Random at GenAI Detection Task 3: A Hybrid Approach to Cross-Domain Detection of Machine-Generated Text with Adversarial Attack Mitigation

Shifali Agrahari
Department of CSE
Indian Institute of
Technology Guwahati
a.shifal@iitg.ac.in

Prabhat Mishra
Observe.AI
Bengaluru, India
mrprabhat010@gmail.com

Sujit Kumar
Nanyang Technological
University, Singapore
kumar.sujit474@gmail.com

Abstract

Machine-generated text (MGT) detection has gained critical importance in the era of large language models, especially for maintaining trust in multilingual and cross-domain applications. This paper presents Task 3 Subtask B: **Adversarial Cross-Domain MGT Detection** for in the COLING 2025 DAIGenC Workshop. Task 3 emphasizes the complexity of detecting AI-generated text across eight domains, eleven generative models, and four decoding strategies, with an added challenge of adversarial manipulation. We propose a robust detection framework transformer embeddings utilizing Domain-Adversarial Neural Networks (DANN) to address domain variability and adversarial robustness. Our model demonstrates strong performance in identifying AI-generated text under adversarial conditions while highlighting condition scope of future improvement.

1 Introduction

2 Introduction

The advent of large-scale generative language models, such as GPT (Generative Pre-trained Transformer) (Liang et al., 2023) based systems, has revolutionized text generation by producing outputs that closely mimic human writing. These advancements have significantly impacted various fields, including content creation, education, and customer service, by providing highly coherent and contextually relevant text. However, this progress has also introduced new challenges, particularly in detecting MGT across diverse domains and languages. The widespread use of these models has raised concerns in areas like academic integrity, misinformation, and malicious use of AI-generated content.

While there are existing models result mention in Raid[1], still struggle with cross-domain issues and adversarial attacks (manipulations of input data to fool machine learning models).

To tackle these issues, The COLING 2025 Workshop on DAIGenC[2] (Dugan et al., 2025) *"Task 3: Cross-domain Machine-Generated Text Detection"* binary problem formulation as Task 1 however the texts will come from 8 different domains, 11 generative models, and four decoding strategies. This challenge is divided into two sub-tasks: Non-Adversarial and Adversarial. We solve one subtask, Subtask B, which is Adversarial Cross-Domain MGT detection.

Our contributions include a pipeline that integrates XLM-RoBERTa (Liu, 2019) embeddings for enhanced text representation, domain adaptation using Domain-Adversarial Neural Networks (DANN) to minimize domain-specific biases and improve generalization across diverse text domains, and adversarial robustness through incorporating adversarial attack classification to detect and mitigate manipulative techniques. Additionally, we focus on label prediction for improved model accuracy. Experimental results demonstrate strong detection performance, particularly in the same domain and against zero-width space attacks. Our findings from experimental results highlight the strengths of the proposed approach while identifying areas for improvement, such as achieving consistent performance across all domains and all attack types. These results emphasize the importance of balanced datasets, adaptive techniques, and comprehensive evaluation to advance the field of MGT detection.

3 Background

Over the last few years, numerous approaches have been proposed to tackle the task of AI-generated text detection. Detecting machine-generated text

[1] https://gitlab.com/genai-content-detection/genai-content-detection-coling-2025

[2] https://github.com/liamdugan/COLING-2025-Workshop-on-MGT-Detection-Task-3

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 365–370
January 19, 2025. ©2025 International Conference on Computational Linguistics

is primarily formulated as a binary classification task (Zellers et al., 2019; Gehrmann et al., 2019; Ippolito et al., 2019), naively distinguishing between human-written and machine-generated text. In general, there are three main approaches: the supervised methods (Wang et al., 2023; Uchendu et al., 2021; Zellers et al., 2019; Zhong et al., 2020; Liu et al., 2023, 2022), the unsupervised ones, such as zero-shot methods (Solaiman et al., 2019; Ippolito et al., 2019; Mitchell et al., 2023; Su et al., 2023; Hans et al.; Shijaku and Canhasi, 2023), and Adversarial measures on detection accuracy (Susnjak and McIntosh, 2024; Liang et al., 2023), especially within the education domain. For example, (Antoun et al., 2023) evaluates the robustness of detectors against character-level perturbations or misspelled words, focusing on French as a case study. (Krishna et al., 2024) train a generative model (DIPPER) to paraphrase paragraphs to evade detection. Although supervised approaches yield relatively better results, they are susceptible to overfitting (Mitchell et al., 2023; Su et al., 2023).

