Team DS@GT at PAN 2026: Fine-Tuning Fundamentals and Data Augmentation

Notebook for the PAN Lab at CLEF 2026

Harshul Basava[1,*], Murilo Gustineli[1,†]

[1]Georgia Institute of Technology, 225 North Avenue, Atlanta, 30332, United States

Abstract

Voight-Kampff is a task under the PAN lab on digital text forensics and stylometry at the Conference and Labs of the Evaluation Forum (CLEF). Participants are required to develop a classification system that denotes a provided text as human-authored or machine-generated, and are provided with a training dataset comprised of sample texts with labels. Evaluation is performed on two distinct sets: a curated evaluation set created by the task developers, and an adversarial set generated by participants of the Eloquent task, which aims to confuse classification systems. The DS@GT PAN team explored one primary strategy to complete the task. We fine-tune Qwen2.5-1.5B on the provided training dataset, and evaluate on external datasets to ensure robustness to new examples. Using this methodology, we achieve a ROC-AUC score of 0.981 on the curated evaluation set, and a ROC-AUC score of 0.918 on the adversarial Eloquent set. Supporting code for this paper can be found at https://github.com/dsgt-arc/pan-2026.

Keywords

Authorship Verification, Supervised Fine-Tuning, LoRA, Obfuscation Training, PAN 2026

1. Introduction

The Voight-Kampff 2026 task hosted by the PAN Lab involves classifying (potentially obfuscated) text as human-authored or machine-generated [1, 2]. Participants are provided ~20,000 labeled writing samples sourced from 20+ language models and thousands of human authors for training and validation. This year, the test set additionally contains new models and unknown obfuscations to evaluate classifier robustness. Evaluation is hosted on the TIRA platform [3].

We LoRA fine-tune Qwen2.5-1.5B on the PAN-provided training data for binary classification of human-authored versus machine-generated text. To handle long inputs, we apply a sliding-window chunking strategy over a 512-token limit, averaging predicted probabilities across chunks. We additionally evaluate obfuscation training via homoglyph substitution and synonym replacement, finding that neither augmentation technique meaningfully improves robustness, and thus submit our base fine-tuned model.

1.1. Overview

In Section 2, we discuss related work and strategies from PAN 2025. In Section 3, we investigate the distribution of the provided data. In Section 4, we review our methodology and the various experiments we've run. In Section 5, we present our results. In Section 6, we discuss these results.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

† Mentorship role.

harshul@gatech.edu (H. Basava); murilogustineli@gatech.edu (M. Gustineli)

https://harshul-basava.github.io (H. Basava); https://murilogustineli.com/ (M. Gustineli)

2. Background and Related Work

Text authorship verification is an example of sequence classification, inviting diverse approaches with varying levels of success. We review several of these approaches, which guide our strategy.

2.1. Provided Baselines

PAN provides trained baselines to measure performance against. The **Linear SVM with TF-IDF features** [1] is a linear support vector machine used to classify between machine-generated and human-authored text. The Linear SVM is trained using Term Frequency-Inverse Document Frequency (TF-IDF), which measures word importance based on frequency within specific documents and frequency across all documents.

PPMd Compression-based Cosine [4] [5] is a technique for authorship verification that relies on compressing files using PPMd (Prediction by Partial Matching, Variant d.). If file A compresses well when using file B as a dictionary, then file A and file B are considered similar. This technique proves to be a useful unsupervised technique and an interesting approach to authorship verification.

Binoculars [6] relies on measuring *perplexity* in text through language models, defined by taking the exponential of the average negative log-probability of the tokens.

$$\text{PPL}(s) = exp\left(-\frac{1}{L}\sum_{i=1}^{L}\log P(s_i)\right)$$

Binoculars utilizes two language models to measure the perplexity in the text, and compares their evaluations in order to classify the text as human-authored or machine-generated. If both models assign high perplexity (high surprise), the text is likely human-generated. Otherwise, the text is most likely machine-authored.

2.2. Data Augmentation

The top submissions to PAN 2025 all approached the task uniquely but relied on data augmentation techniques to improve the robustness of their classification systems.

