A Metric-Driven Evaluation of Rephrased and Generated Texts

Kateryna Antipova[1,*,†] and Hlib Horban[1,†]

[1] *Petro Mohyla Black Sea National University, 10, 68-Desantnykyv St., Mykolaiv, 54003, Ukraine*

Abstract

The rapid development of large language models has raised serious concerns about the reliability of detecting content created by artificial intelligence. This article compares the stylistic metrics of texts generated using a multimodal model and an autoregressive model. The results show that the generated text is very similar to human-written text in terms of lexical diversity and semantic coherence. In terms of perplexity and burstiness, the rephrased texts are practically indistinguishable from the original human-written texts, which leads to a high level of false negatives in autoregressive detectors. Our analysis highlights the need for new detection methods and suggests further directions, including more specific stylometric signatures. Relying solely on a single stylometric metric leads to unreliable differentiation between generated and human-written text.

Keywords

academic abstracts, ai-generated texts, detectors, large language models, natural language processing, stylometric analysis

1. Introduction

The rapid development of large language models (LLMs) has brought about a new era in text generation, opening up a wide range of applications, from automatic content creation to dialogue agents. However, this progress is causing concern in fields that have traditionally relied on human-generated text, and raises important ethical questions. In modern implementations of AI-generated text, it is not uncommon to simply request an essay and then literally copy the result. Fortunately, there are several tools available to assess the likelihood that a text was created using AI. These detection methods mainly target the results of traditional autoregressive models (ARMs).

Most existing AI-based text recognition tools, such as DetectGPT, GPTZero, and RADAR, are designed and tuned to detect the output of autoregressive architectures (GPT-4, LLaMA, etc.). They use sequential token prediction artifacts, such as local peaks in logarithmic probability and perplexity profiles, to distinguish machine-generated text. This reliance on ARM signatures raises the question of whether these detectors can also identify AI-generated content that was created by models with other generation mechanisms. For our research, we focused on multimodal LLMs (MLLMs).

MLLMs are trained on datasets that pair images with descriptive captions or user/assistant interactions. MLLMs learn to describe pictures and answer questions about visual content, which often demands conciseness and step-by-step logic. When applied to a purely text prompt, the

Information Technology and Implementation (IT&I-2025), November 20-21, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

antipova.katerina@chmnu.edu.ua (K. Antipova); hlib.horban@chmnu.edu.ua (H. Horban)

0000-0002-9012-5290 (K. Antipova); 0000-0002-6512-3576 (H. Horban)

model may avoid complex or long clauses, use simpler syntax, and gravitate towards more template-like responses.

Despite the growing number of detection methods, there has been no systematic comparison of detectability across ARM and MLLM. To address this gap, we generated samples from ARM and MLLM for two tasks, rephrasing and text generation. Instead of testing the performance of existing detectors, we calculated chosen metrics: perplexity, burstiness, lexical diversity, semantic coherence, BLEU and ROUGE scores. These stylometric and linguistic metrics were used to assess how distinguishable the different types of generated texts are in practice.

In this work, we:

- Introduce a new dataset of 2,000 samples (500 ARM and 500 MLLM for both rephrasing and generation tasks).
- Compare MLLM- vs. ARM-style outputs and human-written text based on the stylometric and linguistic metrics.
- Use these metrics to measure detection effectiveness, discussing implications for the performance of current autoregressive-focused detection tools when faced with outputs from a MLLM.

2. AI-text Detection

2.1. Detection Methods

Supervised learning approach of fine-tuning the language model with or without adding a classification module was used by OpenAI for their RoBERTa-based classifier [1]. This approach entails fine-tuning language models on a mixture of human-authored and LLM-generated texts, enabling the implicit capture of textual distinctions. Despite the strong performance, obtaining annotations for detection data can be challenging in real-world applications, making the supervised paradigms inapplicable in some cases. While deep learning approaches often yield superior detection outcomes, their black-box nature restricts interpretability [2].