There are some techniques like feature-based, fusion, and ensemble methods Neural based or finetuned (Solaiman et al., 2019; Kumarage et al., 2023b; Shah et al., 2023; Mindner et al., 2023; **?**). Therefore, researchers combine statistics-based and deep learning-based techniques to gain adversarial robustness and high performance (Kushnareva et al., 2021; Crothers et al., 2022; Uchendu et al., 2023).Some studies attempt to address the challenges of cross-domain detection and adversarial attacks (Krishna et al., 2024; Kumarage et al., 2023a).

4 System Overview

We present our proposed **Adversarial Cross-Domain MGT Detection** which combines adaptability across diverse domains, attacks and generative models.

4.1 Data

For Task 3, we used the dataset provided by the shared task organizers (Dugan et al., 2025), which consists of a training set with 5,615,820 rows and a test set with 672,000 rows. The training dataset includes the following features: `id`, `adv_source_id`, `source_id`, `model`, `decoding`, `repetition_penalty`, `attack`, `domain`, `title`, `prompt`, and `generation`. Table 1 provides the unique counts for each feature in the training set. The diversity of these features indicates a wide range of model outputs, domains, and text variations, making the dataset well-suited for evaluating model performance in multilingual MGT detection. The test dataset, used only for evaluation, includes the `id` and `generation` fields.

Feature	**Unique Count**
`id`	5,615,820
`adv_source_id`	467,985
`source_id`	13,371
`model`	12
`decoding`	2
`repetition_penalty`	2
`attack`	12
`domain`	8
`title`	13,221
`prompt`	26,500
`generation`	4,975,574

Table 1: Unique Counts of Training Dataset Features

4.2 Methodology

4.2.1 Data Preprocessing and Feature Engineering

Use `Tokenizer` (Liu, 2019) to represent text (e.g., `generation`, `title`, `prompt`). Let $X \in \mathbb{R}^{m\times n}$ be the matrix of tokenized sequences, where m is the number of samples and n is the maximum sequence length. Encode categorical features (`domain`, `model`, `decoding`, etc.) using embeddings as $E_{\text{domain}}, E_{\text{model}}, E_{\text{decoding}} \in \mathbb{R}^d$ where d is the embedding dimension. Concatenate embeddings to form a vector representation for each text instance as defined $\mathbf{x}_{\text{features}} = [E_{\text{domain}}, E_{\text{model}}, E_{\text{attack}}, \ldots]$. Final input representation for each sample:

$$\mathbf{x}_{\text{input}} = [X; \mathbf{x}_{\text{features}}] \in \mathbb{R}^{m\times(n+d)} \tag{1}$$

first Feature Extraction Using transformer encoders as G_f to learn domain-invariant representations Eq. 2

$$\mathbf{h} = G_f(\mathbf{x}_{\text{input}}) \in \mathbb{R}^k \tag{2}$$

where k is the latent representation dimensionality.

4.2.2 Domain Adaptation

Domain adaptation refers to the process of enabling machine learning models to generalize well on a target domain that differs from the source domain on which the model was initially trained. To address this, domain adaptation techniques aim to minimize

