AI-Monitors at GenAI Detection Task 1: Fast and Scalable Machine Generated Text Detection

Azad Singh[1], Vishnu Tripathi[1], Ravindra Kumar Pandey[1], Pragyanand Saho[1], Prakhar Joshi[1], Neel Mani[1], Richa Alagh[2], Pallaw Mishra[3], Piyush Arora[4]
Dev Sanskriti Vishwavidyalaya Haridwar[1], Graphic Era University [2],
Safexpress Private Limited[3], American Express AI Labs[4]
{azad.singh|vishnu.tripathi|ravindra.pandey}@aicentre.org
{pragyanand.saho|prakhar.joshi}@aicentre.org |neel.mani@dsvv.ac.in
richaalagh@gmail.com|pallaw.mishra@safexpress.com|piyush.arora1@aexp.com

Abstract

We describe the work carried out by our team, *AI-Monitors*, on the Binary Multilingual Machine-Generated Text Detection (Human vs. Machine) task at COLING 2025. This task aims to determine whether a given text is generated by a machine or authored by a human. We propose a lightweight, simple, and scalable approach using encoder models such as RoBERTa and XLM-R We provide an in-depth analysis based on our experiments. Our study found that carefully exploring fine-tuned parameters such as i) no. of training epochs, ii) maximum input size, iii) handling class imbalance etc., plays an important role in building an effective system to achieve good results and can significantly impact the underlying tasks. We found the optimum setting of these parameters can lead to a difference of about 5-6% in absolute terms for measure such as accuracy and F1 measure. The paper presents crucial insights into optimal parameter selection for fine-tuning RoBERTa and XLM-R based models to detect whether a given text is generated by a machine or a human.

1 Introduction

Large language models (LLMs) like GPT-4, Claude 3.5, and Gemini 1.5-pro have rapidly become mainstream tools, offering highly fluent and articulate text generation across a wide range of applications, from social media and news to academic and educational content. These models are capable of producing human-like responses to various queries, making them increasingly attractive for replacing human labor in tasks such as content creation, customer support, and even academic writing.

This challenge (Wang et al., 2025) underscores the need for automated systems designed to detect machine-generated content. As LLMs become more sophisticated and pervasive, developing robust detection methods is critical to preventing misuse. These systems could help mitigate the risks of misinformation, ensure academic integrity, and provide safeguards against the over-reliance on machine-generated material in sensitive contexts.

This work is part of a larger project where we are building solutions for emerging plagiarism detection, LLM-based response generation detection for academic studies, and assignments. One major issue is the difficulty humans face in distinguishing machine-generated text from human-written content. This has posed significant challenges when analyzing and evaluating student assignments and open-book answers, where there are potential issues with using LLM-generated answers and challenges in accurately detecting them. This problem calls for the development of effective automated systems for grading, assessment, and identifying whether a text is generated by a human or a machine. Additionally, it is important to identify instances where machine-generated content has been post-edited by humans to avoid detection by automatic systems.

In this paper we describe the work carried out by our team *AI-Monitors* on the Binary Multilingual Machine-Generated Text Detection (Human vs. Machine) task at COLING 2025 (Wang et al., 2025). Towards building a fast scalable system which can auto-train and learn with more data, we focused on exploring RoBERTa and XLM-R models as they have shown to perform well for this task (Wang et al., 2024).

Main Contributions:

- In our study, we explore different fine-tuning parameters, such as *training epochs, base model, maximum input size, and how to handle data imbalance*.
- The paper presents crucial insights into optimal parameter selection for fine-tuning RoBERTa and XLM-R based models to detect whether a given text is generated by a machine or a human.

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 230–235
January 19, 2025. ©2025 International Conference on Computational Linguistics

- We are releasing our code on GitHub [1] so that this work can be expanded and used by other teams to develop more effective solutions

The paper is organized as follows: Section 2 presents related work on automatic detection of machine generated text. Section 3 provides an overview of the architectures that we explore in this work. Section 4 presents the data and tools details, Section 5 showcases our results. Section 6 talks about our lessons learned and Section 7 presents conclusion and future work.

