PARAPHRASING AS EVASION MECHANISM AGAINST AI-GENERATED TEXT DETECTORS

Kateryna Antipova[1]
Viktor Ralenko[2]

DOI: https://doi.org/10.30525/978-9934-26-651-5-2

Abstract. The demand for reliable methods to identify AI-generated text has become increasingly urgent. Effective detection is necessary to curb misuse and to safeguard domains such as social media from adverse impact. Potential attacks contribute significantly to the unreliability of current AI-text detectors, with paraphrasing-based attacks representing one of the most serious challenges. The evasion methods differ not only in their strength but also in the assumptions about detection they undermine, leading to qualitatively different failure modes across different detector families. *The purpose* of the paper is to investigate evasion methods to uncover weaknesses in existing detectors. The analysis concentrates specifically on how effectively paraphrasing can bypass modern AI detectors. To conduct this research, a dataset was compiled, consisting of sequences produced by three different large language models and subsequently rewritten using three different paraphrasing tools. *Methodology* of the study is based on research techniques of analysis and synthesis, experimental testing, and quantitative analysis. This allows for a comprehensive investigation and comparison of the effectiveness and performance of different detectors. *The results* suggest that current post-hoc AI-generated text detectors provide limited reliability in realistic adversarial settings. Even paraphrases that preserve very high semantic similarity substantially degrade detection performance, particularly at low false-positive operating points required for high-stakes use cases such as academic integrity or hiring. Further research could develop a more nuanced classification system that goes beyond the simple binary of human-written and LLM-generated texts. Future research

[1] Doctor of Philosophy,
Associate Professor at the Department of Software Engineering,
Petro Mohyla Black Sea National University, Ukraine
[2] Lecturer at the Department of Software Engineering,
Petro Mohyla Black Sea National University, Ukraine

should also include different repetition depths of paraphrasing to assess the robustness of the recognition systems as the texts gradually move into the robust recognition area. *Practical implications.* The research will contribute to a more rigorous and security-oriented understanding of AI-generated text detection by quantifying detector robustness under paraphrasing attacks. It advances the field by demonstrating how semantic preservation can coexist with statistical indistinguishability, thereby exposing structural weaknesses in current post-hoc detection methods. *Value/originality.* The scientific novelty lies in jointly analyzing semantic similarity, AUROC, and TPR at fixed FPR, which exposes failure modes that are obscured when reporting accuracy alone. This contributes to a more security-aligned and practically grounded assessment of AI-text detection under realistic evasion conditions.

1. Introduction

The rapid expansion of AI-driven applications has greatly enhanced natural interaction between humans and machines while broadening public access to capabilities that were once limited to experts. Tools ranging from AI-assistants to search engines powered by large language models (LLM) and domain-specific solutions for programming and scientific analysis illustrate how LLMs have become deeply embedded in everyday digital environments, distributing content across numerous platforms.

As LLM adoption accelerates, the demand for reliable methods to identify AI-generated text has become increasingly urgent. Effective detection is necessary to curb misuse and to safeguard domains such as creative work and social media from adverse impact. Recent advances, including watermarking strategies, statistical approaches, and neural network-based classifiers, have substantially improved detection capabilities. Systems designed to flag generated text are central to differentiating human authorship from automated generation, helping to counter disinformation, fraud, and widening disparities in educational contexts.

Text generation detection systems face persistent pressure from evasion methods, with paraphrasing-based attacks representing one of the most serious challenges. Investigating these strategies is critical for uncovering weaknesses in existing detectors and strengthening their resilience.

Consequently, the establishment of rigorous and standardized evaluation frameworks has become increasingly urgent.

This study provides a structured survey of the current literature and emphasizes the need for intensified research on detection technologies. By examining key shortcomings of contemporary systems, particularly their susceptibility to adversarial manipulation, it clarifies central issues within the field. The analysis concentrates specifically on how effectively paraphrasing can bypass modern AI detectors. To conduct this research, a dataset was compiled, consisting of sequences produced by three different large language models and subsequently rewritten using three different paraphrasing tools.

Methodology of the study is based on research techniques of analysis and synthesis, experimental testing, and quantitative analysis. This allows for a comprehensive investigation and comparison of the effectiveness and performance of different detectors.

2. AI-generated text detectors

Current methods for detecting text generated by LLMs can be divided into two categories: watermark-based and non-watermark-based methods. Watermark-based methods involve subtle modifications to the generated text that are imperceptible to human readers but can be identified by specialized algorithms during post-hoc analysis. Effective watermarks are carefully designed to be robust against removal and to minimize the impact on the overall quality of the generated text. It is important to note that watermark-based methods are, in principle, only applicable to certain LLMs. Furthermore, the development of watermark-based algorithms is limited by the need for access to open-source LLMs [1].

