ReText.Ai Team at PAN 2025: Applying a Multiple Classification Heads to a Transformer Model for Human-AI Collaborative Text Classification

Notebook for the PAN Lab at CLEF 2025

Daria Ignatenko[1,2,†], Konstantin Zaitsev[1,2,*,†] and Olga Shkriaba[1]

[1]*ReText.Ai Team, Moscow, Russia*
[2]*HSE University, Moscow, Russia*

Abstract

This paper presents the ReText.Ai team's solution to the Human-AI Collaborative Text Classification subtask of the PAN-2025 Generative AI Authorship Verification Task. Our approach involves fine-tuning transformer models, such as RoBERTa-base and Gemma-2 2B, with a custom multi-head classifier that includes a main multiclass head and auxiliary binary heads to better distinguish closely related labels. Through utilizing a transformer-based model augmented with multiple classification heads and a confidence-based override mechanism, our method outperforms the baseline, achieving macro Recall scores of 80.36% and 83.00% for RoBERTa-base and Gemma-2 2B, respectively, compared to 68.67% and 75.70% for the baseline models. In the competition, our team's fine-tuned Gemma-2-2B model achieved seventh place in the automated evaluation on the test set with a score of 56.11%.

Keywords

PAN 2025, Voight-Kampff Generative AI Detection 2025, Human-AI Collaborative Text Classification, AI-generated text detection, Multi-head classifier, RoBERTa, Gemma-2

1. Introduction

As more large language models (LLMs) produce complex and coherent content, the detection of generated texts becomes an important task. One aspect of this task is to determine whether an LLM was a co-author. In order to explore this challenge, the authors of the PAN-2025 [1] Generative AI Authorship Verification Task [2], a subtask of the Human-AI Collaborative Text Classification task, have provided a dataset that aims to classify texts into the following categories: *Fully human-written*; *Human-initiated, then machine-continued*; *Human-written, then machine-polished*; *Machine-written, then machine-humanized*; *Machine-written, then human-edited*; and *Deeply mixed text.*

In this paper, we describe our ReText.Ai Team approach to human-AI collaborative text classification. This approach involves using a pre-trained transformer model [3] with multiple classification heads to distinguish shared task labels. We fine-tuned several transformer-based models and selected the one that achieved the highest F1-score on the development set. In addition to the classification head with the proposed labels, we apply additional linear layers with classification heads to the model. These models are designed to distinguish closely related labels more effectively, thereby enhancing the overall quality. Our approach significantly improves on the baseline, raising macro Recall scores from 68.67% and 75.70% to 80.36% and 83.00% for RoBERTa-base [4] and Gemma-2 2B [5], respectively.

The paper is structured as follows. In Section 2, we provide information about a dataset. Section 3 provides a detailed description of our proposed architecture. We discuss the preprocessing of the dataset. We describe our initial experiments, which demonstrate that the baseline fails to distinguish between similar labels. To address this issue, we propose applying additional classification heads. In Section 4,

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
†These authors contributed equally.
d.ignatenko@retext.ai (D. Ignatenko); konstantzts@gmail.com (K. Zaitsev); o.shkriaba@retext.ai (O. Shkriaba)


we present the results obtained and compare them with those of the other participants in the shared task. We demonstrate how our approach improves upon the baseline.

2. Data

The dataset from the shared task contains samples with the following labels:

- **Fully human-written**: The document is entirely authored by a human without any AI assistance.
- **Human-initiated, then machine-continued**: A human starts writing, and an AI model completes the text.
- **Human-written, then machine-polished:** The text is initially written by a human but later refined or edited by an AI model.
- **Machine-written, then humanized:** An AI generates the text, which is later modified to obscure its machine origin.
- **Machine-written, then human-edited:** The content is generated by an AI but subsequently edited or refined by a human.
- **Deeply-mixed text:** The document contains interwoven sections written by both humans and AI, without a clear separation.

Table 1
Data distribution across different label categories for Train and Dev sets. The "–" indicates an unknown number of samples.