DetectGPT [3] is a zero-shot detection method that does not require training a separate classifier on human or AI–generated texts; instead, it relies solely on the language model's own probability estimates. The main idea is that AI-generated sequences leave a characteristic «signature» in the probability space of the specific model that generated them. DetectGPT assumes that machine-generated text always lies in the negative curvature region of the model's log probability function. Simply put, the model assigns higher probability to the text it generated than to adjacent alternative fragments. Based on this hypothesis, DetectGPT transforms the input text using a mask-filling language model. It then detects the AI-text by comparing the probabilities of the text and its filled-in variants. Minor changes in human-written text, such as rephrasing or word substitutions, have virtually no impact on the logarithmic probability of the model. Existing zero-shot detectors rely mainly on statistical features and use pre-trained large language models to gather them.

DetectGPT treats a candidate text x and a set of perturbed variants $\{\tilde{x}_i\}$, and computes

$$D(x) = log\, P_M\,(x) - \frac{1}{K}\sum_{i=1}^{K} log\, P_M\,(\tilde{x}_i) \qquad (1)$$

A large positive $D(x)$ means that x is a sharp log-probability peak in model M's space. Therefore, x was probably generated by model M.

DetectGPT and FastDetectGPT [4] are earlier examples of perplexity-based methods which look at the local curvature in probability space around a given example. Binoculars [5] is an even more effective recent approach which uses the cross-perplexity between two different LLMs as a signal that text is LLM-generated.

In contrast, GPTZero [6] uses a trained classifier that relies on perplexity and burstiness:

- **Perplexity analysis.** GPTZero computes sentence-level perplexities to gauge how predictable each sentence is to a language model.
- **Burstiness analysis.** It measures how much perplexities fluctuate sentence to sentence.

GPTZero flags AI text by combining low average perplexity under a reference model and low burstiness (i.e. consistently uniform sentence perplexities). Thresholds on these statistics are tuned to maximize separation between human-written and generated samples.

Ghostbuster [7] feeds LLM-generated texts into a series of weaker LLMs (from unigram models to unadjusted GPT-3 davinci) to obtain token probabilities, and then conducts a structured search on the combinations of these model outputs and trains a linear classifier to distinguish text generated by LLM. This detector achieves an average F1 score of 99.0, which is an increase of 41.6 F1 score over GPTZero and DetectGPT.

Unlike traditional binary classification tasks, stylometry-based approaches focus on distinguishing between the writing styles of different authors. Each AI model has its own stylometric signature, and identifying these different styles proves to be more effective than simple binary classification tasks. DeTeCtive [8] is a multi-task and multi-level platform for contrastive learning that achieves excellent results in detecting AI-generated texts both within and outside of distribution scenarios. It also introduces a novel feature called «training-free incremental adaptation», which allows adaptation to new data without retraining. Shah et al. [9] propose a novel approach that combines features such as lexical diversity, readability metrics, and semantic distribution with machine learning models for classification.

As AI models continue to evolve, the detectors themselves must also adapt to maintain high levels of performance and accuracy. Adversarial methods have been developed to intentionally alter the output of LLMs to evade detection. These methods can include changes in phrasing, structure, or the introduction of artificial noise that confounds detection tools.

2.2. Evading Detection

Much of the literature has also focused on whether or not AI-generated text can be detected at all [10]. Different techniques to attack or evade detectors have been developed and are an active area of research. Evasion techniques such as word or sentence substitution, recursive paraphrasing, and prompting have been developed to point out the failures in detectors [11].

A group of researchers [12] devised a framework to rank LLMs based on their detectability, claiming that more recent models like GPT-4 are less detectable because perplexity and burstiness are less useful evidence markers. The authors of [13] discuss the critical limitations of existing detectors, including issues related to real-world data issues, potential attacks, and the lack of an effective evaluation framework.

In addition, other studies have examined methods of attacking AI detectors, as well as other ways to circumvent or avoid AI detection. Sadasivan et al. [14] showed that AI text detectors can be fooled by paraphrasing attacks. The basic principle is to apply a lightweight paraphrase model on LLMs' outputs and change the distribution of lexical and syntactic features of the text to confuse the detector. Simple rephrasing techniques are sufficient to evade early zero-shot detectors and trained detectors, but recursive rephrasing is necessary to effectively evade more reliable detectors. To this end, Krishna et al. [15] proposed DIPPER, a powerful T5-based paraphrasing model that significantly enhances the effectiveness of such attacks.