2 Related Work

Automatic detection of machine-generated text is typically framed as a binary classification task, where, for a given input, we must classify whether it is generated by a machine or a human (Wang et al., 2024, 2025). Two main approaches are commonly used: one relies on supervised techniques, which require large training datasets, while the other relies on unsupervised approaches using common detection models such as RoBERTa (Liu et al., 2019), XLM-R (Conneau et al., 2020; Goyal et al., 2021), or stylistic features. Jawahar et al. (2020) provides a critical survey of the pros and cons of various alternatives for detecting machine-generated text.

Given sufficient training data, some prior works (Solaiman et al., 2019; Fagni et al., 2021) have experimented with fine-tuning the RoBERTa language model for the detection task. Solaiman et al. (2019) found that RoBERTa established state-of-the-art performance in identifying web pages generated by the largest GPT-2 model, achieving an accuracy of approximately 95%. Fagni et al. (2021) demonstrated that the RoBERTa detector also set the state-of-the-art in accurately distinguishing machine-generated tweets from human-written tweets, outperforming both traditional machine learning models (e.g., bag-of-words) and complex neural network models (e.g., RNN, CNN) by a large margin. This promising result suggests that the RoBERTa detector can generalize well to previously unseen publication sources, such as Twitter.

The automatic detection systems comprise approaches that are used to detect domains including the one that USTC-BUPT has developed for SemEval-2024 Task 8 with the help of DATeD, LLAM, TLE, and AuDM for monolingual as well as multilingual detection targets (Guo et al., 2024). Further, black-box machine-generated text detection where LLMs are fine-tuned with parameter-efficient smaller LLMs and per-language classification-threshold calibration was proposed and was proved to perform well in SemEval-2024 Task 8 (Spiegel and Macko, 2024). Another valuable contribution is Genaios' LLM IXTIC system in which several LLaMA-2 models operate within a Transformer Encoder framework; this system performed extremely well in the monolingual track and underlines the role of token-level probabilistic features for text classification (Sarvazyan et al., 2024).

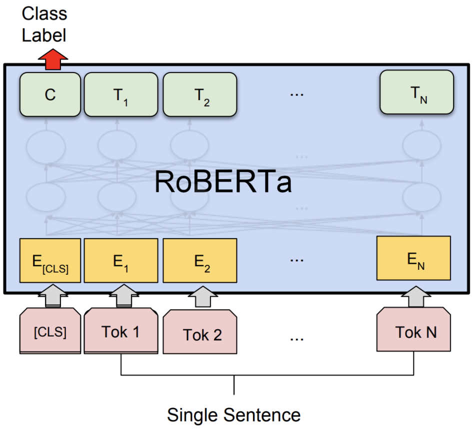

Figure 1: RoBERTa model

3 Methodology

In the task of distinguishing between human-written and machine-generated text, we are performing finetuning of RoBERTa and XLM-R. RoBERTa (Liu et al., 2019), is a robust transformer-based language model, adapted to better capture the nuances in writing style and language patterns that differentiate human and machine-generated content. The model is trained on a labeled dataset consisting of both machine-generated and human-written articles to learn these distinctions. Figure 1 shows an illustration of the RoBERTa model.

To build effective and efficient solution we explore different parameters as shown in Table 1. We conduct following experiments to determine the optimum settings in terms of i) no. of training epochs, ii) maximum input size, iii) handling class imbalance, and iv) selection of base model.

