DCBU at GenAI Detection Task 1: Enhancing Machine-Generated Text Detection with Semantic and Probabilistic Features

ZhaoWen Zhang[1]*, **Songhao Chen**[2]*, **Bingquan Liu**[1]†
[1]Harbin Institute of Technology
[2]Desktop Computing Business Unit, Lenovo
zhangzhaowen02@163.com, chensh8@lenovo.com, liubq@hit.edu.cn

Abstract

This paper presents our approach to the MGT Detection Task 1, which focuses on detecting AI-generated content. The objective of this task is to classify texts as either machine-generated or human-written. We participated in Subtask A, which concentrates on English-only texts. We utilized the RoBERTa model for semantic feature extraction and the LLaMA3 model for probabilistic feature analysis. By integrating these features, we aimed to enhance the system's classification accuracy. Our approach achieved strong results, with an F1 score of 0.7713 on Subtask A, ranking ninth among 36 teams. These results demonstrate the effectiveness of our feature integration strategy.

1 Introduction

In recent years, with the rapid development of large language models, distinguishing between machine-generated text and human-authored text has become increasingly challenging. This issue can lead to several potential problems. Low-quality generated text, when posted on social media, can reduce user experience, hinder the growth of platforms and high-quality content creators (Radivojevic et al., 2024). Generated text that lacks fact-checking can lead to the spread of rumors and misinformation (Chen and Shu, 2023), causing public panic and undermining government credibility. In academia, the presence of generated text raises ethical concerns regarding academic integrity (Meyer et al., 2023). Therefore, there is an urgent need to develop effective techniques for detecting machine-generated content (Wu et al., 2023).

Unlike typical machine-generated text, the data for this shared task are derived from multiple models and spans various domains (Wang et al., 2025). The human-authored texts in Subtask A originate from over 20 specialized fields, including finance, medicine, arXiv, WikiHow, IMDb, and Yelp. Correspondingly, the machine-generated texts are produced by more than 40 different large language models. Due to the diverse sources of this dataset, many simple yet effective statistical features are no longer viable, significantly increasing the challenge of the detection task.

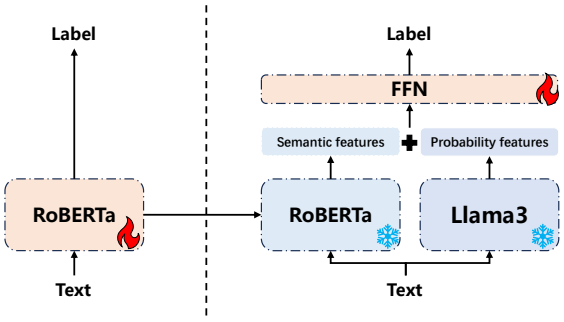

Figure 1: A two-stage machine generated text detection model architecture.

Machine-generated text often exhibits certain characteristics, such as weaker emotional expression, fewer numeric details, simpler grammar and vocabulary, and the absence of word order or spelling errors. However, these characteristics can be mitigated through iterative prompt optimization, which makes detection less reliable. To address this, we aim to develop a more generalized detection method that minimizes the risk of counter-detection. Since large models are pretrained on next-token prediction tasks, machine-generated text inherently exhibits high-probability characteristics. This feature remains consistent across texts generated by different models or under various prompt conditions. Specifically, we leverage the $[CLS]$ vector of the RoBERTa (Liu, 2019) model as the semantic feature of the text and use LLaMA3 (Dubey et al., 2024) model to calculate the difference between the probability of the actual next token and the predicted next token at each token

* Equal contribution.
† Corresponding author.

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 150–154
January 19, 2025.

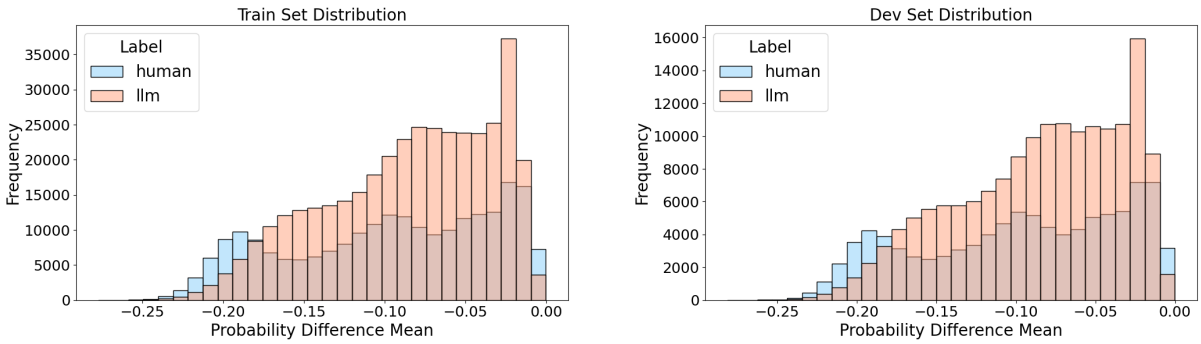

Figure 2: The x-axis represents the mean of the different dimensions of the probabilistic features H_p for each text, while the y-axis represents the number of texts with the same average value.

position, obtaining a vector as the probabilistic feature. By concatenating these two feature sets and feeding them into a feedforward network for binary classification, as illustrated in Figure 1, we achieve robust detection performance.

