CAN AI-GENERATED TEXT BE RELIABLY DETECTED?

Anonymous authors
Paper under double-blind review

ABSTRACT

The rapid progress of Large Language Models (LLMs) has made them capable of performing astonishingly well on various tasks, including document completion and question answering. The unregulated use of these models, however, can potentially lead to malicious consequences such as plagiarism, generating fake news, spamming, etc. Therefore, reliable detection of AI-generated text can be critical to ensure the responsible use of LLMs. Recent works attempt to tackle this problem either using certain model signatures present in the generated text outputs or by applying watermarking techniques that imprint specific patterns onto them. In this paper, we show that these detectors are not reliable in practical scenarios. In particular, we develop a *recursive paraphrasing* attack to apply on AI text, which can break a whole range of detectors, including the ones using the watermarking schemes as well as neural network-based detectors, zero-shot classifiers, and retrieval-based detectors. Our experiments include passages around 300 tokens in length, showing the sensitivity of the detectors even in the case of relatively long passages. We also observe that our *recursive paraphrasing* only degrades text quality slightly, measured via perplexity scores and MTurk human study. Additionally, we show that even LLMs protected by watermarking schemes can be vulnerable against spoofing attacks aimed to mislead detectors to classify human-written text as AI-generated, potentially causing reputational damages to the developers. In particular, we show that an adversary can infer hidden AI text signatures of the LLM outputs without having white-box access to the detection method. Finally, we provide a theoretical connection between the AUROC of the best possible detector and the Total Variation distance between human and AI text distributions that can be used to study the fundamental hardness of the reliable detection problem for advanced language models.

1 INTRODUCTION

Artificial Intelligence (AI) has made tremendous advances in recent years, from generative models in computer vision (Rombach et al., 2022; Saharia et al., 2022) to generative models in natural language processing (NLP) (Brown et al., 2020; Zhang et al., 2022; Raffel et al., 2019). Large Language Models (LLMs) can now generate texts of supreme quality with the potential in many applications. For example, the recent model of ChatGPT (OpenAI, 2022) can generate human-like texts for various tasks such as writing codes for computer programs, lyrics for songs, completing documents, and question answering; its applications are endless. The trend in NLP shows that these LLMs will even get better with time. However, this comes with a significant challenge in terms of authenticity and regulations. AI tools have the potential to be misused by users for unethical purposes such as plagiarism, generating fake news, spamming, generating fake product reviews, and manipulating web content for social engineering in ways that can have negative impacts on society (Adelani et al., 2020; Weiss, 2019). Some news articles rewritten by AI have led to many fundamental errors in them (Christian, 2023). Hence, there is a need to ensure the responsible use of these generative AI tools. In order to aid this, a lot of recent research focuses on detecting AI-generated texts.

Several detection works study this problem as a binary classification problem (OpenAI, 2019; Jawahar et al., 2020; Mitchell et al., 2023; Bakhtin et al., 2019; Fagni et al., 2020) and use **neural network-based detectors**. For example, OpenAI fine-tunes RoBERTa-based (Liu et al., 2019) GPT-2 detector models to distinguish between non-AI generated and GPT-2 generated texts (OpenAI, 2019). This requires such a detector to be fine-tuned with supervision on each new LLM for reliable detection.