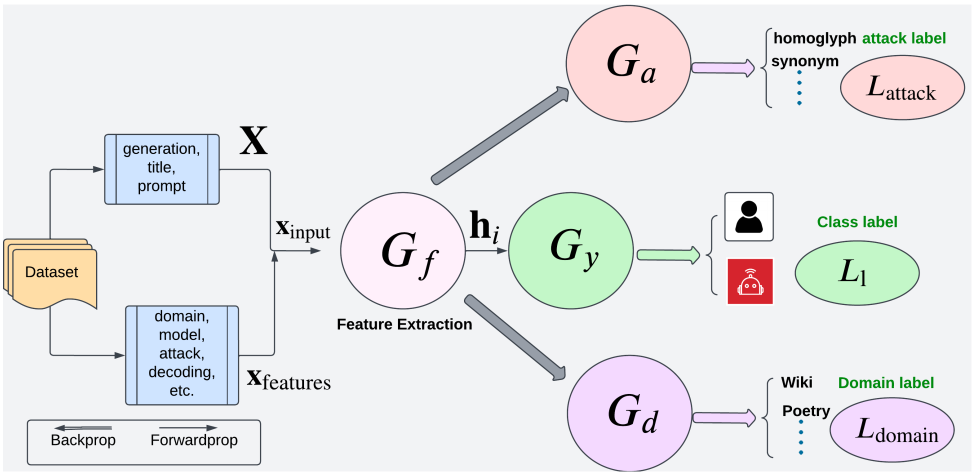

Figure 1: Proposed Detector model architecture: fusing stylometric features with a PLM embedding.

the discrepancy between source and target domain distributions by learning representations that are invariant to domain-specific characteristics while retaining task-relevant features. We use Domain-Adversarial Neural Networks (DANN) to improve cross-domain robustness.

Calculate the Adversarial Loss for Domain Classifier. The domain classifier G_d aims to predict the domain D of the input text. We apply a gradient reversal layer with the following objective:

$$\mathcal{L}_{\text{domain}} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{|\text{domains}|} d_{i,j}\log G_d(\mathbf{h}_i) \quad (3)$$

where $d_{i,j}$ is the true domain label.

4.2.3 Adversarial Attack Classifer for Robuestness

To classify attack types, we introduce an attack classifier G_a, which predicts the specific attack type (e.g., homography, whitespace). The attack types are encoded as categorical labels, and G_a outputs probabilities for each attack type. The cross-entropy loss for attack classification is defined as:

$$\mathcal{L}_{\text{attack}} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{|\text{attacks}|} a_{i,j}\log G_a(\mathbf{h}_i) \quad (4)$$

where $a_{i,j}$ is the true attack label.

4.2.4 Label Classifier and Loss Functions

The MGT classifier G_y is trained to predict whether text is human-Written or machine-generated, using binary cross-entropy (BCE) loss:

$$\mathcal{L}_l = -\frac{1}{m}\sum_{i=1}^{m}(y_i\log(G_y(\mathbf{h}_i)) + (1-y_i)\log(1-G_y(\mathbf{h}_i))) \quad (5)$$

4.2.5 Final Model

The feature extractor G_f is a transformer-based encoder **XLM-RoBERTa** (Liu, 2019) that processes tokenized text and generates domain-agnostic latent representations, capturing high-level semantic information. It produces embedding vector, which is passed through two fully connected (FC) layers.

Each of these FC layers is followed by an activation function such as *ReLU* to introduce non-linearity. The final output layer for the Domain Classifier G_d and Attack Classifier G_a uses the softmax activation to generate a probability distribution over the respective classes. For the Label Classifier G_y, the output layer uses a sigmoid activation function, producing a probability score. The model is trained using a combination of binary cross-entropy loss for classification, adversarial loss for domain adaptation, and attack classification loss. During training, we perform backpropagation to update the weights of the feature extractor, domain classifier, attack classifier, and MGT classifier. The optimizer minimizes the total loss as Equation 6, and we monitor the performance calculating probability that a given text is predicted to be machine-generated.. Early stopping is used to prevent overfitting.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_l + \alpha\cdot\mathcal{L}_{\text{domain}} + \gamma\cdot\mathcal{L}_{\text{attack}} \quad (6)$$

where α and γ control the contributions of attack classification losses. During testing, the trained model is evaluated on a test set, where it provides the probability that a given text is predicted to be machine-generated.

Model	Config 1	Config 2	Config 3	Config 4	Config 5	Config 6
chatgpt	0.361	0.364	0.388	0.319	0.306	0.333
gpt4	0.330	0.325	–	0.300	0.295	–
gpt3	0.307	0.335	–	0.295	0.300	–
gpt2	0.360	0.365	–	0.315	0.285	–
mistral	0.323	0.313	0.340	0.282	0.245	0.295
mistral-chat	0.369	0.378	0.390	0.329	0.325	0.300
cohere	0.386	0.390	0.415	0.359	0.365	0.410
cohere-chat	0.370	0.370	–	0.352	0.335	–
llama-chat	0.372	0.385	–	0.285	0.275	–
mpt	0.350	0.365	0.375	0.305	0.287	0.305
mpt-chat	0.384	0.385	**0.440**	0.369	0.287	0.310