Compared to watermark-based methods, non-watermark-based methods offer a distinct advantage, as they can detect text generated by various LLMs without altering the generation algorithm. Early non-watermark-based recognition strategies focused on detecting statistical anomalies in metrics such as entropy and entanglement. With the advent of ChatGPT, zero-shot detectors were developed that exploit statistical and topological properties of LLM-generated texts. In contrast, classifier-based methods train supervised models to distinguish between human-authored and LLM-generated text. Zero-shot detectors exploit the fact that tokens in

LLM-based outputs are generally predicted as consistently more likely by the LLM itself. DetectGPT [2] and FastDetectGPT [3] are early examples of perplexity-based methods that consider the local curvature in the probability space around a given example. Binoculars [4] is an even more powerful, recent approach that uses cross-perplexity between two different LLMs as a signal that a text was generated by an LLM.

Deep learning methods attempt to detect AI-generated content using neural networks. Specifically, they use large datasets of known texts generated by both humans and AI to train classifiers that can distinguish between them. OpenAI's classifier was one of the first attempts. It used a RoBERTa-based model to classify text written by humans and text written with GPT-2. Ghostbusters [5] recognizes text generated by LLMs using a combination of trained features derived from an embedding language model.

A substantial body of work examines how model size impacts detection systems. This issue can be analyzed along two dimensions: the scale of the generative model and that of the supervised classifier. The capacity of the generative model strongly influences the realism of its outputs. Typically, content produced by smaller models is easier to identify, whereas outputs from larger models are more difficult to detect. An additional concern involves the effect of training detectors on texts generated by models of varying sizes. Detectors trained on data from mid-sized large language models often transfer effectively to larger variants without requiring extra data. In contrast, relying on outputs from models that are excessively small or excessively large may weaken generalization performance. Concerning the supervised classifier, detection performance generally increases with the size of the fine-tuned language model. Nevertheless, recent findings indicate that although larger detectors achieve superior results on test sets drawn from the same distribution as the training data, their ability to generalize beyond that distribution may decline [6].

Stylometry is an established tool for identifying and verifying authorship in many fields. It uses various features to detect stylistic variations within a document, thereby identifying the author of a particular text or uncovering author-specific differences in documents with multiple authors.

Table 1

Key characteristics of detection methods

Detectors	Method	Approach	Strengths	Weaknesses
Kirchenbauer's watermarking, PersonaMark [7]	Watermarking	Incorporates an imperceptible signal to establish the authorship of a specific text; analogous to encryption and decryption	Provides strong ownership tracking, difficult to bypass	Requires AI models to adopt watermarking
OpenAI classifier, Ghostbuster, RADAR	Deep learning based	Trained or fine-tuned for binary classification with datasets containing human and AI-generated texts	High detection accuracy, learns complex text patterns	Requires significant computational resources and labeled datasets
DetectGPT, FastDetectGPT, Binoculars, GPTZero	Metric-based (zero-shot)	Uses predefined metrics like perplexity, burstiness, log-likelihood, etc)	Fast, scalable, interpretable	Less effective against advanced evasion techniques
Kumarage's detector [8]	Stylometric-based	Analyzes the linguistic style of text in order to differentiate between various writers	Interpretable, captures broad linguistic styles	Very susceptible to paraphrasing and style transfer
Krishna's detector [9], SEFD [10]	Retrieval-based	Searches for a candidate passage in a database that stores the LLM outputs	Robust to paraphrasing and surface-level rewriting	Requires large, representative databases of AI outputs

Stylometric analysis is an intensively researched field, and scholars have proposed a wide range of lexical, syntactic, semantic, and structural features for author identification. Stylistic features aim to identify characteristic stylistic signals in a given text fragment. Authors of [8] use three categories of features:

1. Usage – features that quantify how an author arranges words and phrases in a text fragment (e.g., average word count, number of deviations, etc.).

2. Punctuation – features that quantify how an author uses different punctuation marks (e.g., average number of different punctuation marks).

3. Linguistic variation – features that quantify how an author uses different words in a text (e.g., vocabulary richness and readability).

The vocabulary richness of a text is measured by calculating the moving-average of the type-token ratio (MTTR). This ratio considers the average frequency of different words within a given word group and measures lexical diversity. To determine readability, the well-known Flash readability metric is used, which rates the readability of a text on a scale of 0 to 100.

Retrieval-based detectors compare candidate texts with a large repository of known or AI-generated sample texts. Instead of learning a global boundary between human-written and AI-generated texts, the detector embeds the input text and extracts its nearest neighbors using a dense similarity search. The detection decisions are based on semantic proximity, similarity scores, or neighborhood statistics, and assume that the semantic content is preserved through rewording and superficial edits. Therefore, retrieval-based detectors are inherently robust against lexical and syntactic rewriting. However, their performance depends on the completeness, diversity, and recency of the search corpus, as well as the quality of the embedding model used for similarity estimation [9, 10].