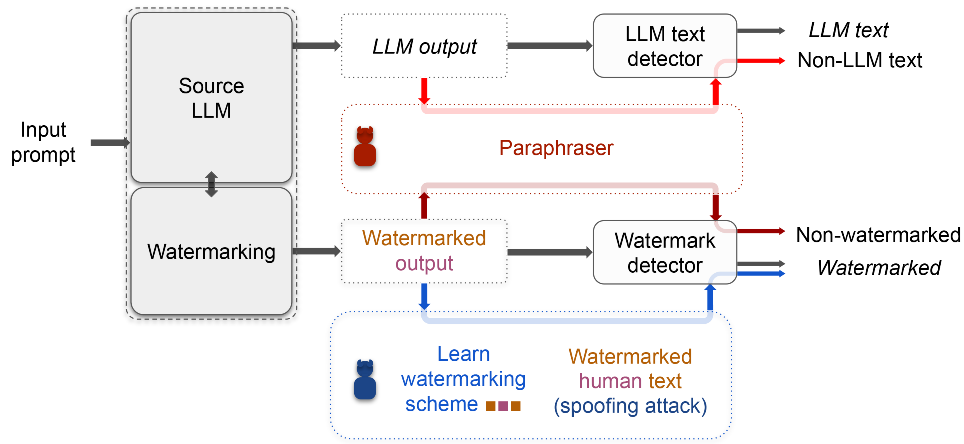


Figure 1: An illustration of vulnerabilities of existing AI-text detectors. We consider both watermarking-based and non-watermarking-based detectors and show that they are not reliable in practical scenarios. Colored arrow paths show the potential pipelines for adversaries to avoid detection. In red: an attacker can use a paraphraser to remove the LLM signatures from an AI-generated text to avoid detection. In blue: an adversary can query the watermarked LLM multiple times to learn its watermarking scheme. This information can be used to spoof the watermark detector.

Another stream of work focuses on **zero-shot AI text detection** without any additional training overhead (Solaiman et al., 2019; Ippolito et al., 2019; Gehrmann et al., 2019). These works evaluate the expected per-token log probability of texts and perform thresholding to detect AI-generated texts. Mitchell et al. (2023) observe that AI-generated passages tend to lie in negative curvature of log probability of texts. They propose DetectGPT, a zero-shot LLM text detection method, to leverage this observation. Since these approaches rely on a neural network for their detection, they can be vulnerable to adversarial and poisoning attacks (Goodfellow et al., 2014; Sadasivan et al., 2023; Kumar et al., 2022; Wang et al., 2022). Another line of work aims to **watermark AI-generated texts** to ease their detection (Atallah et al., 2001; Wilson et al., 2014; Kirchenbauer et al., 2023a; Zhao et al., 2023). Watermarking eases the detection of LLM output text by imprinting specific patterns on them. Soft watermarking proposed in Kirchenbauer et al. (2023a) partitions tokens into "green" and "red" lists, as they define, to help create these patterns. A watermarked LLM samples a token, with high probability, from the green list determined by a pseudo-random generator seeded by its prefix token. The watermarking detector would classify a passage with a large number of tokens from the green list as AI-generated. These watermarks are often imperceptible to humans. Krishna et al. (2023) introduces an **information retrieval-based detector** by storing the outputs of the LLM in a database. For a candidate passage, their algorithm searches this database for semantically similar matches for detection. However, storing user-LLM conversations might cause serious privacy concerns.

In this paper, through several experiments, we show that these state-of-the-art AI-text detectors are unreliable in practical scenarios (Wolff, 2020; Aaronson, 2022; Liang et al., 2023; Pu et al., 2023; Wang et al., 2023). In §2, we have developed a *recursive paraphrasing attack* that use neural network-based paraphrasing to recursively paraphrase the source LLM's output text. Our experiments show that this automated paraphrasing attack can drastically reduce the accuracy of various detectors, including those using soft watermarking (Kirchenbauer et al., 2023a), to increase *type-II error* (detecting AI text as human text). For instance, **our recursive paraphrasing attack on watermarked texts, even over relatively long passages of 300 tokens in length, can drop the detection rate (true positive rate at 1% false positive rate or TPR@1%FPR) from 99.3% to 4.0% with only degradation of 1.5 in perplexity score.** We note that Kirchenbauer et al. (2023a) considers a relatively weak paraphrasing attack in their experiments where they perform span replacement by replacing random tokens (in-place) using an LLM. Our experiments, however, show the vulnerability of the watermarking scheme against stronger paraphrasing attacks that we use.

We also observe that the quality of the paraphrased passages degrades, but only slightly, compared to the original ones. We quantify this both via perplexity score evaluation as well as via MTurk human evaluation study. In particular, our human evaluation study shows that 70% of the recursive paraphrased passages are rated high quality in terms of content preservation, and 89% of them are rated high quality in terms of grammar or text quality.