[1] https://github.com/aimonitors25/machine-generated-text-detection

Parameter type	Parameter Values
Base Model	[RoBERTa, XLM-R]
Fine-tuning Epochs	[1 to 5]
Max Input Token Size	[128, 256]
Weighting parameters (human: machine)	[1:1, 2:1, 3:2]

Table 1: Investigative parameters - experimental setup

4 Dataset and Tools Used

The dataset is divided into three primary subsets: Training Data, Development (Dev) Data, and Test Data. Each of these subsets contains different amounts of data, categorized into two main classes: *Machine* and *Human* as shown in Table 2

Stats	Train Data	Dev Data	Test Data
Total	610,765	261,758	73,941
Machine	381,843	163,430	39,266
Human	228,922	98,328	34,675

Table 2: Data Statistics

There is a noticeable class imbalance in the training and development sets, with the Machine class significantly outnumbering the Human class. This could lead to challenges in model training, where the model might become biased toward the majority class (Machine). Hence in our experiments we explored weighing the minority class more as compared to equal weights for both the classes. The test set is relatively more balanced between the two classes, which is important for evaluating the model's ability to generalize to both Machine and Human instances.

Table 3 provide details on the length of the input text across train, dev and test set, where we report common metrics such as count, mean, std, min, max and percentile based no. of input tokens. As the mean length is around 250 and 50% no. of tokens are less than 300, hence we select two values for our investigation for max input tokens size 128 and 256.

5 Results

We performed multiple parameters explorations as described in Table 1, Below we discuss the results of our investigations and learning from same.

RoBERTA vs XLM-R: We tested two models—Roberta and XLM-R—and evaluated their performance over different epochs the results are summarized in the table 4.

Length stats	Train set	Dev set	Test set
count	610,765	261,758	73,941
mean	244.53	244.87	295.43
std	235.08	235.31	185.98
min	1	1	1
25%	91	91	171
50%	186	187	296
75%	320	320	396
max	4,752	2,916	10,743

Table 3: Input text length for train, dev and test set

Models	Epoch	Dev Set	Test Set
Roberta	1	86.61	**73.78**
XLM-R	1	92.68	71.79
Roberta	3	96.05	71.79
XLM-R	3	94.16	72.40
Roberta	5	**97.82**	72.64
XLM-R	5	95.18	72.50

Table 4: Accuracy results on the dev and test set using max input tokens=128

The XLM-R model performed better in the dev set, achieving an accuracy of 92.68% after 1 epoch, compared to 86.61% for RoBERTa. However, the RoBERTa model saw greater improvements with more training, reaching 96.05% and then 97.82% accuracy on the dev set after 3 and 5 epochs, respectively. Thus, we used RoBERTa for further explorations of optimum parameters. This best solution was also the submission of the official leader board for the shared task. More details to follow in the later section.

RoBERTa finetuning - max input size explorations: Table 5 presents the results in the dev and test data set to determine the optimum no. of the maximum input size. As described in Section 4, we explored max input size=128, 256. We found the results are quite better with max input size =256, accuracy results on dev set are quite similar, but we see quite some boost in the test set while using no. of tokens as 256. We used these settings for further explorations.

RoBERTa finetuning - no. of epochs: Table 6 represents the results of the fine-tuning of RoBERTa with the maximum input token size of 256, across different epochs. We find that model

RoBERTa	Input Size	Dev Set	Test Set
Epoch-1	128	86.61	73.78
Epoch-2	128	95.97	71.80
Epoch-3	128	96.05	71.79
Epoch-1	256	94.40	72.40
Epoch-2	256	95.97	74.21
Epoch-3	**256**	**96.72**	**74.91**

Table 5: Accuracy results on the dev and test set

learning is becoming saturated and loss becoming static. Thus, results are coming similar post Epoch-3. The model is trained with a batch size of 32. We used Adam optimized with learning rate as $2e-5$, and weight_decay as 0.01.

Epoch	Dev Set	Test Set
Epoch-1	94.40	72.40
Epoch-2	95.97	74.21
Epoch-3	**96.72**	**74.91**
Epoch-4	**96.72**	**74.91**

Table 6: RoBERTa max input size=256, accuracy results on the dev and test set

RoBERTa finetuning - handling class imbalance: As discussed in Section 4, train dataset is quite imbalanced, thus to effectively learn signals we tried exploring weighing the minority class more. Table 7 presents the results of different weighing scores to handle class imbalance. The results across the dev and test sets are quite close and vary. We see the best test set performance on model with weights (2.0:1.0) w.r.t, (human:machine). However for dev set best performance is obtained without weighing the classes differently.

