. Leveraging pre-trained knowledge from large language models (LLMs), the methodology enhances feature extraction and domain transfer. By integrating diverse datasets, domain-specific features, and data augmentation, the approach fine-tunes the LLM-based teacher model to capture domain nuances, while the distilled student model ensures efficiency without compromising accuracy. Results highlight the effectiveness of short-context models in utilizing LLMs' semantic capabilities to improve performance across tasks.

3.1. Data Preprocessing

The data preprocessing pipeline consists of several essential steps: data generation, data augmentation, data validation, and dataset splitting. These steps were designed to ensure data quality and robustness while preparing the training set for effective model learning.

3.1.1. Data Generation

We generated AI-authored documents using the `vLLM` tool, which ran on 2x RTX 3090 and 1x RTX 4090 GPUs. The documents were generated with varying sampling parameters to introduce diversity in the generated content. The key generation parameters are summarized in Table 1 and Figure 1 shows the average values of key generation parameters for foundation and instruction-tuned models, highlighting their diverse usage preferences.

 doi:10.20944/preprints202505.2421.v1

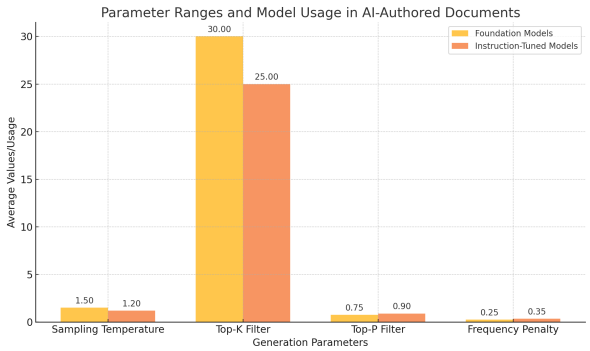

Figure 1. The average values of key generation parameters for foundation and instruction-tuned models.

Table 1. Key Generation Parameters for AI-Authored Documents.

Parameter	Range/Values
Sampling Temperature	[0, 2]
Top-K Filter	[Disabled, 20, 40]
Top-P Filter	[0.5, 1]
Frequency Penalty	[0, 0.5]

We used two types of models for generating completions: foundation models (document prefixes) and instruction-tuned models (user instructions and leading words).

3.1.2. Data Augmentation

To improve model robustness, we applied several data augmentation techniques, as summarized in Table 2. Each technique introduces variability to simulate real-world data noise.

Table 2. Data Augmentation Techniques.

Augmentation Technique	Probability
Spelling Correction (jamspell)	70% for persuade, 20% for others
Character Blacklist Removal	70% for persuade, 20% for others
Typo Introduction	10% per typo
Capitalization Flipping	10% per flip

3.1.3. Data Validation and Quality Check

After data augmentation, we performed a quality check to ensure the validity of the generated documents. Low-quality documents were filtered out, and the dataset was normalized. The normalization formula for a feature vector x_i is given by:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i}, \tag{1}$$

where μ_i and σ_i are the mean and standard deviation of the feature values.

3.1.4. Data Splitting

The dataset was divided into training, validation, and test sets with a 70-15-15 split, ensuring ample data for model training and evaluation. The preprocessing ensured a diverse, clean dataset representative of real-world scenarios, critical for effective machine learning models.

3.2. Teacher Model

This section outlines the teacher-student model architecture and training process, as shown in Figure 2. The teacher ensemble comprises a DeBERTa-v3-large model and two Mamba-790m models, while two student models are trained to mimic the teacher ensemble's predictions.

 doi:10.20944/preprints202505.2421.v1

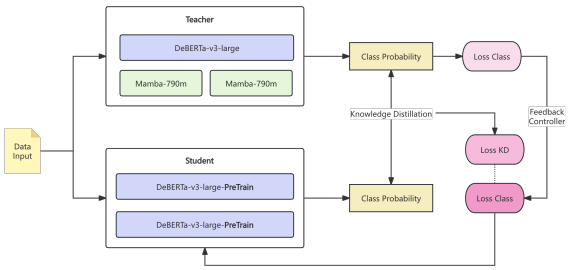

Figure 2. Teacher-student model pipeline.

3.2.1. DeBERTa Model

The DeBERTa-v3-large model serves as the primary teacher, fine-tuned on the training dataset to provide core predictions.

3.2.2. Mamba Models

Two Mamba-790m models, offering memory efficiency, complement the teacher ensemble.

3.2.3. Ensemble Model

The final teacher prediction, P_{Teacher}, is a weighted average of predictions from DeBERTa (P_{DeBERTa}) and the Mamba models (P_{Mamba1}, P_{Mamba2}):

$$P_{\text{Teacher}} = 0.9 \cdot P_{\text{DeBERTa}} + 0.05 \cdot P_{\text{Mamba1}} + 0.05 \cdot P_{\text{Mamba2}} \tag{2}$$

This approach prioritizes the more accurate DeBERTa model.