Label Category	Train	Dev	Test
Machine-written, then machine-humanized	91,232	10,137	–
Human-written, then machine-polished	95,398	12,289	–
Fully human-written	75,270	12,330	–
Human-initiated, then machine-continued	10,740	37,170	–
Deeply-mixed text	14,910	225	–
Machine-written, then human-edited	1,368	510	–
Total	**288,918**	**72,661**	**140,756**

The dataset was derived from various sources. It also contains additional information, such as the model that produced the text and the language used (English, Spanish, or German). The authors provide three subsets of the dataset: training, development, and testing. Labels are known for the training and development sets, but not for the test set. Table 1 presents the statistics for each subset.

3. Method

In this section, we describe our approach to developing a custom classification model for the Human-AI Collaborative Text Classification task. Our methodology leverages text preprocessing and fine-tuning a transformer-based architecture with a custom multi-head classifier.

3.1. Data Preprocessing

Firstly, we preprocess the dataset. Although modern neural network models do not require text preprocessing [6], we found that the texts in the dataset varied. This could lead to overfitting in some dataset sources. To prevent this and create more consistent samples, we implemented a preprocessing pipeline and applied it to each text sample. This consists of the following steps:

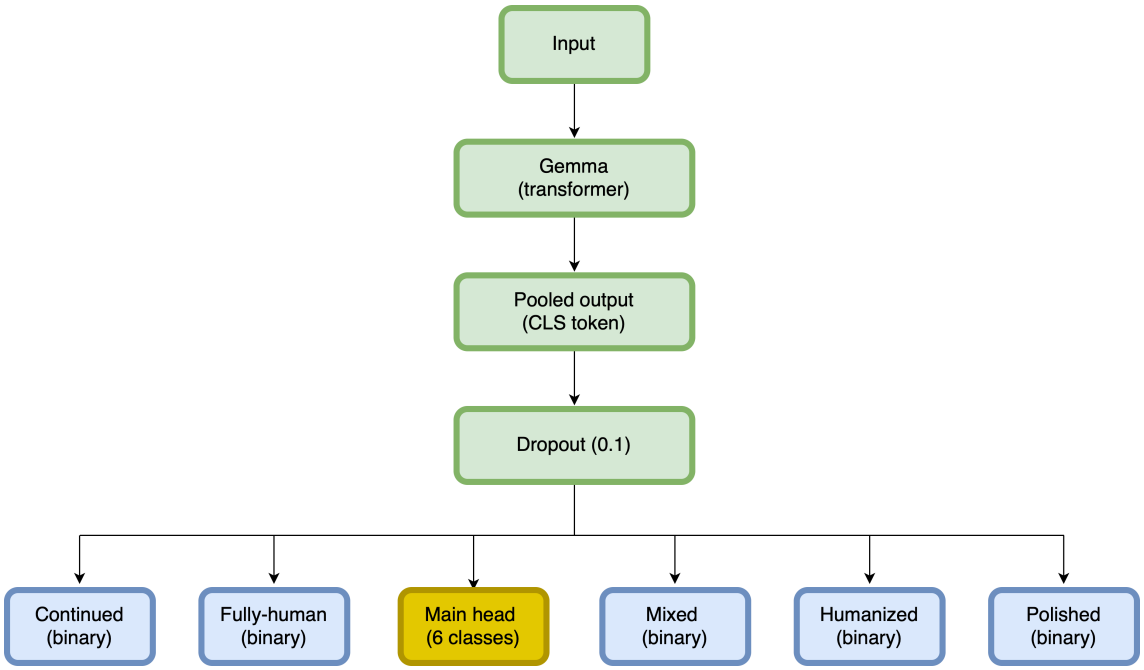

Figure 1: Architecture of custom multi-head classifier. Yellow color stands for the main head, auxiliary binary heads are colored in blue. Here, head *continued* corresponds to *human-initiated, then machine-continued*, *fully-human* corresponds to *Fully human-written*, *mixed* corresponds to *deeply-mixed text*, *humanized* corresponds to *Machine-written, then machine-humanized*, *polished* corresponds to *Human-written, then machine-polished*.