	Weight 1:1		Weight 2:1		Weight 3:2	
Epoch	Dev	Test	Dev	Test	Dev	Test
1	94.40	72.40	94.23	72.1	95.16	69.98
2	95.97	74.21	95.92	**74.99**	95.46	71.13
3	**96.72**	74.91	95.76	73.13	95.88	72.06

Table 7: RoBERTa max input size=256, accuracy results on the dev and test set, handling class imbalance

Official Solution on the benchmarking leaderboard: Table 8 presents the results of our official submission to the shared task. This submission is based on a finetuned RoBERTa model, using maximum input size as 128 tokens and no. separate weights for class imbalance. This combination gave the best results on the dev set as shown in Table 4. These settings seems to be not the optimum as reviewed with other experiments that we performed post the task deadline.

Models	F1-Macro	F1-Micro
Best System	**83.07**	**83.11**
Baseline	73.42	73.81
Our Submission	70.57	72.64

Table 8: Results on the blind test set, F1-Micro represents Accuracy

6 Lessons Learned

1. Our initial experiments were over-fitted on the majority class, and hence, our understanding of whether the solution is generic was limited due to insufficient testing of different parameter settings and configurations, as discussed in Table 1
2. There is a need for carefully examining the choice of base models, parameters and settings.
3. Overall, the analysis and investigation after the official submission, using the test and development sets, indicate that a better understanding of various factors, such as training epochs, base model, maximum input size, and how to handle data imbalance, can lead to an improvement of about 5-6% in the metric scores, as shown in this paper.
4. We plan to continue these explorations as part of a larger project where we are working towards a general solution for handling plagiarism detection and LLM-generated text detection in academic settings.

7 Conclusion & Future Work

A critical survey on automatic text detection (Jawahar et al., 2020) provides a summary of key error categories made by these automated models, namely: *fluency, brevity, factuality, spurious entries, contradictions, repetitions, common sense reasoning, typos, grammatical errors etc,.* We plan to perform a similar error analysis on this task dataset and work towards building a hybrid pipeline that leverages techniques like those in (Sarvazyan et al., 2024). A summary of this pipeline on leveraging transformer encoder that incorporates token-level probabilistic features extracted from the Llama models is shown in Figure 2 and discussed in Appendix A. In future, we aim to explore ensemble-based solutions, comprising a simple fine-tuned RoBERTa pipeline alongside a richer pipeline as described in Appendix A, that leverages multiple LLMs to better capture patterns and data distributions for detecting machine generated text.