2 Related Works

The task of detecting machine-generated text is essentially a binary classification problem. Wu et al. (2023) provide a comprehensive overview of the field of LLM-generated text detection, thoroughly examining the necessity of this task. They categorize detection techniques into watermarking methods, statistical detectors, neural network-based detectors, and human-assisted approaches, and also list available data resources. Suvra Ghosal et al. (2023) conducted a similarly excellent review, focusing on the possibilities and limitations of text inspection. They categorize detection frameworks into a priori and post hoc detectors, as well as analyzing novel attack strategies for evading detection in machine-generated text. Due to challenges in achieving consistency and widespread adoption of watermarking methods, post hoc detection is currently the only feasible approach for real-world detection tasks. This approach is further divided into zero-shot detection and fine-tuned model detection, with the latter being the category of this shared task.

Zero-shot detection does not require labeled datasets. Typically, it involves calculating entropy, perplexity, n-gram frequency, or the average log probability per token of a given sequence, followed by thresholding. Mitchell et al. (2023) detect machine-generated text by examining the curvature of a language model's log probability function. They generate perturbations of a given text sample, compares their log probabilities with the original text, and identifies machine-generated text based on a higher discrepancy metric. Yang et al. (2023) detects machine-generated text by truncating a text in the middle, regenerating the remainder using a language model, and then analyzing n-gram differences between the original and newly generated text segments.

Fine-tuned model detection, on the other hand, trains binary classifiers using features extracted from pretrained language models. Petukhova et al. (2024) combine RoBERTa-base embeddings with diversity features and resample the training set. Verma et al. (2023) calculate the log probability of tokens using a series of weaker language models, generates additional synthetic features through vector and scalar operations, and uses a logistic regression classifier to detect machine-generated text based on these features.

3 Method

As shown in Figure 1, our model is divided into two stages. In the first stage, we perform supervised learning for binary classification using the RoBERTa model, aiming to enhance the $[CLS]$ vector of the RoBERTa model with features relevant to the task of detecting machine-generated text. In the second stage, we freeze the parameters of the RoBERTa model and obtain the $[CLS]$ vector for each text as the semantic feature H_s. For a given text $x = [x_1, \ldots, x_n]$, where n is the token length of the text, we freeze the parameters of the LLaMA3-8B-Instruct model and compute the probabilistic features $H_p = [h_1, \ldots, h_n]$, where h_i is calculated according to Equation 1:

$$h_i = p_\theta\left(x_{i+1}|x_{\leq i}\right) - \max_{y \in V} p_\theta\left(y|x_{\leq i}\right) \quad (1)$$

That is, under model θ, the probability of predicting the next token x_{i+1} given the prefix $x_{\leq i}$

is subtracted by the maximum probability of any token being predicted as the next token given the prefix $x_{\leq i}$. V represents the entire vocabulary.

For the different dimensions h_i of the probabilistic features H_p for the same text, we performed normalization, as shown in Equation 2:

$$h'_i = \frac{h_i - \min(H_p)}{\max(H_p) - \min(H_p)} \quad (2)$$

We compute the mean of the probabilistic features H_p for a text. The distribution of the probabilistic features mean is illustrated in Figure 2, where we can observe that machine-generated text tends to follow high-probability sampling for the next token, whereas human-authored text does not exhibit this distinct characteristic.

The semantic features H_s and probabilistic features H_p are first subjected to dimensionality reduction individually. These reduced vectors are then concatenated to form a unified representation. This concatenated representation is subsequently processed through a series of linear layers. Finally, a softmax activation function is applied to produce the final label predictions.

4 Experiments

As shown in Figure 3, the text lengths in the dataset are primarily concentrated around 500 words. In the first stage illustrated in Figure 1, we uniformly truncate texts to the first 512 tokens and experiment with four models: RoBERTa, RoBERTa-large, DeBERTa (He et al., 2021), and DeBERTa-large. We use the baseline script for training, with hyperparameters set as follows: a learning rate of 2e-5, batch size of 16, three epochs, and an L2 weight regularization of 0.01. On the validation set, RoBERTa-large achieved the best performance, with comparative results shown in Table 1.

	score	micro f1	accuracy
Baseline	0.8163	-	-
RoBERTa-large	0.8502	0.8571	0.8571
DeBERTa	0.8273	0.8378	0.8378
DeBERTa-large	0.8384	0.8439	0.8439
RoBERTa-large+LLaMA3	**0.8980**	**0.9015**	**0.9015**

Table 1: Performance Comparison of Models.