RADAR [16] is a detector based on RoBERTa-large and trained using an adversarial learning model. In this model, a paraphraser is designed to rephrase machine-generated text and mimic paraphrase attacks. The RADAR framework incrementally refines the paraphrase model, drawing

on feedback garnered from the detector and employing the proximal policy optimization algorithm, outperforming zero-shot detection methods including DetectGPT and OpenAI detector.

AI humanizer, also known as a paraphrasing tool or «text humanization» tool, is used to rewrite the AI's output data multiple times in order to imitate the characteristics of human writing style. The authors of [17] evaluated 19 popular humanizer tools (e.g., Undetectable AI, WriteHuman, StealthWriter) and found that many state-of-the-art detectors fail to flag humanized AI text in over 80% of cases (e.g. only 15–20% detection rates). Simple paraphrasing loops restore perplexity and burstiness to a human-like level and effectively neutralize autoregressive detectors.

Detection tools are also shown to be unable to cope with texts translated from other languages. According to a report released by OpenAI, their AI-text detector is not fully reliable on that front [1]. In the reported evaluation of some challenging cases for English texts, their classifier only correctly identifies 26% of generated texts while incorrectly classifying 9% of human-written texts. The authors of [18] study the effect on AI detectors of translating AI-generated text through multiple languages before translating it back into English and find some methods significantly more robust than others.

The accuracy and reliability of AI-generated text detection tools can vary depending on several factors, such as the specific tool used, the type of AI model generating the text, and the content being analyzed. Most of the detection tools achieve a 70-80% accuracy rate in detecting text generated by models like GPT-3. Detectors also struggle with short text paragraphs and with more advanced outputs from later-generation models like GPT-4 [11].

3. Multimodal LLMs

The trend toward integrating multiple modalities into architectures is becoming increasingly widespread and is leading to the emergence of multimodal large language models (MLLMs). Multimodal generation represents the pinnacle of achievements in individual modalities and integrates text, images, video, and audio into context-aware outputs. For example, tasks such as text-to-image, text-to-video, and text-to-speech represent multimodal systems that go beyond pure text generation and use text prompts to control the generation of visual content [19].

The architecture is typically modular or monolithic. However, most existing MLLMs use a modular architecture in which visual encoding and language decoding are processed separately. This approach is typically realized by combining a pre-trained visual encoder (e.g., a CLIP-based ViT) with a LLM [20].

These models differ from traditional text-only LLMs not only in architecture but also in the diversity of their training data, which includes image-text pairs, visual reasoning tasks, and cross-modal alignments. Broader training scope introduces new challenges for detectors: while traditional detectors focus on linguistic features, MLLM-generated text may exhibit distinct stylistic patterns influenced by multimodal conditioning, making detection strategies based solely on text stylometry less reliable.

Both commercial models, such as GPT-4o and the Gemini series, and open-source ones, such as BLIP [21] and LLaVA [22], have been actively working on combining image and language modalities. They often link LLMs with large vision models (LVMs) through intermediate layers. Recent open-source frameworks demonstrate the efficacy of modular designs. Through large-scale multimodal pre-training and advanced visual-language alignment techniques, they achieve outcomes on par with leading commercial models.

Despite their multimodal capabilities, MLLMs can perform pure text generation tasks, functioning similarly to autoregressive LLMs. During inference without visual inputs, the language component processes text prompts and generates continuations based on learned distributions. However, their stylistic tendencies often differ from text-only LLMs because of exposure to image-caption datasets and conversational multimodal instructions during training. This bias can manifest as shorter, more descriptive sentences, preference for concrete nouns, and a more

directive or explanatory tone. Additionally, MLLMs often rely on special tokens or structured prompts to manage dialogue or multimodal context, which influences their default response format.

The text generation process in MLLMs, as in LLMs, depends on sampling methods that control diversity and determinism. In multimodal contexts, sampling occurs after modality fusion or token alignment, so the language model conditions its predictions on both text and any embedded visual features. For deterministic tasks such as text generation, non-sampling settings are typically preferred to ensure consistency and minimize stylistic variance. It affects not only output fluency but also the detectability of generated text, as different decoding strategies produce different stylometric fingerprints.

4. Metrics

Stylistic features primarily focus on the frequency of words that specifically highlight the stylistic elements of the text, including the frequency of capitalized words, proper nouns, verbs, past tense words, stopwords, technical words, quotes, and punctuation. Complexity features are extracted to represent the complexity of the text, such as the type-token ratio and textual lexical diversity. Psychological features are generally related to sentiment analysis and can be derived based on existing tools to calculate sentiment scores, or extracted using sentiment classifiers.