After paraphrasing, the area under the receiver operating characteristic (AUROC) curves of zero-shot detectors (Mitchell et al., 2023) drops from 96.5% to 25.2%. We also observe that the performance of neural network-based trained detectors (OpenAI, 2019) deteriorates significantly after our paraphrasing attack. For instance, the TPR@1%FPR of the RoBERTa-Large-Detector from OpenAI drops from 100% to 60% after paraphrasing. In addition, we show that the retrieval-based detector by Krishna et al. (2023) designed to evade paraphrase attacks is vulnerable to our recursive paraphrasing. In fact, the accuracy of their detector falls from 100% to 25% with our recursive paraphrase attack.

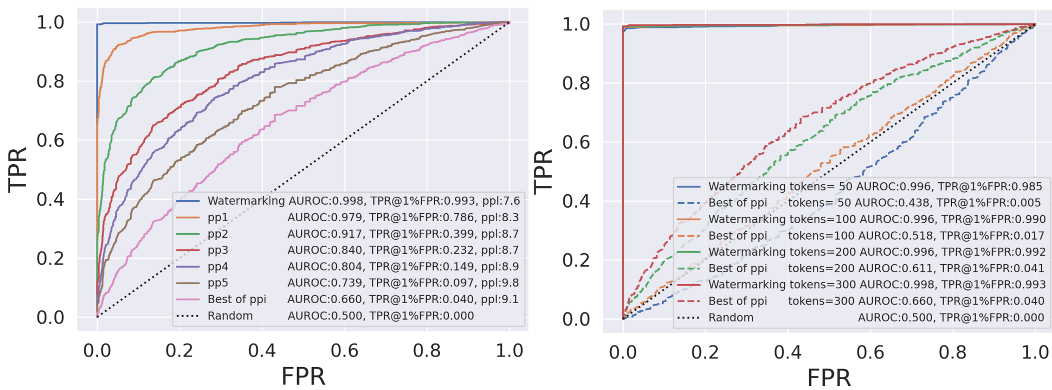
Moreover, we show the possibility of **spoofing attacks** on various AI text detectors in §3. In this setting, an attacker generates a non-AI text that is detected to be AI-generated, thus increasing *type-I error* (falsely detecting human text as AI text). An adversary can potentially launch spoofing attacks to produce derogatory texts that are detected to be AI-generated to affect the reputation of the target LLM's developers. In particular, we show that an adversary can infer hidden AI text signatures without having white-box access to the detection method. For example, though the pseudo-random generator used for generating watermarked text is private, we develop an attack that adaptively queries the target LLM multiple times to learn its watermarking scheme. An *adversarial human* can then use this information to compose texts that are detected to be watermarked. Figure 1 shows an illustration of some of the vulnerabilities of the existing AI-text detectors.

Finally, in §4, we present a theoretical result regarding the hardness of AI-text detection. Our main result in Theorem 1 states that the AUROC of the best possible detector differentiating two distributions $\mathcal{H}$ (e.g., human text) and $\mathcal{M}$ (e.g., AI-generated text) reduces as the total variation distance $\mathsf{TV}(\mathcal{M}, \mathcal{H})$ between them decreases. Note that this result is true for any two arbitrary distributions $\mathcal{H}$ and $\mathcal{M}$. For example, $\mathcal{H}$ could be the text distribution for a person or group and $\mathcal{M}$ could be the output text distribution of a general LLM or an LLM trained by an adversary to mimic the text of a particular set of people. Essentially, adversaries can train LLMs to mimic human text as they get more sophisticated, potentially reducing the TV distance between human and AI text, leading to an increasingly more difficult detection problem according to our Theorem 1. Although estimating the exact TV between text distributions from a finite set of samples is a challenging problem, we provide some empirical evidence, over simulated data or via TV estimations, showing that more advanced LLMs can potentially lead to smaller TV distances. Thus, our Theorem 1 would indicate an increasingly more difficult reliable detection problem in such cases.