Table 2: Performance metrics for adversarial cross-domain MGT detection across various configurations. **Key:** Config 1: Wiki + All Decoding Strategies + All Repetition Penalties + Zero-width Space, Config 2: Wiki + Greedy Decoding Strategy + All Repetition Penalties + Zero-width Space, Config 3: Wiki + Greedy Decoding Strategy + Yes Repetition Penalty + Zero-width Space, Config 4: Wiki + All Decoding Strategies + All Repetition Penalties + Homoglyph, Config 5: Wiki + Greedy Decoding Strategy + All Repetition Penalties + Homoglyph, Config 6: Wiki + Greedy Decoding Strategy + Yes Repetition Penalty + Homoglyph.

5 Experimental Setup

We utilize the dataset mentioned in Section 4.1 Further details of experimental setup and hyperparameters are presented in the Appendix section A.1. Our model was implemented using PyTorch.

6 Results

As mentioned in Table 2, Our detector performs best on the Wikipedia domain but struggles with other domains. We analyze the data set, revealing that the average generation length in the Wikipedia domain is higher than in different domains, as illustrated in Fig. 2. Similarly, our detector performs better against two adversarial attacks—Homoglyph and Zero-Width Space. However, it struggles to detect text affected by other adversarial attacks. Fig. 3 highlights that Zero-Width Space attacks produce the most extended average text length compared to different attack types.

Among the generated model texts, our detector most effectively identifies the text generated by *mpt-chat*. Further analyses, as shown in Fig. 4 and Fig. 5, indicate that the detector is particularly effective for the Wikipedia domain and the *mpt-chat* model.

In summary, our detector demonstrates optimal performance in the Wikipedia domain, with the greedy decoding strategy, the repetition penalty enabled, the zero-width space adversarial attacks, and text generated by the *mptchat* model. All details of results are mentioned in the Raid cooling shared task leaderboard.

7 Conclusion

This paper introduces a robust approach to **Adversarial Cross-Domain MGT Detection**, leveraging transformer embeddings and domain adaptation to address the challenges of domain variability and adversarial robustness. The proposed architecture, based on Domain-Adversarial Neural Networks, demonstrates strong performance, particularly in detecting machine-generated text from specific domains like Wikipedia and against attacks like homoglyphs and zero-width space manipulations. but still fail to generalization. Future work should focus on curating more balanced datasets, enhancing model adaptability to diverse attack types, and exploring lightweight architectures for real-time applications. These steps are crucial to advance the reliability and scalability of MGT detection systems in multilingual and cross-domain scenarios.

References

Wissam Antoun, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah. 2023. Towards a robust detection of language model-generated text: Is chatgpt that easy to detect? In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1: travaux de recherche originaux–articles longs*, pages 14–27.

Evan Crothers, Nathalie Japkowicz, Herna Viktor, and

Paula Branco. 2022. Adversarial robustness of neural-statistical features in detection of generative transformers. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Liam Dugan, Andrew Zhu, Firoj Alam, Preslav Nakov, and Callison-Burch Chris. 2025. Genai content detection task 3: Cross-domain machine generated text detection challenge. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, Abu Dhabi, UAE. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

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

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.

Tharindu Kumarage, Amrita Bhattacharjee, Djordje Padejski, Kristy Roschke, Dan Gillmor, Scott Ruston, Huan Liu, and Joshua Garland. 2023a. J-guard: Journalism guided adversarially robust detection of ai-generated news. *arXiv preprint arXiv:2309.03164*.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023b. Stylometric detection of ai-generated text in twitter timelines. *arXiv preprint arXiv:2303.03697*.

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. Artificial text detection via examining the topology of attention maps. *arXiv preprint arXiv:2109.04825*.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7).

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2022. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *arXiv preprint arXiv:2212.10341*.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models. *arXiv preprint arXiv:2304.07666*.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. 2023. Classification of human-and ai-generated texts: Investigating features for chatgpt. In *International Conference on Artificial Intelligence in Education Technology*, pages 152–170. Springer.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Aditya Shah, Prateek Ranka, Urmi Dedhia, Shruti Prasad, Siddhi Muni, and Kiran Bhowmick. 2023. Detecting and unmasking ai-generated texts through explainable artificial intelligence using stylistic features. *International Journal of Advanced Computer Science and Applications*, 14(10).