To quantify stylistic and statistical differences among original, ARM-generated, and MLLM-generated texts, we compute the following metrics.

Perplexity, to measure how predictable a text is to a strong ARM. For a model M the perplexity of text $x_{1:N}$ is

$$PP_M(x_{1:N}) = exp\left(-\frac{1}{N}\sum_{t=1}^{N} log\, P_M\,(x_t|x_{1:t-1})\right) \quad (2)$$

Lower PP means that the model finds the text more predictable. AI text has lower PP than human text.

Burstiness, which refers to how unevenly or clustered certain words or other features appear in a text. The variance of sentence-level perplexities:

$$Burst(x) = Var\{PP_M(s)|s \in sentences(x)\} \quad (3)$$

Human writing often shows higher burstiness than machine-generated texts; generated outputs tend to be more uniform. Autoregressive models show less burstiness especially if they are trained to avoid repetition. Burstiness can be measured by the following:

- variance-to-mean ratio (index of dispersion) for word frequency across segments of a text;
- statistical indicators of deviation from a uniform distribution;
- temporal autocorrelation in sequential token occurrence.

Lexical diversity, a type-token ratio.

Semantic consistency, average cosine similarity between adjacent sentence embeddings:

$$\frac{1}{K-1}\sum_{i=1}^{K-1} cos(emb(s_i), emb(s_{i+1})) \quad (4)$$

Higher values indicate smoother transitions and greater coherence.

BLEU, precision. Scores are calculated for individual sentences by comparing them with a set of reference sentences. Those scores are then averaged to get an estimate of the overlap. Intelligibility and grammatical correctness are not taken into account.

ROUGE-1, refers to the overlap of unigrams between the model and reference sequences.

ROUGE-L, is based on the longest common subsequence. It considers sentence-level structure similarity and identifies longest co-occurring in sequence n-grams.

These stylometric and linguistic metrics reveal both surface-level and deeper linguistic patterns.

5. Experimental Part

For our experiments we used the ArXiv Paper Abstracts dataset [23], which comprises articles' titles, and abstracts. We randomly sample 500 title-abstract pairs from this corpus. We define two tasks for each selected pair:

1. Rephrasing task. We feed the original abstract to each model with the prompt:

Rephrase the following academic abstract: {original abstract}. Provide only the rephrased abstract.

2. Abstract generation. We feed the article's title to each model with the prompt:

Write an academic abstract for a paper titled: {title}. Provide only the abstract.

We compare text variants per each pair (original, ARM output, MLLM output) for both tasks, for a total of 2 × 3 × 500 = 3000 samples.

ARM baseline: We use the Microsoft model Phi-3-Mini-4K [24] via HuggingFace Transformers. At inference we apply deterministic sampling with the following settings:

temperature = 0.0, top_p = 1.0, max_new_tokens = 128, do_sample = False.

MLLM baseline: We use the IDEFICS-9B model [25] with the same settings and a dummy image the model requires.

Perplexity was calculated using pretrained gpt2 model, the smallest version of GPT-2 with 124M parameters [26]. Semantic consistency was calculated using all-MiniLM-L6-v2, a pretrained Sentence Transformers model with over 22M parameters [27].

All runs are performed on a Google Colab A100 GPU.

Table 1
Mean and SD of key metrics for the rephrasing task

Metric	Original	ARM	MLLM
Perplexity	47.99 ± 18.56	46.49 ± 24.08	44.56 ± 18.5
Burstiness	5.82 ± 2.66	2.2 ± 1.376	4.22 ± 2.41
Lexical diversity	0.63 ± 0.06	0.77 ± 0.059	0.665 ± 0.07
Semantic coherence	0.39 ± 0.086	0.35 ± 0.125	0.396 ± 0.1

Table 2
Mean and SD of key metrics for the text generation task

Metric	Original	ARM	MLLM
Perplexity	47.99 ± 18.56	24.72 ± 6.85	17.49 ± 7.25
Burstiness	5.82 ± 2.66	2.25 ± 1.29	1.79 ± 1.28
Lexical diversity	0.63 ± 0.06	0.74 ± 0.05	0.62 ± 0.11
Semantic coherence	0.39 ± 0.086	0.435 ± 0.12	0.46 ± 0.15

Table 3
Mean of BLEU/ROUGE metrics for the rephrasing task

	BLEU	ROUGE-1	ROUGE-L
ARM	0.046	0.39	0.26
MLLM	0.6	0.82	0.78

To give a better overview the distributions are illustrated in Figures 1-3.