Macko 2025 [7] fine-tuned Qwen3-14B using QLoRA for binary classification. To augment his data, he utilized obfuscation via a *homoglyph attack* (replacement of characters with similar-looking Unicode characters) of parts of the training data. Additionally, he created his validation set using external training data to select the best model based on out-of-domain performance. The external training dataset is a collection of 2,000 examples across seven languages, sampled from 18 different AI-detection datasets of different genres and domains.

Valdez-Valenzuela et al. 2025 [8] begins by generating a syntactic dependency graph representation of the text. Sentence-level graphs are merged into document-level graphs, which are embedded using a graph neural network for use in a dense neural network for classification. The dataset is augmented with three different kinds of obfuscations (*shortening, Unicode replacement, paraphrasing*) to make the system more robust.

Liu et al. 2025 [9] Used a soft ensemble of fine-tuned Qwen3-4B and fine-tuned ModernBERT with contrastive loss to distinguish between human and LLM-generated text. In order to improve robustness, Liu et al. took 1000 of the human written samples and paraphrased using GPT-4.1 to acquire more AI generated text. They generated these paraphrased samples at the sentence level – these new documents are marked as machine generated.

2.3. Evaluation Metrics

The PAN committee provided five metrics for measuring how successful techniques are.

ROC-AUC is calculated as the area under the Receiver Operating Characteristic curve. This curve plots true positive rate against false positive rate at various thresholds. A score of 0.5 signifies random chance. A score of 1.0 signifies perfect classification.

Brier Score is calculated based on the differences between forecasted probabilities and final outcomes. Brier Score (for PAN 2026) is defined by the below equation, where f is the forecasted probability and o is a binary outcome.

$$\text{BS} = 1 - \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2$$

C@1 is a modified accuracy score that assigns non-answers (score = 0.5) the average accuracy of the remaining cases. This penalizes overconfident mistakes over non-responses.

$\mathbf{F_1}$ score is defined as the harmonic mean of precision and recall. This metric proves useful for the Voight-Kampff task due to the class imbalance in the provided data (see Section 3).

$\mathbf{F_{0.5u}}$ is a modified $F_{0.5}$ measure (precision-weighted F measure) that treats non-answers (score = 0.5) as false negatives. In contrast to C@1, this discourages non-answers by penalizing indecision.

Table 1
Comparison Table across Performance Metrics

Team	ROC-AUC	Brier	$C@1$	F_1	$F_{0.5u}$	**Mean ↓**
Macko [7]	0.995	0.984	0.982	0.989	0.993	**0.989**
Valdez-Valenzuela [8]	0.939	0.902	0.897	0.926	0.960	**0.929**
Liu [9]	0.962	0.891	0.889	0.923	0.963	**0.928**
TF-IDF SVM [1]	0.963	0.900	0.897	0.904	0.946	**0.922**
Binoculars [6]	0.827	0.856	0.818	0.866	0.907	**0.863**
PPMd CBC [4] [5]	0.644	0.802	0.759	0.817	0.839	**0.790**

3. Exploratory Data Analysis

Sample Distribution PAN provided 23,707 total samples, comprised of 9,101 human-written samples and 14,606 machine-generated samples, or approximately a 40/60 split. Data is split between three genres: fiction, essays, and news. For the machine-generated data specifically, data was generated by 23 unique generator models (e.g. gpt-3.5-turbo, gemini-2.0-flash, llama-3.3-70b-instruct, etc.).

Length Distribution We compare machine-generated and human-written text across textual metrics related to length and uniqueness. We find that human-written text is longer than machine-generated text on average, yet machine-generated text tends to use longer words (Figure 1). Sentence length and Type-Token Ratio remain relatively consistent across both author groups, with slight variations.

External Datasets We additionally evaluate against 6 external datasets: HC3 [10], RAID [11][1], MAGE [12], OpenGPTText [13], AI Detection Pile [14], and M4 [15]. Each of these datasets are approximately balanced between human and AI samples, and focus on a different genre than PAN. See Table 2 for further details.

[1]RAID results are averaged across the base and adversarial splits.

Table 2
External Benchmark Datasets. We evaluate against each of these datasets to test classification robustness. We typically sample 5000 examples from each dataset to test our model against.