Retrieval-based methods avoid the need to retrain distribution models. This means they do not have to learn a global boundary between "human" and "AI" text, which becomes increasingly blurred as the model improves. Instead, they answer a more specific question: is the text semantically similar to what the model is known to generate?

Retrieval-based detection reduces to a nearest-neighbor or similarity search problem, making the assumptions explicit. These detectors also exhibit a more gradual decay: as the strength of the paraphrase increases, the similarity score decreases continuously rather than exponentially, making them easier to analyze under attack.

3. Potential Attacks

Much of the literature focuses on attacking AI detectors and other methods for circumventing AI detection [11]. One study [9] develops an approach to identify soft prompts that can generate texts that evade detection. Another

study [12] investigates the impact of translating AI-generated texts into multiple languages and back-translating them into English on detectors, revealing this method to be significantly more reliable than others.

The study [13] directly optimizes a language model by using an AI detector as a negative reward. This involves creating pairs of LLM-generated texts, in each of which one fragment is recognized and the other is not. The language model is then optimized using dual-perturbation optimization (DPO) to favor the unrecognized output. RADAR [14] develops a more reliable detector by training a language model detector and a paraphrase generator against each other.

Potential attacks contribute significantly to the unreliability of current LLM-generated text detectors. The evasion methods differ not only in their strength but also in the assumptions about detection they undermine, leading to qualitatively different failure modes across different detector families.

Paraphrasing attacks rank among the most powerful strategies against watermarking systems, fine-tuned supervised detectors, and zero-shot detection methods. Their core mechanism involves passing language model outputs through a light-weight paraphrasing model, which disrupts detection by modifying lexical choices and syntactic patterns. DIPPER [9], an 11B paraphrasing model, enables fine-tuning of paraphrase diversity and the extent of content restructuring, leading to notable degradation in the effectiveness of current detection techniques. Although retrieval-based defenses have shown promise in mitigating such attacks, they depend on continuous cooperation from language model API providers and remain susceptible to iterative paraphrasing strategies [15].

Adversarial attacks such as substitution can significantly impair the accuracy of detectors [16]. The authors of [6] group attacks that manipulate text features under the term "adversarial attacks." This includes cutoff (cutting out features or parts of the input), shuffle (randomly changing the word order in the input), mutation (changing characters or words), word swapping (replacing related words with others, taking context into account), and misspelling.

The authors of [17] report that a permutation approach is effective for attack detection systems. By replacing words with context-based synonyms, they developed effective attacks against fine-tuned classifiers, watermarking, and DetectGPT. This reduced the detector's performance by

over 18%, 10%, and 25%, respectively. Studies have shown that the detector becomes more sensitive to syntactic perturbations, such as splitting longer sentences, removing definite articles, and reformatting machine-generated text.

Table 2

Impact of evasion techniques on detection accuracy

Technique	Description	Impact	Affected detectors
Paraphrasing	Rewriting text while maintaining the original meaning	Decreases accuracy by altering surface-level features	Stylometric, metric-based
Recursive paraphrasing	Iteratively rewording AI-generated text to disguise AI patterns	Greatly reduces detection accuracy by removing statistical patterns	Metric-based, stylometric, classifier-based
Back-translation (round-trip translation)	Translating into different language, then translating back into the original language	Strong normalization effect; often comparable to heavy paraphrasing	Metric-, stylometric-based, watermarking
Obfuscation	Modifying text to confuse detectors (misspellings, special characters)	Exploits detector sensitivity to surface cues	Stylometric-, classifier-based
Substitution	Replacing words with synonyms or altering sentence structures	Reduces reliance on exact-word matching and perplexity analysis	Metric-, stylometric-based
Prompting	Crafting AI inputs to generate more human-like responses	Reduces effectiveness of perplexity-based metrics	Classifier-, metric-based
Adversarial learning methods	Using reinforcement learning to refine generative models to circumvent detectors	Undermines static detection assumptions	Classifier-, metric-, stylometric-based

Existing detection methods are highly vulnerable to adversarial attacks, with the degree of resilience varying depending on the detector type [18]. One study [19] showed that supervised approaches provide effective protection against these attacks. Adversarial learning can significantly improve a detector's ability to recognize text manipulated by such attacks.

Research has shown that synonym substitution, fake-fake replacement, insertion instead of substitution, and changing the substitution position do not impair Grover's detection capability. However, adversarial embedding techniques can effectively trick Grover into classifying counterfeit items as genuine. This attack significantly degrades the performance of fine-tuned classifiers, although these can learn the attack's distribution properties and build a strong defense [6].