Identifying AI-generated text is a critical problem to avoid its misuse by users for unethical purposes such as plagiarism, generating fake news, and spamming. However, deploying vulnerable detectors may *not* be the right solution to tackle this issue since it can cause its own damages, such as falsely accusing a human of plagiarism. Our results highlight the sensitivities of a wide range of detectors to both evasion and spoofing attacks and indicate the difficulty of developing reliable detectors in practical scenarios — to maintain reliable detection performance, LLMs would have to trade off their performance. We hope that these findings can help the ethical and dependable utilization of AI-generated text.

In summary, we make the following contributions in this work.

- Our work is the first to comprehensively analyze the performance of four different classes of detectors, including watermarking-based, neural network-based, zero-shot, and retrieval-based detectors, and reveal their reliability issues (in §2). In particular, the *recursive paraphrasing attack* that we develop is the first method that can break watermarking (Kirchenbauer et al., 2023a) and retrieval-based (Krishna et al., 2023) detectors with only a small degradation in text quality.
- Our work is the first to show that existing detectors are vulnerable against *spoofing attacks* where an adversarial human aims to write a (potentially derogatory) passage falsely detected as AI-generated *without* having a white-box access to the detection methods (in §3). For instance, we show that an adversary can infer the watermarking signatures by probing the watermarked LLM and analyzing the statistics of the generated tokens.
- Our work is the first to establish a theoretical connection between the AUROC of the best possible detector and the TV distance between human and AI-text distributions that can be used to study the hardness of the reliable text detection problem (in §4).



(a) Watermarked text with mean token length 300 (b) Watermarked text with varying token lengths

Figure 2: ROC plots for soft watermarking with recursive paraphrasing attacks. AUROC, TPR@1%FPR, and perplexity scores measured using OPT-13B are given in the legend. (a) Even for 300 tokens long watermarked passages, recursive paraphrasing is effective. As paraphrasing rounds proceed, detection rates degrade significantly with a slight trade-off in text quality. (b) Attacking watermarked passages become easier as their length reduces.

ppi		i=1	i=2	i=3	i=4	i=5	All ppi
Content preservation	Avg. rating	4.0 ± 0.8	4.1 ± 0.8	3.9 ± 0.9	4.2 ± 0.9	3.7 ± 1.1	$\mathbf{4.0 \pm 0.9}$
	Ratings 5&4	70.2%	77.2%	63.2%	80.0%	61.4%	**70.4%**
Grammar or text quality	Avg. rating	4.28 ± 0.67	4.12 ± 0.50	4.12 ± 0.53	4.11 ± 0.64	4.07 ± 0.53	$\mathbf{4.14 \pm 0.58}$
	Ratings 5&4	87.72%	92.98%	91.23%	84.21%	89.47%	**89.12%**

Table 1: Summary of the MTurk human evaluation study on content preservation and grammar or text quality of the recursive paraphrases that we use for our attacks. Ratings are on a Likert scale of 1 to 5. See Appendix B.1 for details.

2 Evading AI-Detectors using Paraphrasing Attacks

2.1 Paraphraser Models and Text Dataset

We use the "document" features of the XSum dataset (Narayan et al., 2018) containing long news articles for our experiments. We use two different neural network-based paraphrases – DIPPER with 11B parameters (Krishna et al., 2023), and T5-based paraphraser (Damodaran, 2021) with 222M parameters. Suppose a passage $S = (s_1, s_2, ..., s_n)$ where s_i is the i^{th} sentence. DIPPER paraphrases S to be $S' = f_{dipper}(S)$ in one-shot while the light-weight T5-based paraphraser would output $S' = (f_{t5}(s_1), f_{t5}(s_2), ..., f_{t5}(s_n))$ where they can only paraphrase sentence-by-sentence. DIPPER also has the ability to input a context prompt text C to generate higher-quality paraphrasing $S' = f_{dipper}(S, C)$. We can also vary two different hyperparameters of DIPPER to generate a diverse number of paraphrases for a single input passage.