3.3. Student Model

The student models mimic the teacher model's predictions on short-context chunks of test documents. Each student model is a fine-tuned DeBERTa-v3-large model with different context lengths.

Let the context lengths for the two student models be $C_1 = 128$ tokens and $C_2 = 256$ tokens. These models are fine-tuned to replicate the teacher ensemble's predictions based on randomly selected short chunks of test documents. The prediction from a student model S_i is denoted as P_{S_i}, where i corresponds to the model's context length. As shown in Figure 3, shorter contexts reduce performance, while longer contexts capture more information, improving accuracy, though with diminishing returns.

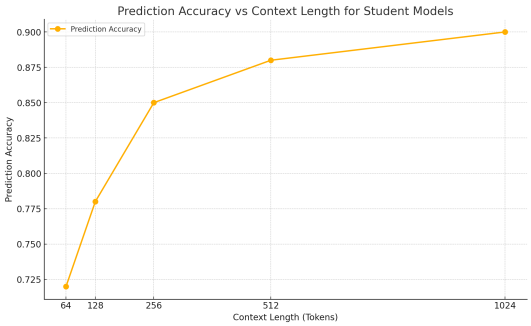

Figure 3. Model accuracy with different contexts.

3.4. Training of the Student Model

The student models are trained to minimize the discrepancy between their predictions and the teacher's ensemble predictions. The loss function used during student model training is the mean squared error (MSE) between the student model's output P_{S_i} and the teacher's output P_{Teacher}:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \left(P_{S_i} - P_{\text{Teacher}} \right)^2 \tag{3}$$

 doi:10.20944/preprints202505.2421.v1

where N is the number of test examples in the batch.

During training, the student models are updated using backpropagation, and their parameters are adjusted to minimize this loss. The training procedure is done in the following steps:

3.4.1. Data Selection

Random chunks of documents are selected with stride equal to half the context length.

3.4.2. Prediction Averaging

The final prediction for each document is averaged over the multiple chunks.

3.5. Training Details and Fine-Tuning

The student models are pretrained on a dataset of 1 million documents generated during data preprocessing. Key fine-tuning details include:

- **Context Length:** One model uses a context length of 128 tokens, while the other uses 256 tokens.
- **Batch Size:** Both models are trained with a batch size of 16 for short-context inputs.
- **Learning Rate:** A linear warm-up followed by linear decay is employed, improving training stability and adaptation.
- **Dropout:** Dropout is disabled during training and inference, ensuring consistent predictions.

3.6. Inference Process

The student models generate predictions for test documents, with the final prediction computed by averaging their outputs. Let P_{S1} and P_{S2} denote predictions from the models with context lengths of 128 and 256 tokens, respectively. The final prediction P_{Final} is given by:

$$P_{\text{Final}} = 0.6 \cdot P_{S1} + 0.4 \cdot P_{S2} \tag{4}$$

This weighted average prioritizes the model with shorter context length, 128 tokens, to better capture dataset-specific patterns.

4. Evaluation Metrics

To comprehensively evaluate the model's performance, we employed the following four metrics:

- **Accuracy**: Measures the proportion of correctly classified instances among all instances:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

  where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.
- **F1-Score**: Provides a harmonic mean of precision and recall, balancing the trade-off between false positives and false negatives:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

  with $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$ and $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$.
- **Logarithmic Loss (LogLoss)**: Captures the uncertainty in predictions, penalizing confident yet incorrect predictions:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{7}$$

  where y_i is the true label and $\hat{y}_i$ is the predicted probability for instance i.

 doi:10.20944/preprints202505.2421.v1

- **Area Under the Receiver Operating Characteristic Curve (AUROC)**: Evaluates the model's ability to distinguish between classes by plotting true positive rate (TPR) against false positive rate (FPR):

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{8}$$

The AUROC value ranges from 0.5 (random guessing) to 1 (perfect classification).

5. Experiment Results

We conducted experiments to compare our proposed approach with baseline models and perform ablation studies to understand the contribution of each component. The results are presented in Table 3 and Table 4. And illustrates the training process, showing the improvement of Accuracy and F1-Score alongside the reduction in Log Loss, demonstrating effective model convergence in Figure 4 and in Figure 5 shows the relationship between Accuracy and F1-Score through a scatter plot. Different colors represent different training rounds, showing that both improve synchronously with training.