1. **Newline Removal:** All newline characters in the text are replaced with spaces to create a continuous string. This step prevents the model from interpreting newlines as token boundaries, which could disrupt the contextual understanding of sentences spanning multiple lines.
2. **Whitespace Normalization:** Multiple consecutive whitespace characters (e.g., spaces, tabs) are replaced with a single space.
3. **Text Stripping:** Leading and trailing whitespace is removed.

3.2. Fine-Tuning Multi-Head Classification Model

The next step in our approach involves fine-tuning a classifier. We conducted a series of experiments and found that models struggle to distinguish between certain classes. Based on the confusion matrix in Figure 2 for the RoBERTa baseline, we can see that the true labels *Machine-written, then humanized* are often predicted as *Fully human-written*, *Human-initiated, then machine-continued* and *Human-written, then machine-polished*. This suggests that it is difficult for the classifier to distinguish between these classes.

To tackle this issue, we propose that in addition to training the classifier on the task of predicting the main classes, we train the classifier to distinguish similar classes using additional heads that solve binary classification tasks. The essence of the approach is to predict, for similar classes, whether the text belongs to this class, or whether it belongs to any other class. The intuition of this approach is that the signals obtained from the binary classification heads will allow better delineation of examples with similar classes and, as a consequence, this may lead to an improvement in the final quality of the classifier.

As shown in Figure 1, the classifier is designed to predict multiple related labels using several heads that are trained in parallel:

- **Main head**: A multiclass classification head predicting one of the six categories: fully human-written, human-initiated, then machine-continued; human-written, then machine-polished; machine-written, then humanized; machine-written, then human-edited; and deeply mixed.

- **Auxiliary binary heads**: Five binary classification heads to detect specific subcategories (human-written, mixed, polished, continued, and humanized text), enhancing the model's ability to capture nuanced patterns. The introduction of binary heads helped decompose the complex task of distinguishing subtle patterns from the data into series of simpler ones.

Each classification head comprises a linear layer applied to the transformer's pooled output. Using single linear layers keeps the model's complexity in check, maintaining similar training times as without extra classification heads. A dropout rate of 0.1 is applied in classification heads to mitigate overfitting. The main head's loss is computed using weighted cross-entropy to address class imbalance, defined as:

$$\text{Loss}_{\text{main}} = -\sum_{i=1}^{N}\sum_{c=1}^{C} w_c \cdot y_{i,c} \cdot \log(\hat{y}_{i,c}), \tag{1}$$

where N is the number of samples, $C = 6$ is the number of classes, w_c is the weight for class c (inversely proportional to class frequency), $y_{i,c}$ is the true label indicator, and $\hat{y}_{i,c}$ is the predicted probability for class c.

To obtain the loss value $Loss_{aux}$ for each auxiliary classification head, we sum all the losses for the auxiliary heads:

$$\text{Loss}_{\text{aux}} = \text{Loss}_{\text{fully human}} + \text{Loss}_{\text{mixed}} + \text{Loss}_{\text{polished}} + \text{Loss}_{\text{continued}} + \text{Loss}_{\text{humanized}} \tag{2}$$

The final loss combines losses from all heads, weighted to prioritize the main multi-class head's prediction:

$$\text{Loss} = 0.6 \cdot \text{Loss}_{\text{main}} + 0.4 \cdot \text{Loss}_{\text{aux}} \tag{3}$$

During the evaluation phase in training and inference, the model generates logits for each classification head. To improve prediction accuracy, we implement a confidence-based override mechanism. For each sample, we compute softmax probabilities for all heads and apply class-specific confidence thresholds presented in Table 2.

Table 2
Confidence thresholds for auxillary classification heads.

	Fully Human	Mixed	Polished	Continued	Humanized
Threshold	0.85	0.7	0.85	0.9	0.8

The thresholds were assigned respectively to the assessed quality (F1) of each head. If a head's maximum probability exceeds its threshold and is the highest among all heads, the corresponding class is selected, overriding the main head's prediction by setting other logits to a large negative value (-1e9). This ensures that high-confidence predictions from specialized heads guide the final classification. The final prediction is then determined by the argmax of the modified logits.