Rexhep Shijaku and Ercan Canhasi. 2023. Chatgpt generated text detection. *Publisher: Unpublished*.

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*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Teo Susnjak and Timothy R McIntosh. 2024. Chatgpt: The end of online exam integrity? *Education Sciences*, 14(6):656.

Adaku Uchendu, Thai Le, and Dongwon Lee. 2023. Toproberta: Topology-aware authorship attribution of deepfake texts. *arXiv preprint arXiv:2309.12934*.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296*.

Zecong Wang, Jiaxi Cheng, Chen Cui, and Chenhao Yu. 2023. Implementing bert and fine-tuned roberta to detect ai generated news by chatgpt. *ArXiv*, abs/2306.07401.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. *arXiv preprint arXiv:2010.07475*.

A Example Appendix

A.1 Details of Experimental Setups

Table 3 lists key hyperparameters: learning rates (1×10^{-5} to 1×10^{-3}), batch sizes (16–64), epochs (50–200), and dropout rates (0.1–0.5), with some parameters experimentally fine-tuned.

Hyperparameter	**Typical Values**
Learning Rate (η)	$1e-5$ to $1e-3$
Batch Size	16, 32, 64
Number of Epochs	50 to 200
Dropout Rate	0.1 to 0.5
Embedding size	768
First FC & Second FC	512 & 256
Optimizer (Adam)	Beta1: 0.9, Beta2: 0.999, Epsilon: $1e-8$
Learning Rate	Scheduler

Table 3: List of Hyperparameters for the Experiment

A.2 Dataset Analysis Details

As mentioned in Fig. 2, 3, 4, 5, analysis of the distribution of columns over text or other columns.

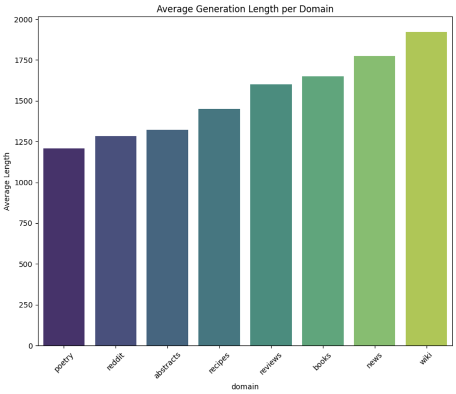

Figure 2: Average Generation Length per Domain

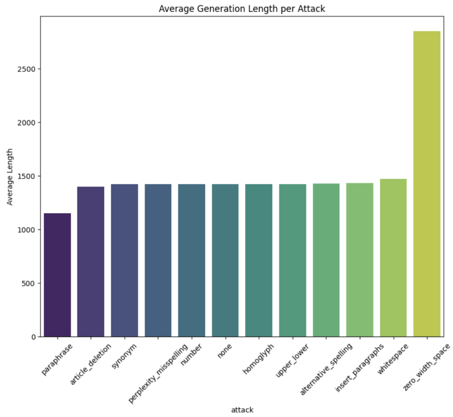

Figure 3: Average Generation Length per attack

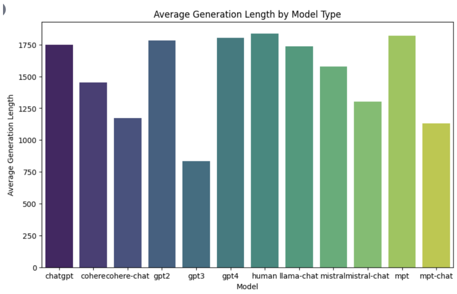

Figure 4: Average Generation Length per model type

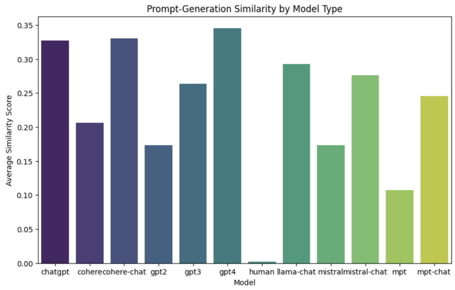

Figure 5: Average similarity score between prompt and text generated by each model.