In the second stage, we select RoBERTa-large to extract the $[CLS]$ vector with a dimension of 1024. The text is again truncated to the first 512 tokens and input into LLaMA3-8B-Instruct to compute the probabilistic feature vector with a dimension of 512. We then train a feedforward neural network with three hidden layers and ReLU activation functions. The first layer reduces both features to 128 dimensions, which are then concatenated. The second layer further reduces the dimensionality to 64, and the final layer reduces it to 2 classes. We use a learning rate of 1e-4 and a dropout rate of 0.5. This approach achieves a macro F1 score of 0.8980 on the validation set. Our experiments were conducted using an NVIDIA GeForce RTX 4090 24GB.

	Llm	Human	Total
Train	381845	228922	610767
Dev	163430	98328	261758
Test	-	-	73941

Table 2: Statistics for datasets.

It is evident that using the same generative model as the text source for computing the probabilistic features in the second stage would yield better results. However, on the one hand, the dataset for the competition does not originate from a single model, and on the other hand, in real-world scenarios, we cannot know the potential model source of the text. We chose to use LLaMA3-8B-Instruct for computing the probabilistic features because the LLaMA series models have had a significant influence in the open-source model domain. Many subsequent open-source models have been affected by it and may have been trained on the same general datasets, leading to similar probability distributions in text generation. Additionally, LLaMA3-8B-Instruct performs exceptionally well in the English domain. Due to the large scale of the competition dataset and our limited computational resources, we did not conduct comparative experiments using other large models for probabilistic feature extraction.Table 2 presents the scale of the dataset.

Although we did not participate in the final submission for Subtask B, we conducted experiments on the validation set for this subtask. We used a combination of XLM-RoBERTa (Conneau, 2019) and LLaMA3-8B-Instruct, achieving a score of 0.6766 compared to the baseline of 0.6546 for Subtask B. This result suggests that probabilistic features can be helpful for detecting multilingual text, but the current model framework does not perform outstandingly.

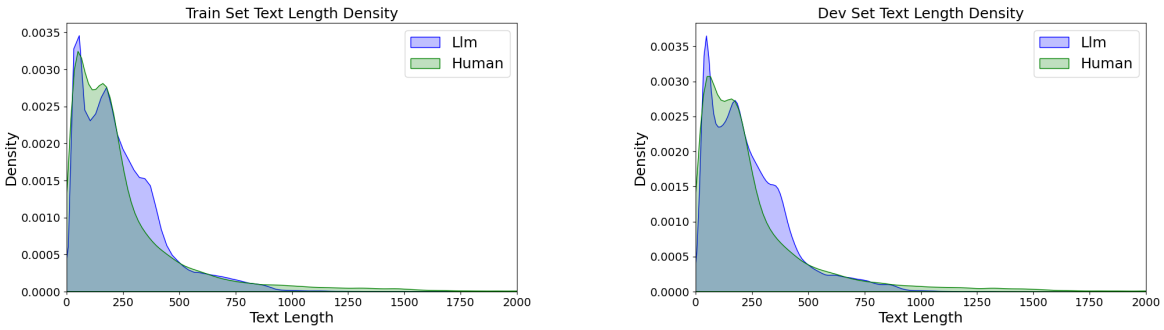

Figure 3: The x-axis represents the text length segmented by words, while the y-axis shows the probability density corresponding to each length. It can be observed that the text lengths in both the training and validation sets are primarily concentrated within 500 words.

5 Conclusion

In this work, we proposed a two-stage detection system for machine-generated text. By integrating semantic features from RoBERTa with probabilistic features from LLaMA3, our system achieves a Macro F1 score of 0.7713 on the test set, ranking ninth overall. Our experiments confirmed the effectiveness and generalizability of this feature integration approach. Compared to average results, our proposed system demonstrates robustness and strong generalization capability, which we aim to further enhance in future work.

References

Canyu Chen and Kai Shu. 2023. Can llm-generated misinformation be detected? *arXiv preprint arXiv:2309.13788*.

A Conneau. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

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

Jesse G Meyer, Ryan J Urbanowicz, Patrick CN Martin, Karen O'Connor, Ruowang Li, Pei-Chen Peng, Tiffani J Bright, Nicholas Tatonetti, Kyoung Jae Won, Graciela Gonzalez-Hernandez, et al. 2023. Chatgpt and large language models in academia: opportunities and challenges. *BioData Mining*, 16(1):20.

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.

Kseniia Petukhova, Roman Kazakov, and Ekaterina Kochmar. 2024. Petkaz at semeval-2024 task 8: Can linguistics capture the specifics of llm-generated text? *arXiv preprint arXiv:2404.05483*.

Kristina Radivojevic, Matthew Chou, Karla Badillo-Urquiola, and Paul Brenner. 2024. Human perception of llm-generated text content in social media environments. *arXiv preprint arXiv:2409.06653*.

Soumya Suvra Ghosal, Souradip Chakraborty, Jonas Geiping, Furong Huang, Dinesh Manocha, and Amrit Singh Bedi. 2023. Towards possibilities & impossibilities of ai-generated text detection: A survey. *arXiv e-prints*, pages arXiv–2310.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*.

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.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F Wong, and Lidia S Chao. 2023. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724*.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.