Initially, we conducted experiments with the RoBERTa-base model[1]. The aim of these experiments was to demonstrate that our approach can enhance the baseline and, consequently, be transferred to stronger model architectures. After this, we fine-tuned the Gemma-2 2B model[2]. This model was chosen because of its size and its proven performance in classification tasks related to the detection of AI-generated content, as demonstrated in several studies [7, 8, 9].

All models were fine-tuned over 10 epochs. To prevent overfitting, we selected the best checkpoint according to the weighted F1-score across all classification heads on the development set. Such choice of key metric was made because it prioritizes performance on more frequent classes (e.g., fully human-written), which are likely more common in real-world scenarios, while still evaluating performance on

[1] https://huggingface.co/FacebookAI/roberta-base
[2] https://huggingface.co/google/gemma-2-2b

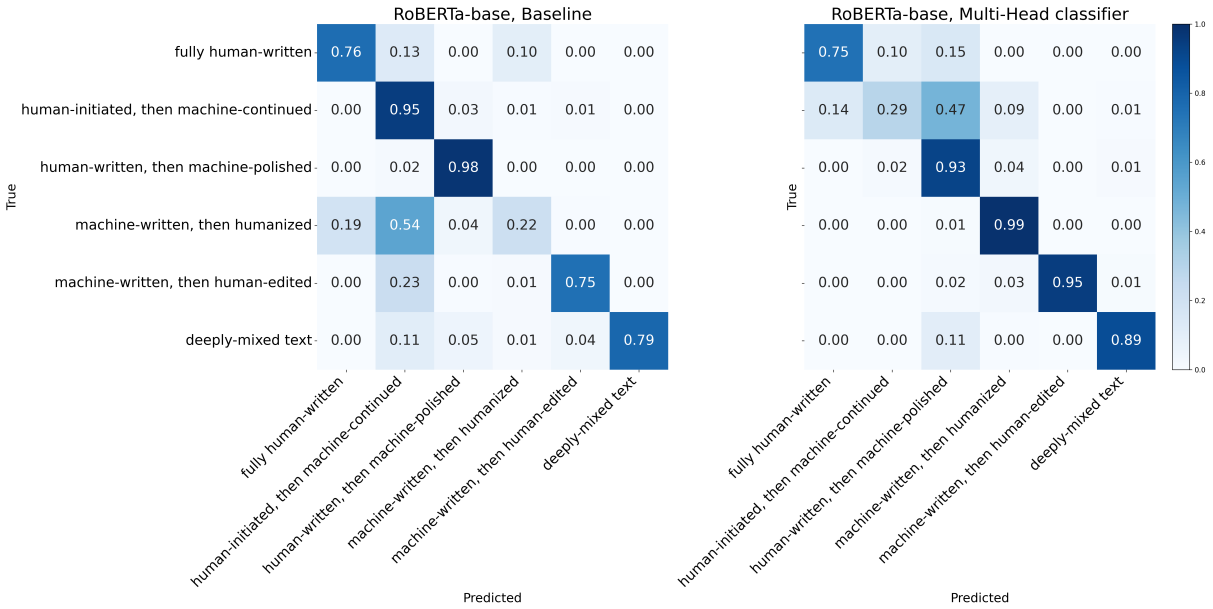

Figure 2: Confusion matrices for RoBERTa-base baseline (left) and for our fine-tuned multi-head RoBERTa-base model (right).

rare classes. This ensures that the metric reflects practical utility. Hyperparameters are shown in the appendix A.

4. Results

Table 3
Obtained metric results for the development set. The main metric used in the shared task is Macro Recall.

	Recall (Macro) ↑	F1 (Macro)	F1 (Micro)	Accuracy
Multi-Head Gemma-2 2B	83%	68%	59%	59%
Multi-Head RoBERTa-base	80%	66%	58%	58%
Baseline Gemma-2 2B	76%	55%	52%	52%
Baseline RoBERTa-base	74%	62%	55%	55%

Table 4
Table showing the performance metrics of different teams on the test set. Only the top seven scores out of 21 participants are shown here.