Prompt attacks pose a major challenge to current methods for detecting LLM-generated text. The quality of LLM-generated text is related to the complexity of the prompts that guide LLMs in text generation. As model and corpus sizes increase, LLMs with superior ICL (in-context learning) capabilities for generating more complex texts emerge. Numerous effective prompting methods have been developed, including combining prompt, few-shot prompt, Chain of Thought (CoT), and zero-shot CoT, etc., which have significantly improved the quality and functionality of LLMs. Previous research on LLM-generated text detectors has primarily used datasets generated with simple, direct prompts. For example, study [20] showed that detectors can struggle to identify texts generated with complex prompts. Study [21] reports that the detection capability of detectors using finely tuned language models decreases significantly with a large number of prompts. This suggests that using different prompts can substantially affect the detection performance of existing detectors.

It is important to note that both the paraphrasing and adversarial attacks mentioned above can be carried out through targeted prompt design. Given the ongoing research in natural language processing, the risks of prompt attacks are expected to increase further. This underscores the need to develop more robust detection methods that can effectively counter such constantly evolving threats.

Adversarial learning methods have already proven effective in attacking existing detectors. The authors of [13] used the "humanity" scores of various open-source and commercial detectors as a reward function for reinforcement learning, which fine-tunes language models to deceive existing detectors. A similar approach was demonstrated in [22]. By improving the generative model using reinforcement learning, the authors were able to successfully bypass BERT-based classifiers with detection accuracies as low as 0.15 AUROC, even when using linguistic features as the reward function.

In [8], a universal escape framework called EScaPe was proposed to help models generate "human-like text" that can mislead detectors. By evasive soft prompt learning and data transfer, the performance of DetectGPT and OpenAI Detector can be effectively reduced by up to 40% of the AUROC score. This study also revealed another potential vulnerability of detectors: if a generative model has access to the human-written text on which the detector was trained and can use it for fine-tuning, it becomes impossible for that detector to recognize text on the generative model. This suggests that LLMs trained on larger corpora of human-written text are more robust against existing detectors and that training a specific detector can give an LLM a more powerful weapon to overcome their defenses.

Sample enhancement based adversarial training focuses on the use of adversarial attacks. The main goal is to generate misleading inputs to improve a model's ability to solve a wider range of potentially misleading scenarios. This technique particularly emphasizes the importance of sample augmentation and achieves this by injecting predefined adversarial attacks. This augmentation process is essential for improving the robustness of a detector because it provides an expanded pool of adversarial samples. The study [17] performed adversarial data augmentation on LLM-generated texts and showed that models trained with carefully augmented data exhibited remarkable robustness against potential attacks.

The two-player games methods typically involve the simultaneous construction of an attack model and a detection model. Their detection capabilities are improved through repeated competition. The authors of [14] presented the RADAR framework, which was designed for the simultaneous training of a robust detector using adversarial learning. This framework enables interaction between a paraphrasing model, which generates realistic content to evade detection, and a detector, whose goal is to better identify text generated by LLMs. The RADAR framework uses feedback from the detector and applies Approximate Policy Optimization (PPO) to incrementally improve the paraphrasing model.

In parallel, the authors of [23] proposed a methodology for training a detector based on continuous interaction between the attacker and the detector. In contrast to RADAR, OUTFOX focuses more on the detector's potential to identify the attacker using ICL. Specifically, in the OUTFOX framework, the attacker uses the labels predicted by the detector to generate

difficult-to-detect text. Conversely, the detector uses adversarially generated content to improve its detection capabilities against serious attackers. Reciprocal consideration of each other's outputs increases the detector's robustness against text generated by LLMs. Empirical results suggest that the OUTFOX method outperforms traditional statistical methods and methods based on RoBERTa [9].

4. Paraphrasing

Paraphrasing attacks have become the most effective evasion strategy. These attacks systematically paraphrase AI-generated content while preserving meaning, thus "washing" synthetic text to make it appear as if it was written by humans [9]. Advanced techniques such as recursive paraphrasing significantly reduce recognition efficiency but preserve text quality [15]. Unlike techniques that require deep technical knowledge, paraphrasing is easy to perform. This results in the accuracy of even the most advanced detection methods being almost random and poses serious risks ranging from education to information security.

The proliferation of paraphrasing attacks has exposed significant weaknesses in current systems for evaluating the robustness of recognition methods. While existing benchmarks provide comprehensive recognition assessments, they rely on single-step DIPPER-based paraphrasing without systematic robustness evaluation. PARAPHRASUS [24] also evaluates the paraphrase detection of various models using Classify, Min, and Max calls on established datasets. However, even if these calls yield good results, this does not necessarily mean they provide robust protection against malicious activity. These artificial scenarios focus on paraphrase detection rather than systematically evaluating the detector's vulnerability to iterative evasion attacks. None of these frameworks addresses a crucial gap: evaluating detector performance against realistic attacks based on multiple iterative paraphrases.