Dataset	Genre/Domain	Source
HC3	Wikipedia Q&A	Guo et al. [10]
RAID	Multi-domain	Dugan et al. [11]
MAGE	Multi-domain	Li et al. [12]
OpenGPTText	Web Articles	Chen et al. [13]
AI Detection Pile	Essays, News	Artemyev [14]
M4	Multi-domain, Multi-lingual	Wang et al. [15]

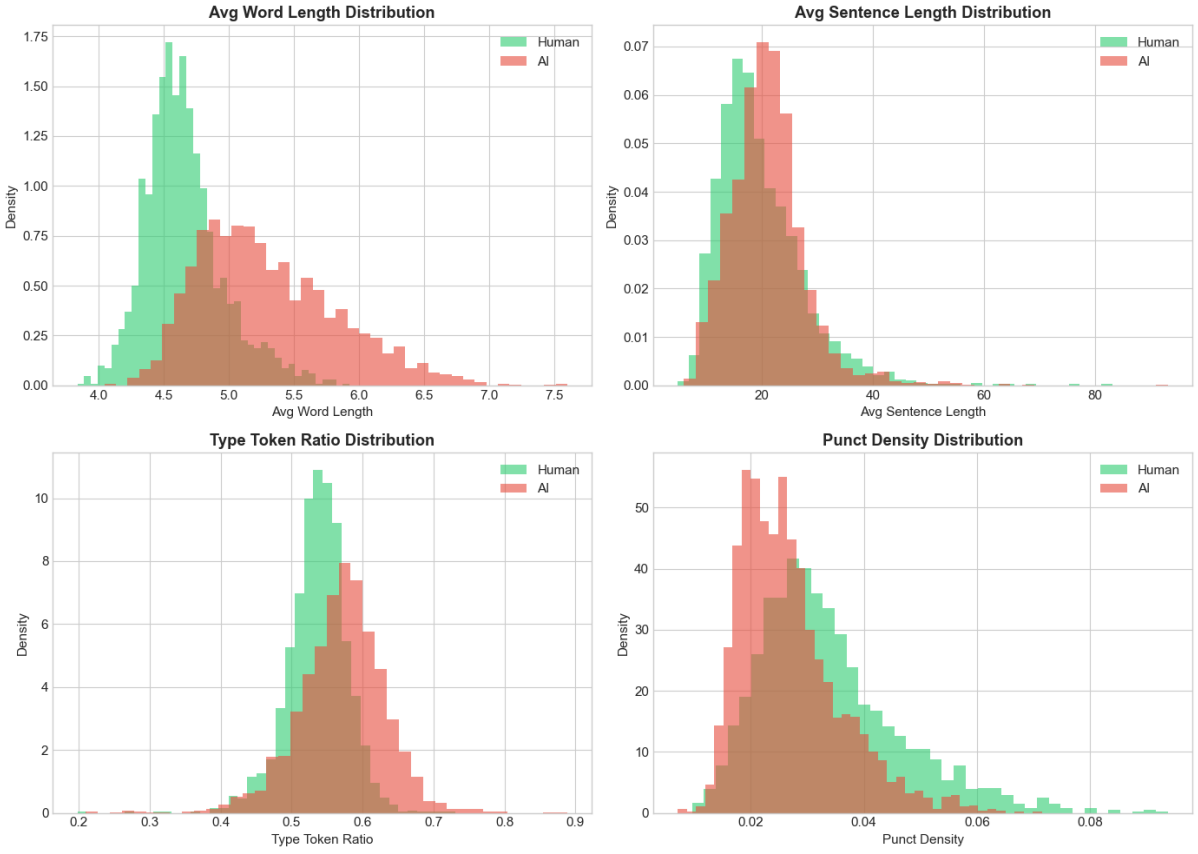

Figure 1: Linguistic feature comparison between human-written and machine-generated samples. We observe differences in average word length distribution (top left) and punctuation density (bottom left). Human writing tends toward shorter average word lengths within a narrower range (averages falling between 4 and 6 characters) while AI-generated text shows averages extending up to 7 characters. We also see a consistent decrease in punctuation density for AI models compared to human writing.

4. Methodology

We LoRA fine-tune Qwen2.5-1.5B on the training dataset provided by PAN. We evaluate our fine-tuned model on several out-of-distribution (OOD) datasets to measure robustness to unseen data and obfuscation.

Training We fine-tune Qwen2.5-1.5B with the following hyperparameters: Epochs 3, LoRA Rank 16, LoRA Alpha 32, Batch Size 16, Learning Rate 2e-4. We performed validation using the provided validation set, and used early stopping patience 3 for ROC AUC.