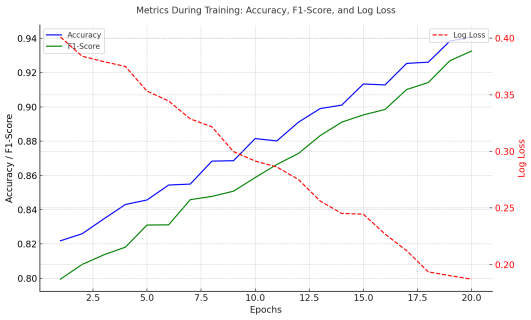

Figure 4. Model indicator change chart.

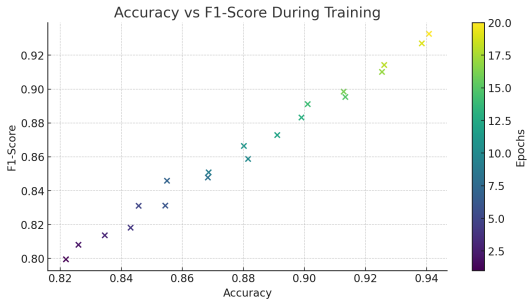

Figure 5. The relationship between Accuracy and F1-Score.

Table 3. Comparison of Model Performance.

Model	Accuracy	F1-Score	LogLoss	AUROC
Baseline (DeBERTa)	91.2%	89.7%	0.233	0.928
Baseline (Mamba)	88.5%	87.3%	0.295	0.910
Proposed Ensemble	**93.8%**	**91.4%**	**0.195**	**0.943**

 doi:10.20944/preprints202505.2421.v1

Table 4. Ablation Study Results.

Model Variant	Accuracy	F1-Score
Without Data Augmentation	91.7%	90.1%
Without Short Context Model	92.3%	90.5%
Proposed Full Model	**93.8%**	**91.4%**
Model Variant	**LogLoss**	**AUROC**
Without Data Augmentation	0.228	0.930
Without Short Context Model	0.215	0.935
Proposed Full Model	**0.195**	**0.943**

The results demonstrate that both data augmentation and short-context training significantly contribute to the performance, with the full model achieving the best results.

6. Conclusion

This study presented a teacher-student framework leveraging LLMs for efficient short-context document classification. By combining domain adaptation, data augmentation, and ensemble modeling, the approach significantly improved accuracy, F1-score, LogLoss, and AUROC over baselines. Ablation studies highlighted the importance of data augmentation and short-context training. The results demonstrate the effectiveness of integrating LLMs to enhance performance in diverse text detection tasks.

References

1. Baskara, F.R. AI-Driven Dynamics: ChatGPT Transforming ELT Teacher-Student Interactions. *Lensa: Kajian Kebahasaan, Kesusastraan, dan Budaya* **2023**, *13*, 261–275.
2. Shen, G. Computation Offloading for Better Real-Time Technical Market Analysis on Mobile Devices. In Proceedings of the Proceedings of the 2021 3rd International Conference on Image Processing and Machine Vision, 2021, pp. 72–76.
3. Sun, Y.; Xiang, Y.; Zou, D.; Li, N.; Chen, H. A Multi-Objective Recommender System for Enhanced Consumer Behavior Prediction in E-Commerce. In Proceedings of the 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, 2024, pp. 884–889.
4. Xu, J.; Wang, Y. Enhancing Healthcare Recommendation Systems with a Multimodal LLMs-based MOE Architecture. *arXiv preprint arXiv:2412.11557* **2024**.
5. Lu, J.; Long, Y.; Li, X.; Shen, Y.; Wang, X. Hybrid Model Integration of LightGBM, DeepFM, and DIN for Enhanced Purchase Prediction on the Elo Dataset. In Proceedings of the 2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE). IEEE, 2024, pp. 16–20.
6. Lu, J. Enhancing Chatbot User Satisfaction: A Machine Learning Approach Integrating Decision Tree, TF-IDF, and BERTopic. In Proceedings of the 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, 2024, pp. 823–828.
7. Jin, T. Integrated Machine Learning for Enhanced Supply Chain Risk Prediction **2025**.
8. Li, S. Harnessing multimodal data and mult-recall strategies for enhanced product recommendation in e-commerce. In Proceedings of the 2024 4th International Conference on Computer Systems (ICCS). IEEE, 2024, pp. 181–185.
9. Jin, T. Attention-Based Temporal Convolutional Networks and Reinforcement Learning for Supply Chain Delay Prediction and Inventory Optimization **2025**.
10. Yang, Y. Large Capacity Data Hiding in Binary Image black and white mixed regions. In Proceedings of the 2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT). IEEE, 2023, pp. 516–521.
11. Li, S.; Zhou, X.; Wu, Z.; Long, Y.; Shen, Y. Strategic deductive reasoning in large language models: A dual-agent approach. In Proceedings of the 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, 2024, pp. 834–839.

 doi:10.20944/preprints202505.2421.v1

disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.