Paraphrasing naturally shifts the statistical characteristics of machine-generated text, enabling it to mislead anomaly detectors and classification models while decreasing the presence of watermark signals. To bypass these systems effectively, a paraphrasing model must handle broader context, such as prompts or multi-sentence inputs. Its transformations should also be carefully regulated, introducing only those modifications required to

circumvent a specific detector. At the same time, the semantic content of the original text must remain largely intact. Additionally, the paraphrasing system should differ from the watermarked source model. Otherwise, the resulting output may inherit the same watermark.

The study [15] demonstrates the effectiveness of paraphrasing attacks compared to trained detectors. The AUROC value of DetectGPT drops from 96.5% before the attack to 59.8% afterward. An AUROC value of 50% is comparable to a random detector. The other zero-shot detectors also perform poorly after the attack. The trained, neural network-based detector performs better than the zero-shot detector but is less robust. Although the performance of the trained detector deteriorates with each paraphrasing round, it appears to be more resistant to paraphrasing attacks than the other detectors.

The retrieval-based detector in [9] is designed to protect against paraphrasing attacks. However, studies in [14] show that it can be vulnerable to recursive paraphrasing attacks developed with DIPPER. This detector recognizes almost all AI output after just one paraphrasing round. However, after five rounds of recursive paraphrasing, its detection accuracy drops below 60%. Furthermore, retrieval-based detectors are a cause for concern because they store the conversations of LLM users, potentially creating serious privacy issues.

Repeated paraphrasing leads to semantic shifts while preserving generative patterns. This mechanism enables two distinct attack scenarios:

1. ***Authorship Obfuscation.*** Human-written texts that have been repeatedly paraphrased retain human-like stylistic features despite semantic shifts. This creates a detection blind spot and allows for the unauthorized appropriation of human texts.

2. ***Plagiarism Detection Evasion.*** Even after repeated paraphrasing, LLM-generated texts retain human-like generative patterns while undergoing semantic transformations sufficient to circumvent plagiarism detection systems and thus contribute to scholarly misconduct.

While iterative paraphrasing of human-written texts (authorship obfuscation) and repeated paraphrasing of LLM-generated texts (plagiarism detection evasion) pose fundamentally different risks, both exploit the same space of text manipulation. The research shows that paraphrased texts, regardless of their origin, converge into this semantic space, characterized

by the preservation of generative patterns and semantic changes. So, we must:

1) move beyond the binary classification of "human vs. AI" and understand how texts of different origins traverse and occupy this space;

2) include different repetition depths of paraphrasing to assess the robustness of the recognition systems as the texts gradually move into this robust recognition area.

5. Detection accuracy

Detection accuracy measures how often input text is correctly identified as being AI-generated. Since the recognition rate is highly dependent on the chosen threshold, the AUROC metric [2], which considers different thresholds, is frequently used to measure the performance of a detector.

The AUROC metric, derived from the Receiver Operating Characteristic curves, considers the true and false positive rates at different classification thresholds and is therefore suitable for evaluating classification performance at different thresholds. This is particularly important in scenarios where a specific false positive rate and error rate are required, such as with unbalanced datasets or binary classification tasks. Because the detection rate of zero-shot methods is highly threshold-dependent, the AUROC metric is frequently used to evaluate performance at different thresholds. The formula for AUROC is:

$$AUROC = \int_0^1 \frac{TP}{TP + FP} d \frac{FP}{FP + TN}$$

Maintaining a low false positive rate is essential; human-authored content should seldom be mislabeled as machine-generated [1]. Accordingly, we fix the FPR at 1% across all detection methods and tune the decision thresholds to this level when reporting accuracy.

Accuracy alone does not fully capture the effectiveness of attacks. It is also necessary to confirm that the original and paraphrased machine-generated texts preserve comparable meaning. To assess this, we employ P-SP [9], a state-of-the-art semantic similarity model based on embedding averaging and trained on a large, curated paraphrase corpus.

P-SP demonstrates strong calibration, performing reliably in both semantic evaluation benchmarks and plagiarism detection tasks. P-SP is

also robust against thematically similar, non-paraphrased texts. The average P-SP value for actual human-written and paraphrased text pairs is 0.76. If the P-SP value exceeds this average value of 0.76 for human paraphrasing, we assume that the meaning is largely preserved.

We used synthetic prompts to generate the dataset. We designed four prompt categories to mirror a representative cross-section of common assessment tasks in undergraduate computer science curricula. The categories covered are algorithm explanation, description of a technical concept, system design, and general problem solving.