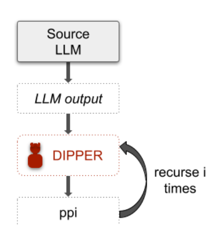


Figure 3: Recursive paraphrasing

We use DIPPER for recursive paraphrasing attacks since it provides high-quality paraphrasing (Krishna et al., 2023). Let an LLM L generate AI text output $S = L(C)$ for an input prompt C. DIPPER can be used to generate a paraphrase `pp1` $= f_{dipper}(S, C)$. This paraphrasing can be performed in recursion (see Figure 3). That is, `ppi` $= f_{dipper}($`pp(i-1)`$, C)$. In the next section, we show that recursive paraphrasing is effective in removing watermarks from AI text when compared to a single round of paraphrasing. Moreover, by quantifying text quality with perplexity measured using OPT-13B as well as performing a human study, we show that our recursive paraphrasing method only degrades the text quality slightly.

2.2 Paraphrasing Attacks on Watermarked AI Text

In this section, we evaluate our recursive paraphrasing attacks on the soft watermarking scheme proposed in Kirchenbauer et al. (2023a). Soft watermarking encourages LLMs to output token $s^{(t)}$ at time-step t that belongs to a "green list". The green list for $s^{(t)}$ is created using a private pseudo-random generator that is seeded with the prior token $s^{(t-1)}$. A watermarked output from the LLM is designed to have tokens that are majorly selected from the green list. Hence, a watermark detector with the pseudo-random generator checks the number of *green* tokens in a candidate passage to detect whether it is watermarked or not. Here, we target a watermarked OPT-1.3B (Zhang et al., 2022) with 1.3B parameters for our experiments.

Dataset. We perform our experiments on 2000 text passages that are around 300 tokens in length (1000 passages per human and AI text classes). We pick 1000 long news articles from the XSum "document" feature. For each article, the first 300 tokens are input to the target OPT-1.3B to generate 1000 watermarked AI text passages that are each 300 tokens in length. The second 300 tokens from the 1000 news articles in the dataset are treated as baseline human text. We note that our considered dataset has more and longer passages compared to the experiments in Kirchenbauer et al. (2023a).

Detection results after paraphrasing attack. After a single round of paraphrasing (`pp1`), TPR@1%FPR of watermark detector only degrades from 99.3% to 78.6%. Similarly, another weaker paraphrasing attack discussed in Kirchenbauer et al. (2023a) is not effective in removing watermarks. They perform "span replacement" by replacing random tokens (in-place) using a language model. However, we show that our stronger recursive paraphrasing attack can effectively evade watermark detectors with only a slight degradation in text quality. As shown in Figure 2a, the recursive paraphrase attack further degrades the detection rate of the detector to 9.7% after 5 rounds of paraphrasing (`pp5`). `Best of ppi` in the figure refers to the method where, for each passage, we select the paraphrase out of all the `ppi`'s that has the worst detector score. For `Best of ppi`, **the detection rate reduces drastically from 99.8% to 4.0% with only a trade-off of 1.5 in the perplexity score.** Figure 2b shows that the watermarking detector becomes weaker as the length of the watermarked text reduces. Note that for watermarked texts that are 50 or 100 tokens long, the detection performance after the recursive paraphrasing attack is similar to that of a random detector. We provide examples of paraphrased text that we use for our attacks in Appendix B.2.

Quality of the paraphrased passages. In order to reliably study the quality of the paraphrases we use in our experiments, we perform two human evaluations using MTurk. A summary of the study is provided in Table 1. We investigate the content preservation and text quality or grammar of the recursive paraphrases with respect to the watermarked texts (see Tables 4 and 5 in Appendix B.1 for more details). **In terms of content preservation, 70$\%$ of the paraphrases were rated high quality and 23$\%$ somewhat equivalent. In terms of text quality or grammar, 89$\%$ of the paraphrases were rated high quality.** On a Likert scale of 1 to 5, the paraphrases that we use received an average rating of 4.14 ± 0.58 for text quality or grammar and 4.0 ± 0.9 for content preservation. These results confirm that watermarking can be effectively attacked using recursive paraphrasing with only a slight degradation in text quality. See Appendix B.1 for more details on the human study.