#	Team Name	Recall (Macro)	F1 (Macro)	Accuracy
1	mdok	64.46%	65.06%	74.09%
2	lbh-1130	61.72%	61.73%	69.28%
3	anastasiya.vozniuk	60.16%	60.85%	69.04%
4	Gangandandan	57.46%	56.31%	66.81%
5	Atu	56.87%	56.45%	66.33%
6	TaoLi	56.74%	55.39%	66.27%
7	*Our (Multi-Head Gemma-2 2B)*	56.11%	55.25%	64.79%
	Baseline	48.32%	47.82%	57.09%

The evaluation results on the development set are presented in Table 3. For the development set, we used Macro Recall, F1 Macro, F1 Micro, and Accuracy as these are used in the shared task. As

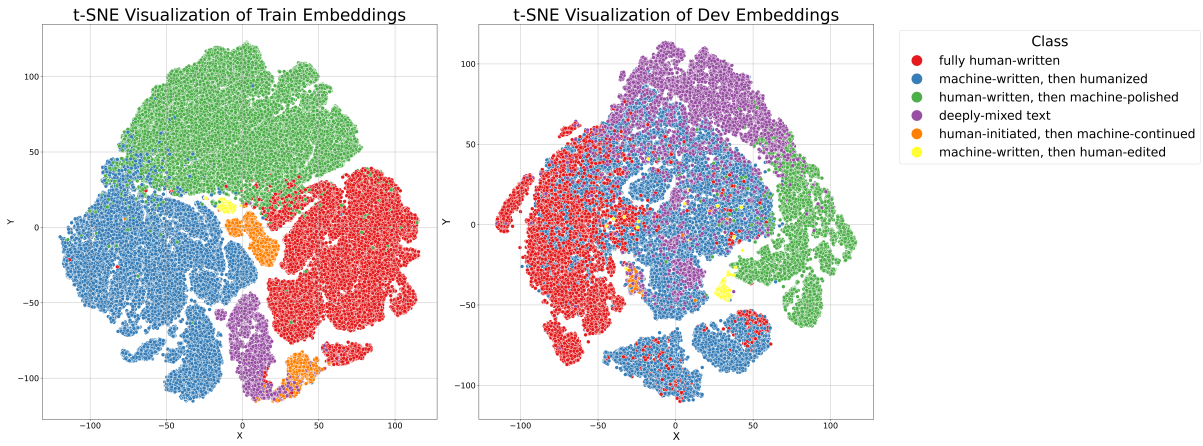

Figure 3: T-SNE visualization of the embeddings obtained from the Multi-Head RoBERTa classifier for the training (left) and development (right) sets.

can be seen in the table, adding additional heads increased all metrics for both RoBERTa-base and Gemma-2 2B. Specifically, the main metric for the shared task, Macro Recall, increased from 74% to 80% for RoBERTa-base, and from 76% to 83% for Gemma-2 2B.

Table 4 demonstrates our approach performance compared to the other participants in the shared task. As can be seen from the table, our team achieved 7th place, significantly improving on the baseline of 46.32% Macro Recall to reach 56.11%.

To compare the predictions obtained by a baseline model and a fine-tuned model, we created confusion matrices for the baseline and fine-tuned RoBERTa-base models. The confusion matrices were obtained by predicting the samples in the development set. Figure 2 presents these matrices. As can be seen from the figure, significant improvements were made to *machine-written, then humanized*, *machine-written, then human-edited* and *deeply-mixed text* labels. However, our approach failed to distinguish between *human-initiated, then machine-continued* and *human-written, then machine-polished*.

For further exploration of the quality of class differentiation, we obtained the final hidden states from the fine-tuned multi-head RoBERTa-base model for each data sample in the training and development sets. We then used the t-SNE algorithm [10] to visualize the embeddings, which are presented in Figure 3.