Since we utilized a 512 token limit, we used a sliding-window strategy to chunk text and average the scores across each chunk. The model outputs a probability, and we chose the standard threshold of 0.5 to differentiate between human and machine.

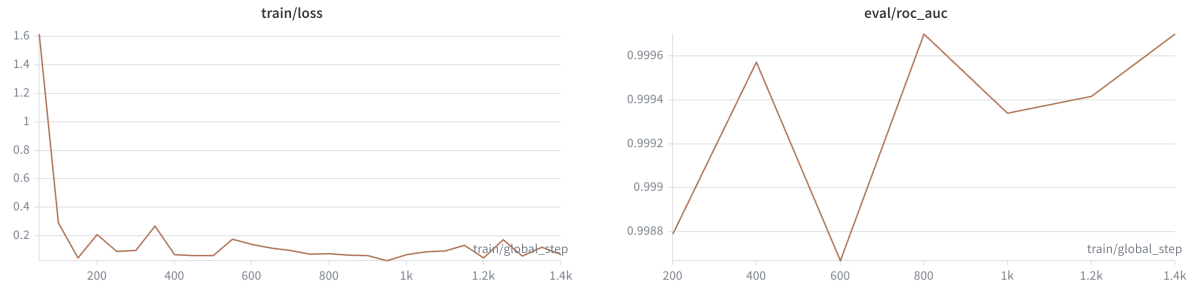

Figure 2: Train **(Left)** and test **(Right)** curves for Qwen2.5-1.5B fine-tuned on the original PAN training data. The model performed extremely strongly on in-distribution data early on in training, triggering early stopping and causing training to complete in around a singular epoch.

OOD Evaluation We evaluate on 6 external datasets. See Section 3 for further dataset details. Table 3 summarizes our results across datasets and metrics, including PAN validation data. Notably, our model performs exceedingly well on some OOD datasets, yet struggles on others.

Obfuscation Training We additionally fine-tune our model on two augmented datasets.

- We fine-tune on the original training dataset augmented using homoglyph substitutions according to [7]. We used a 5% character swap rate and a 5% zero-width joiner insertion rate on 10% of machine-generated texts.

- We fine-tune on the first augmented training dataset with an additional synonym substitution. We used a 20% per-word replacement rate on a different 10% of machine-generated texts.

Table 3
Performance Metrics Across Evaluation Datasets

Dataset	ROC-AUC	Brier	$C@1$	F_1	$F_{0.5u}$	**Mean**
HC3 Wiki	0.9977	0.9924	0.9917	0.9917	0.9896	**0.9926**
RAID	0.9985	0.9945	0.9935	0.9935	0.9932	**0.9946**
M4 / SemEval 2024	0.9992	0.9928	0.9920	0.9920	0.9944	**0.9941**
OpenGPTText	0.9447	0.8695	0.8560	0.8345	0.9167	**0.8843**
MAGE	0.6789	0.6606	0.6460	0.4701	0.6698	**0.6251**
AI Detection Pile	0.4486	0.5749	0.5620	0.2748	0.4531	**0.4627**
PAN Validation	0.9993	0.9979	0.9983	0.9987	0.9990	**0.9986**

We find that obfuscation training offers little benefit on model performance for some datasets, and reduces performance on others. Thus, we stick to fine-tuning on the original PAN training data and avoid both data augmentation techniques.

5. Results

We list results on the two evaluation sets provided by PAN in Table 4: the official validation set and the eloquent submissions. Notably, the model performs significantly worse on the eloquent submissions, which are specifically generated to fool classifiers of human-written and machine-generated text.

Table 4
PAN Official Evaluation Results

Submission	ROC-AUC	Brier	$C@1$	F_1	$F_{0.5u}$	**Mean**
eloquent-20260523-test	0.918	0.681	0.625	0.751	0.880	**0.771**
pan26-generative-ai-detection-20260507-test	0.981	0.885	0.873	0.907	0.961	**0.921**

6. Discussion

OOD Variation One of the surprising findings in our evaluation of model performance was the stark difference in out-of-distribution performance on different datasets. In some cases, our model performed near perfectly, while in others, the model suffered significant performance drops, sometimes falling below random chance. We will continue to explore why the model's performance has extremely high variance - one hypothesis is that training solely on the PAN data caused the model to overfit on specific stylistic choices that seem to match in some datasets, but negatively correlate in others.