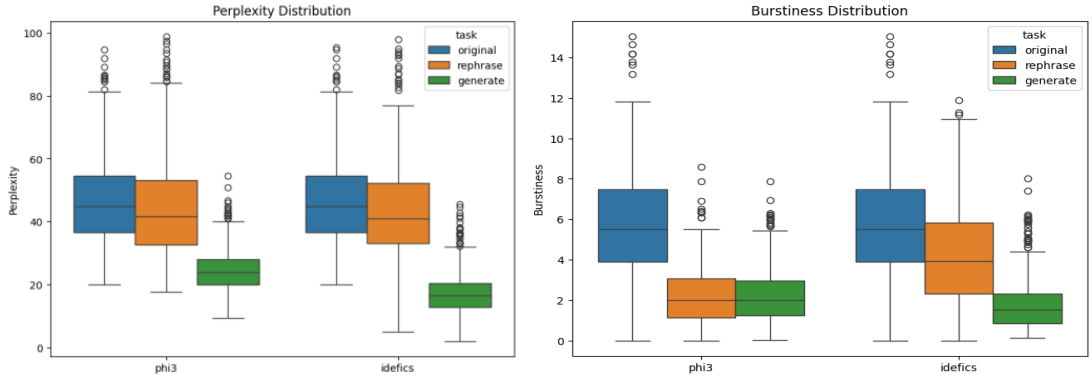

Figure 1: Perplexity and Burstiness Distribution.

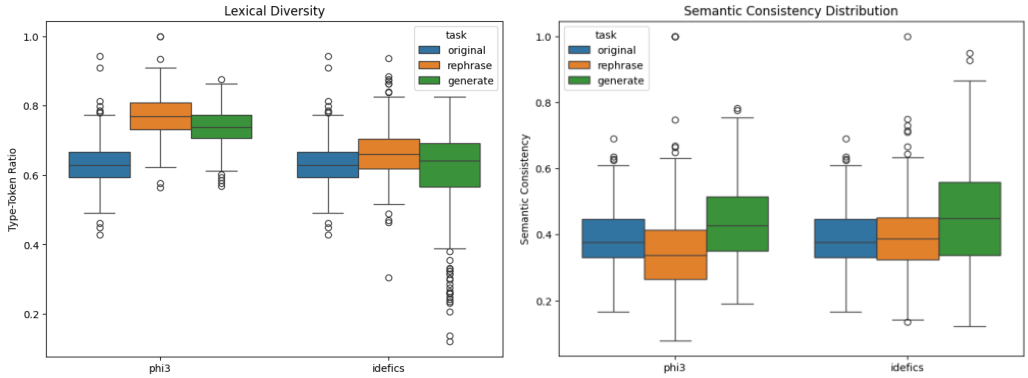

Figure 2: Lexical Diversity and Semantic Consistency Distribution.

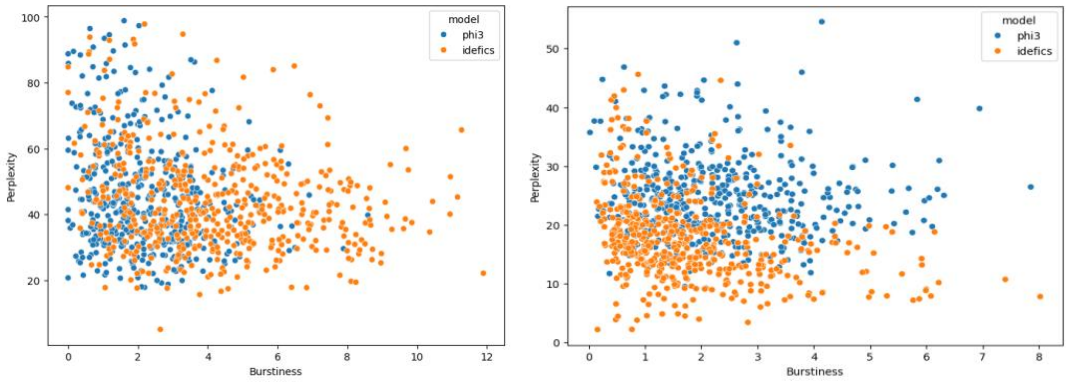

Figure 3: Perplexity vs. Burstiness for Rephrasing and Generation Tasks.