The figure shows that the classifier accurately distinguishes between embeddings related to different classes in the training set. However, for the development set, there are many noisy points located close to embeddings related to different classes. This means that the classifier has overfitted to the training set and is unable to generalize to unseen samples.

5. Conclusion and Future Work

In conclusion, our approach demonstrates the enhancement of text classification through the human-AI collaboration classification task. Using multiple heads on a pre-trained model and then fine-tuning the architecture significantly improves classification performance. Our key contribution lies in decomposing the complex classification problem into auxiliary binary tasks, thereby improving generalization and achieving significantly better results than the provided baselines in the test and development sets. On the test set leaderboard, we achieved a Macro Recall of 56.11% and came 7th out of 21 participants.

A possible direction for future research could be to add contrastive training to our approach. The detection of generated or collaborative texts could be defined as an authorship detection task, as has been demonstrated in other studies [11, 12]. Some texts were generated by specific models, and considering these models as authors, it may be possible to train a classifier contrastively to distinguish between models that produced a text. Such signals could be important for the classification model as they highlight texts produced by particular models.

Declaration on Generative AI

During the preparation of this work, the author(s) used Deepl in order to: 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, D. Dementieva, M. Fröbe, B. Gipp, A. Greiner-Petter, J. Karlgren, M. Mayerl, P. Nakov, A. Panchenko, M. Potthast, A. Shelmanov, E. Stamatatos, B. Stein, Y. Wang, M. Wiegmann, E. Zangerle, Overview of PAN 2025: Voight-Kampff Generative AI Detection, Multilingual Text Detoxification, Multi-Author Writing Style Analysis, and Generative Plagiarism Detection, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 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. N. Ta, K. A. Elozeiri, T. Gu, R. V. 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: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[4] 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, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[5] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, J. Ferret, P. Liu, P. Tafti, A. Friesen, M. Casbon, S. Ramos, R. Kumar, C. L. Lan, S. Jerome, A. Tsitsulin, N. Vieillard, P. Stanczyk, S. Girgin, N. Momchev, M. Hoffman, S. Thakoor, J.-B. Grill, B. Neyshabur, O. Bachem, A. Walton, A. Severyn, A. Parrish, A. Ahmad, A. Hutchison, A. Abdagic, A. Carl, A. Shen, A. Brock, A. Coenen, A. Laforge, A. Paterson, B. Bastian, B. Piot, B. Wu, B. Royal, C. Chen, C. Kumar, C. Perry, C. Welty, C. A. Choquette-Choo, D. Sinopalnikov, D. Weinberger, D. Vijaykumar, D. Rogozińska, D. Herbison, E. Bandy, E. Wang, E. Noland, E. Moreira, E. Senter, E. Eltyshev, F. Visin, G. Rasskin, G. Wei, G. Cameron, G. Martins, H. Hashemi, H. Klimczak-Plucińska, H. Batra, H. Dhand, I. Nardini, J. Mein, J. Zhou, J. Svensson, J. Stanway, J. Chan, J. P. Zhou, J. Carrasqueira, J. Iljazi, J. Becker, J. Fernandez, J. van Amersfoort, J. Gordon, J. Lipschultz, J. Newlan, J. yeong Ji, K. Mohamed, K. Badola, K. Black, K. Millican, K. McDonell, K. Nguyen, K. Sodhia, K. Greene, L. L. Sjoesund, L. Usui, L. Sifre, L. Heuermann, L. Lago, L. McNealus, L. B. Soares, L. Kilpatrick, L. Dixon, L. Martins, M. Reid, M. Singh, M. Iverson, M. Görner, M. Velloso, M. Wirth, M. Davidow, M. Miller, M. Rahtz, M. Watson, M. Risdal, M. Kazemi, M. Moynihan, M. Zhang, M. Kahng, M. Park, M. Rahman, M. Khatwani, N. Dao, N. Bardoliwalla, N. Devanathan, N. Dumai, N. Chauhan, O. Wahltinez, P. Botarda, P. Barnes, P. Barham, P. Michel, P. Jin, P. Georgiev, P. Culliton, P. Kuppala, R. Comanescu, R. Merhej, R. Jana, R. A. Rokni, R. Agarwal, R. Mullins, S. Saadat, S. M. Carthy, S. Cogan, S. Perrin, S. M. R. Arnold, S. Krause, S. Dai, S. Garg, S. Sheth, S. Ronstrom, S. Chan, T. Jordan, T. Yu, T. Eccles, T. Hennigan, T. Kocisky, T. Doshi, V. Jain, V. Yadav, V. Meshram, V. Dharmadhikari, W. Barkley, W. Wei, W. Ye, W. Han, W. Kwon, X. Xu, Z. Shen, Z. Gong, Z. Wei, V. Cotruta, P. Kirk, A. Rao, M. Giang, L. Peran, T. Warkentin,