– We focused on three language models for text generation:

– OPT-13B model, renowned for its unique architecture;

– text-davinci-003 variant from GPT-3.5 family, which has 175B parameters;

– GPT-4o mini, a cost-efficient version of GPT-4 with 8B parameters.

For all LMs, we sampled and truncated sequences before passing them through paraphrasers for the attack experiments. We truncated the context window to 512 tokens to constrain the model to using only short-range features. When necessary, we simply cropped the input to fit the context window.

We sampled texts produced by three LLMs and performed five paraphrasing iterations. We used three different neural network-based paraphrasers: DIPPER, DPO-Evader [12], and To-blend [25].

DIPPER is designed to be a semantic-preserving paraphraser. Its objective is to rewrite text while staying close in meaning and overall structure, typically with constraints on semantic similarity and fluency. As a result, many low-level regularities of LLM generation survive: sentence rhythms, discourse markers, token frequency profiles, and even some syntactic templates remain intact.

To-Blend, by contrast, is explicitly optimized to blend generated text into the human distribution. Rather than minimizing semantic distance alone, it targets distributional alignment: lexical choice, sentence length variance, discourse pacing, and stylistic heterogeneity. In other words, it attacks the assumptions detectors rely on, not just the surface form.

DPO-Evader is effective for a different and more fundamental reason: it explicitly optimizes against the detector itself. By framing evasion as a preference-learning or reinforcement-style objective, DPO-Evader trains

the paraphraser to minimize detector confidence rather than to merely preserve meaning or improve fluency.

After paraphrasing, we ensure that sequences have an equal number of words by truncating them to the length of the shortest one. Recursive paraphrasing is more effective in evading detection when compared to a single round of paraphrasing. Using automated evaluation techniques, we show that recursive paraphrasing method only degrades the text quality slightly most of the time.

We attack three open-source detectors: GPTZero, RoBERTa, Binoculars, using the default hyperparameters for each detector.

Table 3

Semantic similarity and AUROC scores of detectors

	Generated by	Semantic similarity	AUROC		
			GPTZero	RoBERTa	Binoculars
	OPT	-	91.1	95.1	93.6
	GPT-3.5	-	87.9	93.4	90.4
	GPT-4	-	80.9	89.3	90.2
Paraphrased by DIPPER	**OPT**	99.1	62.8	75.0	78.3
	GPT-3.5	97.6	60.5	72.8	75.0
	GPT-4	96.2	58.3	70.5	73.8
Paraphrased by DPO-Evader	**OPT**	94.8	55.7	65.7	68.1
	GPT-3.5	91.5	53.8	62.8	65.7
	GPT-4	91.2	51.5	60.6	63.3
Paraphrased by To-Blend	**OPT**	90.6	52.6	62.7	65.0
	GPT-3.5	89.0	51.5	61.5	63.8
	GPT-4	87.9	50.5	58.3	62.0

At a strict operating point of FPR = 1%, paraphrasing attacks reduce the true positive rate of most detectors to near-random levels, despite AUROC values that may appear moderately high. While paraphrasing causes a sharp drop in detection accuracy, AUROC reveals a more fundamental collapse in score separability, with several detectors approaching random discrimination under adversarial rewriting.

GPTZero AUROC collapses toward 50 under adversarial paraphrasing, which means no threshold can save it. Classifier-based detector loses

calibration before separability: AUROC drops slower than accuracy. Binoculars retains AUROC longer, because likelihood ratios encode deeper distributional differences, but semantic drift still lowers separability.

P-SP optimizes for semantic equivalence under paraphrastic variation, while detectors rely on distributional and stylistic signals orthogonal to semantics. Detector separability collapses even when paraphrases maintain >0.95 semantic similarity under a paraphrase-trained embedding model.

Table 4

Detectors performance at 1% FPR

	Generated by	Detection accuracy		
		GPTZero	**RoBERTa**	**Binoculars**
	OPT	69.5	88.1	82.0
	GPT-3.5	61.9	80.7	75.9
	GPT-4	38.5	64.7	74.1
Paraphrased by DIPPER	OPT	18.2	39.4	46.9
	GPT-3.5	12.5	33.7	38.8
	GPT-4	7.1	30.2	32.9
Paraphrased by DPO-Evader	OPT	5.7	19.2	26.3
	GPT-3.5	4.0	15.2	21.5
	GPT-4	2.8	13.1	19.0
Paraphrased by To-Blend	OPT	2.2	11.9	18.2
	GPT-3.5	1.6	9.4	13.5
	GPT-4	0.8	7.0	11.4

Paraphrasing significantly lowers detection accuracy while preserving input semantics. Under adversarial paraphrasing all detectors collapse, but GPTZero collapses first and overall yields the worst results. Binoculars degrades more slowly than GPTZero, but still fails under semantic drift and adversarial optimization. GPT-4 is hardest to detect even without paraphrasing, especially for metric-based detector.