In the rephrase task ARM paraphrases at T = 0 achieve perplexities virtually indistinguishable from human originals, whereas MLLM outputs yield burstiness only slightly lower than human originals, which makes its outputs stealthier. Therefore, detectors that report low perplexity miss both models whose deterministic samples fall within the human range.

In the text generation task, both models used T = 0 again, yielding much lower perplexity and burstiness than human-written abstracts, which makes both models quite predictable. Although, the models' outputs can still trick many detectors, that are tuned only to perplexity and sentence-length variability.

Setting the temperature to zero emphasizes the characteristic features of each model's style and at the same time ensures that the output results are completely deterministic and predictable compared to texts whose samples were selected at higher temperatures. Since our analysis is limited to abstract-length sentences, we may miss stylometric cues that are only found in longer documents.

Main strengths and weaknesses of the two models are listed below.

ARM:

- High perplexity in the rephrasing task.
- High lexical diversity and semantic coherence in both tasks support novel wording and stylistic variety.
- In text generation, perplexity is low which makes the results obvious to autoregressive detectors.

MLLM:

- In rephrasing, perplexity and burstiness remain within human range.
- High BLEU/ROUGE scores preserve source wording.
- High lexical diversity and semantic coherence is demonstrated in both tasks.
- Low perplexity and burstiness in the text generation task.

6. Conclusions

In this work, a comprehensive stylometric analysis was conducted to assess the detection-performance of AI-generated abstracts, comparing the results from an autoregressive model, a multimodal model, and original human-written texts. All model outputs were generated with a zero decoding temperature to exclude stochastic fluctuations and expose each model's stylistic bias. Deterministic approach reduces perplexity, but makes autoregressive detectors over-confident. Future studies should examine higher-temperature samples to assess its effect on detectability.

The obtained results show that reliance on a single metric, such as a fixed perplexity threshold, is insufficient for robust AI-text detection. Detection pipelines should combine multiple stylometric signals (perplexity, burstiness, lexical diversity, etc.) to improve sensitivity to both ARM and MLLM outputs. These results indicate the need for next-generation detection. Future research will expand the dataset to other model families and explore other linguistic and semantic metrics.

Future research will extend the analysis from articles' abstracts to full-text in order to investigate how stylometric metrics evolve with document length and topic. In addition, attack-and-defense cycles will be studied to evaluate the resistance of detectors to adversarial attacks. Adaptation to novel humanization tools will enhance detectors robustness.

Declaration on Generative AI

During the preparation of this work, the authors used Phi-3-Mini-4K and IDEFICS-9B models to generate datasets for subsequent linguistic analysis. The authors used pretrained gpt2 model to calculate perplexity in text. The authors used all-MiniLM-L6-v2 model to calculate semantic consistency in text. The authors take full responsibility for the publication's content.

References

[1] New AI classifier for indicating AI-written text, 2023. URL: https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text

[2] R. Tang, Y. Chuang, X. Hu, The Science of Detecting LLM-Generated Text, in: Communications of the ACM, vol. 67, 2023, pp 50-59. doi:10.48550/arXiv.2303.07205.

[3] E. Mitchell, Y. Lee, A. Khazatsky, C.D. Manning, C. Finn, DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature, in: International Conference on Machine Learning, 2023. doi:10.48550/arXiv.2301.11305.

[4] G. Bao, Y. Zhao, Z. Teng, L. Yang, Y. Zhang Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature, in: International Conference on Learning Representations, 2023. doi:10.48550/arXiv.2310.05130.

[5] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, Spotting LLMs with Binoculars: Zero-shot detection of machine-generated text, in: International Conference on Machine Learning, 2024. doi:10.48550/arXiv.2401.12070.

[6] AI Detector – the Original AI Checker for ChatGPT & More, 2023 URL: https://gptzero.me/

[7] V. K. Verma, E. Fleisig, N. Tomlin, D. Klein, Ghostbuster: Detecting Text Ghostwritten by Large Language Models, in: North American Chapter of the Association for Computational Linguistics, 2023. doi:10.48550/arXiv.2305.15047.