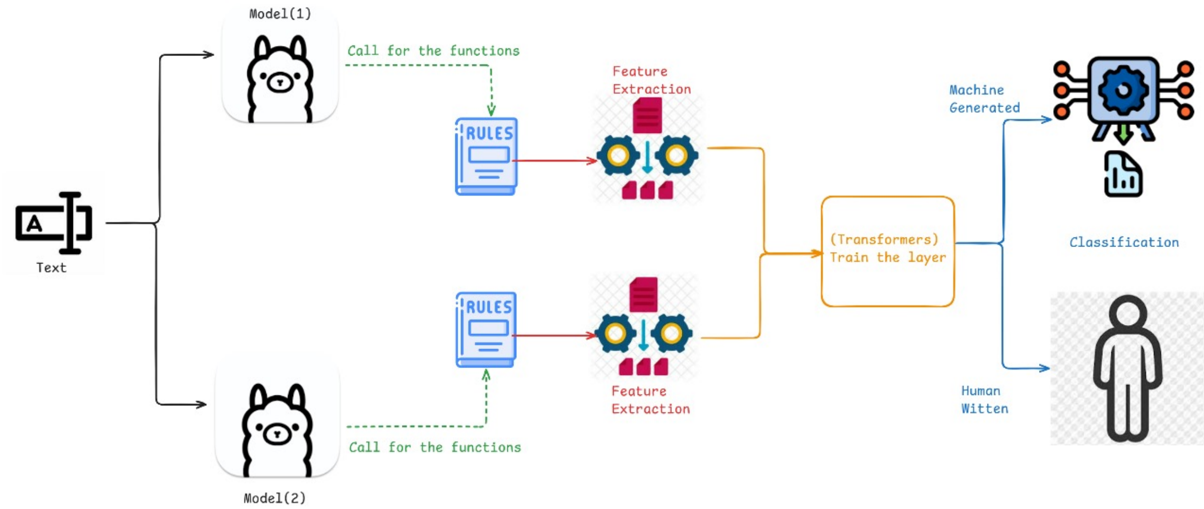

Figure 2: Transformer Encoder Architecture with Llama Model for Extracting Statistical Features from Text

Acknowledgments

We would like to thank the Task Organizers for organizing such an interesting and relevant benchmarking task.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. *arXiv preprint arXiv:2105.00572*.

Zikang Guo, Kaijie Jiao, Xingyu Yao, Yuning Wan, Haoran Li, Benfeng Xu, Licheng Zhang, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2024. Ustc-bupt at semeval-2024 task 8: Enhancing machine-generated text detection via domain adversarial neural networks and llm embeddings. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1511–1522.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Areg Mikael Sarvazyan, José-Ángel González, and Marc Franco-Salvador. 2024. Genaios at semeval-2024 task 8: Detecting machine-generated text by mixing language model probabilistic features. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 101–107.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Michal Spiegel and Dominik Macko. 2024. Kinit at semeval-2024 task 8: Fine-tuned llms for multilingual machine-generated text detection. *arXiv preprint arXiv:2402.13671*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. *arXiv preprint arXiv:2404.14183*.

Yuxia Wang, Artem Shelmanov, Jonibek Mansurov, Akim Tsvigun, Vladislav Mikhailov, Rui Xing, Zhuohan Xie, Jiahui Geng, Giovanni Puccetti, Ekaterina Artemova, Jinyan Su, Minh Ngoc Ta, Mervat Abassy, Kareem Elozeiri, Saad El Dine Ahmed, Maiya Goloburda, Tarek Mahmoud, Raj Vardhan Tomar, Alexander Aziz, Nurkhan Laiyk, Osama Mohammed Afzal, Ryuto Koike, Masahiro Kaneko, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2025. GenAI content detection task 1: English and multilingual machine-generated text detection: AI vs. human. In *Proceedings of the 1st*

Workshop on GenAI Content Detection (GenAIDetect), Abu Dhabi, UAE. International Conference on Computational Linguistics.

Appendix A

Transformer Encoder with Llama to extract statistical Features: Motivated by (Sarvazyan et al., 2024) we explored alternative approach that leverages a transformer encoder and incorporates token-level probabilistic features extracted from the Llama models as shown in Figure 2. The features used for each token in a given text are: i) the log probability of the observed token, ii) the log probability of the predicted token, iii) the entropy of the token distribution, iv) the rank of the observed token, v) the log rank and vi) the LLM-Deviation.

These features are designed to capture the statistical "style" of machine-generated text (MGT) in a precise manner. The log probabilities provide insight into how confidently the model predicts each token. At the same time, the entropy captures the unpredictability or randomness in the generation process, and the Rank and Log Rank are also noted by the model in terms of tokens where the lower Rank represents higher confidence in the correct token. LLM-Deviation assesses the variance of the model outputs from a uniform distribution reflecting higher structure in the MGT model's outputs. These probabilistic measures are particularly useful for distinguishing between human writing, which tends to be more diverse and unpredictable, and machine-generated text, which often follows more structured patterns.