These results show that AI-text detectors can be effectively attacked using recursive paraphrasing with only a slight degradation in text quality. Trends are similar for LLMs like GPT-4, for which paraphrasing reduces classifier accuracy.

6. Conclusions

This article provides an overview of the latest research on the detection of LLM-generated and LLM-paraphrased text, aiming to help researchers identify challenges and promising avenues for further research. We examined the mechanisms of text generation and paraphrasing, and highlighted recent breakthroughs. We assembled a dataset of generated and paraphrased sequences to attack three detectors, then evaluated the detectors' vulnerability to recursive paraphrasing attacks.

Paraphrasing represents the most prevalent and potent strategy for bypassing AI-text detectors, substantially degrading their effectiveness, even when relatively mild rewriting tools such as DIPPER are applied. Detector based on statistical metrics was the first to break down, since paraphrasing disrupts token-level likelihoods, burstiness measures, and rank-based features. The classifier-driven approach showed a more progressive decline, yet it still suffered marked reductions in class separability, especially under strict false-positive constraints. Notably, this deterioration persists despite strong semantic alignment between original and rewritten texts, suggesting that many current detectors depend more on superficial textual cues than on underlying meaning.

More forceful paraphrasing methods, including DPO-Evader and To-Blend, which are explicitly trained or tuned to avoid detection, generated distinctly different error distributions. These approaches not only weaken overall detection stability but also shift prediction patterns, moving a substantial portion of true positives into the range typically associated with genuinely human-authored content. As a result, even detectors capable of maintaining a moderate AUROC under attack became virtually unusable at realistic false-positive rates. At a false-positive rate of 1%, hit rates often approached randomness, even with high-performance supervised detector. This suggests that paraphrasing represents a structural weakness of post-hoc detection. As long as meaning is preserved and surface form is altered, detector signals can be systematically removed without affecting fluency or coherence.

We further examined how well meaning and overall quality are maintained under recursive paraphrasing of machine-generated text. DIPPER largely retained the original semantics, whereas To-Blend deliberately modified stylistic and distributional characteristics to more closely approximate

human writing, making it particularly effective at avoiding detection. In contrast, DPO-Evader showed a faster decline in semantic fidelity than DIPPER, yet it still degraded the performance of detectors operating under strict thresholds, despite only moderate semantic similarity in its outputs. More fundamentally, DPO-Evader highlights a structural vulnerability: any fixed detector with a stable scoring mechanism can be systematically circumvented once it is treated as an explicit optimization objective.

Future work should assess whether carefully engineered prompts, such as instructions that encourage a more human-like style, can also bypass current detection systems. In addition, the distinction between human and AI authorship warrants deeper examination. When text originally written by a person is rewritten by an LLM, labeling the result as purely human or purely machine-generated becomes conceptually problematic. Further research could develop a more nuanced classification system that goes beyond the simple binary of human-written and LLM-generated texts.

References:

1. Kirchenbauer, J., Geiping, J., Wen, Y., Shu, M., Saifullah, K., Kong, K., Fernando, K., Saha, A., Goldblum, M., & Goldstein, T. (2023). On the Reliability of Watermarks for Large Language Models. DOI: https://doi.org/10.48550/arXiv.2306.04634

2. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C.D., & Finn, C. (2023). DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. *International Conference on Machine Learning*. DOI: https://doi.org/10.48550/arXiv.2301.11305

3. Bao, G., Zhao, Y., Teng, Z., Yang, L., & Zhang, Y. (2023). Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature. DOI: https://doi.org/10.48550/arXiv.2310.05130

4. Hans, A., Schwarzschild, A., Cherepanova, V., Kazemi, H., Saha, A., Goldblum, M., Geiping, J., & Goldstein, T. (2024). Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text. DOI: https://doi.org/10.48550/arXiv.2401.12070

5. Verma, V. K., Fleisig, E., Tomlin, N., & Klein, D. (2023). Ghostbuster: Detecting Text Ghostwritten by Large Language Models. *North American Chapter of the Association for Computational Linguistics*. DOI: https://doi.org/10.48550/arXiv.2305.15047

6. Wu, J., Yang, S., Zhan, R., Yuan, Y., Wong, D.F., & Chao, L.S. (2023). A Survey on LLM-generated Text Detection: Necessity, Methods, and Future Directions. DOI: https://doi.org/10.48550/arXiv.2310.14724

7. Zhang, Y., Lv, P., Liu, Y., Ma, Y., Lu, W., Wang, X., Liu, X., & Liu, J. (2024). PersonaMark: Personalized LLM watermarking for model protection and user attribution. DOI: https://doi.org/10.48550/arXiv.2409.09739