[8] X. Guo, Sh. Zhang, Y. He, T. Zhang, W. Feng, H. Huang, Ch. Ma, Detective: Detecting ai-generated text via multi-level contrastive learning, in: Neural Information Processing Systems, 2024. doi:10.48550/arXiv.2410.20964.

[9] A. Shah, P. Ranka, U. Dedhia, S. Prasad, S. Muni, K. Bhowmick, Detecting and unmasking ai-generated texts through explainable artificial intelligence using stylistic features, in: International Journal of Advanced Computer Science and Applications, vol. 14(10), 2023.

[10] İ. Tarım, A. Onan, Can You Detect the Difference?, 2025. doi:10.48550/arXiv.2507.10475.

[11] K. Antipova, H. Horban, Improving detection of AI-generated text in education, in: Directions for the development of science in the context of global transformations, Baltija Publishing, Riga, Latvia, 2025, pp. 1–19. doi:10.30525/978-9934-26-562-4-1.

[12] M. Chakraborty, S. M. T. Tonmoy, S. M. Zaman, S. Gautam, T. Kumar, Counter Turing Test (CT2): AI-generated text detection is not as easy as you may think – introducing AI detectability index (ADI) in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2023, pp. 2206–2239. doi:10.18653/v1/2023.emnlp-main.136.

[13] J. Wu, S. Yang, R. Zhan, Y. Yuan, L. S. Chao, D. F. Wong, A survey on LLM-generated text detection: Necessity, methods, and future directions, in: Computational Linguistics, 2025, pp. 1-65. doi:10.48550/arXiv.2310.14724.

[14] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, S. Feizi, Can AI-generated text be reliably detected?, 2023. doi:10.48550/arXiv.2303.11156.

[15] K. Krishna, Y. Song, M. Karpinska, J. Wieting, M. Iyyer, Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense, in: Advances in Neural Information Processing Systems, vol. 36, 2023, pp. 27469-27500. doi:10.48550/arXiv.2303.13408.

[16] X. Hu, P. Chen, T. Ho, RADAR: Robust AI-Text Detection via Adversarial Learning, 2023. doi.org/10.48550/arXiv.2307.03838.

[17] E. Masrour, B. Emi, M. Spero, Damage: Detecting adversarially modified AI generated text, 2025. doi.org/10.48550/arXiv.2501.03437.

[18] N. Ayoobi, L. Knab, W. Cheng, D. Pantoja, H. Alikhani, Esperanto: Evaluating synthesized phrases to enhance robustness in AI detection for text origination, 2024. doi:10.48550/arXiv.2409.14285.

[19] Y. Zou, P. Li, Z. Li, H. Huang, X. Cui, X. Liu, C. Zhang, R. He, Survey on AI-Generated Media Detection: From Non-MLLM to MLLM, 2025. doi:10.48550/arXiv.2502.05240.

[20] Z. Chen, W. Wang, H. Tian, S. Ye, Z. Gao, How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites, in: Science China Information Sciences, vol. 67(12), 2024. doi:10.48550/arXiv.2404.16821.
[21] J. Li, D. Li, S. Savarese, S. Hoi, BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, in: International conference on machine learning, 2023, pp. 19730-19742. doi:10.48550/arXiv.2301.12597.
[22] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, Llavanext: Improved reasoning, ocr, and world knowledge, 2024. URL: https://llava-vl.github.io/blog/2024-01-30-llava-next
[23] Kaggle, arXiv paper abstract dataset for building multi-label text classifiers. URL: https://www.kaggle.com/datasets/spsayakpaul/arxiv-paper-abstracts
[24] M. Abdin, S.A. Jacobs, A.A. Awan, J. Aneja, A. Awadallah, Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone, 2024. doi:10.48550/arXiv.2404.14219
[25] HuggingFace, IDEFICS, 2023. URL: https://huggingface.co/HuggingFaceM4/idefics-9b-instruct
[26] HuggingFace, GPT-2, 2019. URL: https://huggingface.co/openai-community/gpt2
[27] Sbert.net, Sentence Transformers Documentation, 2025. URL: https://www.sbert.net/index.html