E. Collins, J. Barral, Z. Ghahramani, R. Hadsell, D. Sculley, J. Banks, A. Dragan, S. Petrov, O. Vinyals, J. Dean, D. Hassabis, K. Kavukcuoglu, C. Farabet, E. Buchatskaya, S. Borgeaud, N. Fiedel, A. Joulin, K. Kenealy, R. Dadashi, A. Andreev, Gemma 2: Improving open language models at a practical size, 2024. URL: https://arxiv.org/abs/2408.00118. arXiv:2408.00118.

[6] M. Siino, I. Tinnirello, M. La Cascia, Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers, Information Systems 121 (2024) 102342. URL: https://www.sciencedirect.com/science/article/pii/S0306437923001783. doi:https://doi.org/10.1016/j.is.2023.102342.

[7] G. Mehak, A. Qasim, A. G. M. Meque, N. Hussain, G. Sidorov, A. Gelbukh, TechExperts(IPN) at GenAI detection task 1: Detecting AI-generated text in English and multilingual contexts, 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. 161–165. URL: https://aclanthology.org/2025.genaidetect-1.14/.

[8] N. H. Doan, K. Inui, Grape at GenAI detection task 1: Leveraging compact models and linguistic features for robust machine-generated text detection, 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. 209–217. URL: https://aclanthology.org/2025.genaidetect-1.22/.

[9] K. Kuznetsov, L. Kushnareva, P. Druzhinina, A. Razzhigaev, A. Voznyuk, I. Piontkovskaya, E. Burnaev, S. Barannikov, Feature-level insights into artificial text detection with sparse autoencoders, 2025. URL: https://arxiv.org/abs/2503.03601. arXiv:2503.03601.

[10] L. van der Maaten, G. E. Hinton, Visualizing high-dimensional data using t-sne, Journal of Machine Learning Research 9 (2008) 2579–2605.

[11] S. Liu, X. Liu, Y. Wang, Z. Cheng, C. Li, Z. Zhang, Y. Lan, C. Shen, Does DetectGPT fully utilize perturbation? bridging selective perturbation to fine-tuned contrastive learning detector would be better, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 1874–1889. URL: https://aclanthology.org/2024.acl-long.103/. doi:10.18653/v1/2024.acl-long.103.

[12] X. Guo, Y. He, S. Zhang, T. Zhang, W. Feng, H. Huang, C. Ma, Detective: Detecting AI-generated text via multi-level contrastive learning, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024. URL: https://openreview.net/forum?id=cdTTTJfJe3.

A. Fine-Tuning Hyperparameters

Table 5
Hyperparameters; The fine-tuning of the RoBERTa-base model was performed on a computing setup leveraging an NVIDIA GeForce RTX 3090 Ti GPU. The GPU was driven by NVIDIA driver with CUDA version 12.6. Gemma-2 2B was fine-tuned on 1xH100 GPU with CUDA version 12.8. The checkpoint_epoch corresponds to the last best epoch at which the highest weighted F1 score was achieved.

	RoBERTa-base	**Gemma-2 2B**
batch_size	64	16
learning_rate	6e-5	6e-5
weight_decay	0.01	0.01
num_epochs	10	10
max_length	512	512
warmup_steps	500	500
patience	5	5
checkpoint_epoch	6	4