8. Kumarage, T., Garland, J., Bhattacharjee, A., Trapeznikov, K., Ruston, S.W., & Liu, H. (2023). Stylometric Detection of AI-Generated Text in Twitter Timelines. DOI: https://doi.org/10.48550/arXiv.2303.03697

9. Krishna, K., Song, Y., Karpinska, M., Wieting, J., & Iyyer, M. (2023). Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. DOI: https://doi.org/10.48550/arXiv.2303.13408

10. He, W., Hou, B., Shang, T., Tarzanagh, D.A., Long, Q., & Shen, L. (2024). SEFD: Semantic-Enhanced Framework for Detecting LLM-Generated Text. *2024 IEEE International Conference on Big Data (BigData)*, 1309-1314. DOI: https://doi.org/10.48550/arXiv.2411.12764

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

12. Ayoobi, N., Knab, L., Cheng, W., Pantoja, D., Alikhani, H., Flamant, S., Kim, J., & Mukherjee, A. (2024). ESPERANTO: Evaluating Synthesized Phrases to Enhance Robustness in AI Detection for Text Origination. *Proceedings of the 36th ACM Conference on Hypertext and Social Media*. DOI: https://doi.org/10.1145/3720553.3746665

13. Nicks, C., Mitchell, E., Rafailov, R., Sharma, A., Manning, C.D., Finn, C., & Ermon, S. (2024). Language Model Detectors Are Easily Optimized Against. *International Conference on Learning Representations*. URL: https://www.semanticscholar.org/paper/Language-Model-Detectors-Are-Easily-Optimized-Nicks-Mitchell/13e6ab5bde4103e3128d409c3341ed16e30ac1d2

14. Hu, X., Chen, P., & Ho, T. (2023). RADAR: Robust AI-Text Detection via Adversarial Learning. DOI: https://doi.org/10.48550/arXiv.2307.03838

15. Sadasivan, V.S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). Can AI-Generated Text be Reliably Detected? DOI: https://doi.org/10.48550/arXiv.2303.11156

16. Peng, X., Zhou, Y., He, B., Sun, L., & Sun, Y. (2024). Hidding the Ghostwriters: An Adversarial Evaluation of AI-Generated Student Essay Detection. DOI: https://doi.org/10.48550/arXiv.2402.00412

17. Shi, Z., Wang, Y., Yin, F., Chen, X., Chang, K., & Hsieh, C. (2023). Red Teaming Language Model Detectors with Language Models. *Transactions of the Association for Computational Linguistics, 12*, 174-189. DOI: https://doi.org/10.48550/arXiv.2305.19713

18. Chakraborty, M., Tonmoy, S., Tonmoy, I., Mehedi, S.M., Sharma, K., Barman, N.R., Gupta, C., Gautam, S., Kumar, T., Jain, V., Chadha, A., Sheth, A.P., & Das, A. (2023). Counter Turing Test CT2: AI-Generated Text Detection is Not as Easy as You May Think - Introducing AI Detectability Index. *Conference on Empirical Methods in Natural Language Processing*. DOI: https://doi.org/10.48550/arXiv.2310.05030

19. Antoun, W., Mouilleron, V., Sagot, B., & Seddah, D. (2023). Towards a Robust Detection of Language Model-Generated Text: Is ChatGPT that easy to detect? DOI: https://doi.org/10.48550/arXiv.2306.05871

20. Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., & Wu, Y. (2023). How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. DOI: https://doi.org/10.48550/arXiv.2301.07597

21. Liu, Z., Yao, Z., Li, F., & Luo, B. (2023). On the Detectability of ChatGPT Content: Benchmarking, Methodology, and Evaluation through the Lens of Academic Writing. *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. DOI: https://doi.org/10.48550/arXiv.2306.05524

22. Schneider, S., Steuber, F., Schneider, J.A., & Rodosek, G.D. (2023). How well can machine-generated texts be identified and can language models be trained to avoid identification? DOI: https://doi.org/10.48550/arXiv.2310.16992

23. Koike, R., Kaneko, M., & Okazaki, N. (2023). OUTFOX: LLM-generated Essay Detection through In-context Learning with Adversarially Generated Examples. *AAAI Conference on Artificial Intelligence*. DOI: https://doi.org/10.48550/arXiv.2307.11729

24. Michail, A., Clematide, S., & Opitz, J. (2024). PARAPHRASUS: A Comprehensive Benchmark for Evaluating Paraphrase Detection Models. *International Conference on Computational Linguistics*. DOI: https://doi.org/10.48550/arXiv.2409.12060

25. Huang, F., Kwak, H., & An, J. (2024). ToBlend: Token-Level Blending With an Ensemble of LLMs to Attack AI-Generated Text Detection. DOI: https://doi.org/10.48550/arXiv.2402.11167