Model Size Notably, we performed our experiments using a relatively small model of only 1.5B parameters. While this enabled greater efficiency of classification, we are unsure of the performance costs this brings. Future work includes replicating our experiments using larger models, and measuring changes in performance - whether they will be positive or negative, we are unsure.

Acknowledgments

We thank the Data Science at Georgia Tech (DS@GT) CLEF competition group for their support. This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) [16] at the Georgia Institute of Technology, Atlanta, Georgia, USA.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. Bevendorff, M. Fröbe, A. Greiner-Petter, A. Jakoby, M. Mayerl, P. Nakov, H. Plutz, M. Potthast, B. Stein, M. N. Ta, Y. Wang, E. Zangerle, Overview of PAN 2026: Voight-Kampff Generative AI Detection, Text Watermarking, Multi-Author Writing Style Analysis, Generative Plagiarism Detection, and Reasoning Trajectory Detection, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2026.

[2] J. Bevendorff, R. R. Gunti, J. Karlgren, M. Fr"obe, M. Potthast, B. Stein, Overview of the third "Voight-Kampff" Generative AI / LLM Detection Task at PAN and ELOQUENT 2026, in: A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[3] 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. URL: https://link.springer.com/chapter/10.1007/978-3-031-28241-6_20. doi:10.1007/978-3-031-28241-6_20.

[4] D. Sculley, C. Brodley, Compression and machine learning: a new perspective on feature space vectors, in: Data Compression Conference (DCC'06), 2006, pp. 332–341. doi:10.1109/DCC.2006.13.

[5] O. Halvani, C. Winter, L. Graner, On the usefulness of compression models for authorship verification, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, ARES '17, Association for Computing Machinery, New York, NY, USA, 2017. URL: https://doi.org/10.1145/3098954.3104050. doi:10.1145/3098954.3104050.

[6] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. URL: https://arxiv.org/abs/2401.12070. arXiv:2401.12070.

[7] D. Macko, mdok of kinit: Robustly fine-tuned llm for binary and multiclass ai-generated text detection, 2025. URL: https://arxiv.org/abs/2506.01702. arXiv:2506.01702.

[8] A. Valdez-Valenzuela, H. Gómez-Adorno, M. Montes-y Gómez, Text graph neural networks for detecting AI-generated content, in: F. Alam, P. Nakov, N. Habash, I. Gurevych, S. Chowdhury, A. Shelmanov, Y. Wang, E. Artemova, M. Kutlu, G. Mikros (Eds.), Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect), International Conference on Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 134–139. URL: https://aclanthology.org/2025.genaidetect-1.10/.

[9] J. Liu, L. Kong, Z. Peng, F. Chen, Generative ai authorship verification based on contrastive-enhanced dual-model decision system, in: Conference and Labs of the Evaluation Forum, 2025. URL: https://api.semanticscholar.org/CorpusID:282376419.

[10] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is ChatGPT to human experts? comparison corpus, evaluation, and detection, arXiv preprint arXiv:2301.07597 (2023).

[11] L. Dugan, A. Hwang, F. Trhlik, J. M. Ludan, A. Zhu, H. Xu, D. Ippolito, C. Callison-Burch, RAID:

A shared benchmark for robust evaluation of machine-generated text detectors, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2024, pp. 12463–12492.

[12] Y. Li, Q. Li, L. Cui, W. Bi, Z. Wang, L. Wang, L. Yang, S. Shi, Y. Zhang, MAGE: Machine-generated text detection in the wild, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2024, pp. 36–53.

[13] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, B. Raj, GPT-Sentinel: Distinguishing human and ChatGPT generated content, arXiv preprint arXiv:2305.07969 (2023).

[14] A. Artemyev, AI text detection pile, https://huggingface.co/datasets/artem9k/ai-text-detection-pile, 2023.

[15] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, C. Whitehouse, O. Mohammed Afzal, T. Mahmoud, T. Sasaki, T. Arnold, A. F. Aji, N. Habash, I. Gurevych, P. Nakov, M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection, in: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2024, pp. 1369–1407.

[16] PACE, Partnership for an Advanced Computing Environment (PACE), 2017. URL: http://www.pace.gatech.edu.