2.3 Paraphrasing Attacks on Non-Watermarked AI Text

Neural network-based trained detectors such as RoBERTa-Large-Detector from OpenAI (OpenAI, 2019) are trained or fine-tuned for binary classification with datasets containing human and AI-generated texts. Zero-shot classifiers leverage specific statistical properties of the source LLM outputs for their detection. Retrieval-based methods search for a candidate passage in a database that stores the LLM outputs. Here, we perform experiments on these non-watermarking detectors to show they are vulnerable to our paraphrasing attack.

Trained and Zero-shot detectors. We use a pre-trained GPT-2 Medium model (Radford et al., 2019) with 355M parameters as the target LLM to evaluate our attack on 1000 long passages from the XSum dataset (Narayan et al., 2018). We use the T5-based paraphrasing model (Damodaran, 2021) with 222M parameters to rephrase the 1000 output texts generated using the target GPT-2 Medium

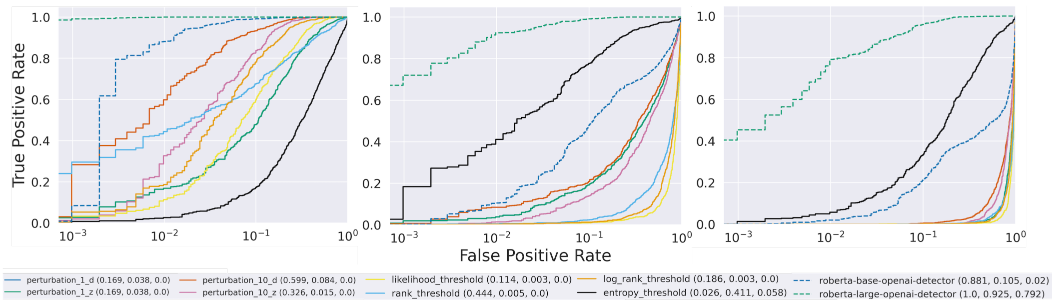


Figure 4: ROC curves for various trained and zero-shot detectors. **Left:** Without attack. **Middle:** After paraphrasing attack using T5-based paraphraser. The performance of zero-shot detectors drops significantly. **Right:** Here, we assume we can query the detector ten times for the paraphrasing attack. We generate ten paraphrasings for each passage and query multiple times to evade detection. Notice how all detectors have low TPR@1%FPR. In the plot legend – `perturbation` refers to the zero-shot methods in Mitchell et al. (2023); `threshold` refers to the zero-shot methods in Solaiman et al. (2019); Gehrmann et al. (2019); Ippolito et al. (2019); `roberta` refers to OpenAI's trained detectors (OpenAI, 2019). The TPR@1%FPR scores of different detectors before the attack, after the attack, and after the attack with multiple queries, respectively, are provided in the plot legend.

model. Figure 4 shows the effectiveness of the paraphrasing attack over these detectors. **The AUROC scores of DetectGPT (Mitchell et al., 2023) drop from 96.5% (before the attack) to 59.8% (after the attack).** Note that AUROC of 50% corresponds to a random detector. The rest of the zero-shot detectors (Solaiman et al., 2019; Gehrmann et al., 2019; Ippolito et al., 2019) also perform poorly after our attack. Though the performance of the trained neural network-based detectors (OpenAI, 2019) is better than that of zero-shot detectors, they are also not reliable. For example,TPR@1%FPR of OpenAI's RoBERTa-Large-Detector drops from 100% to around 92% after our attack.

In another setting, we assume the attacker may have multiple access to the detector. That is, the attacker can query the detector with an input AI text passage, and the detector would reveal the detection score to the attacker. For this scenario, we generate ten different paraphrases for an input passage and query the detector for the detection scores. For each AI text passage, we then select the paraphrase with the worst detection score for evaluating the ROC curves. As shown in Figure 4, **with multiple queries to the detector, an adversary can paraphrase more efficiently to bring down TPR@1%FPR of the RoBERTa-Large-Detector from 100% to 80%.**

Retrieval-based detectors. Detector in Krishna et al. (2023) is designed to be robust against paraphrase attacks. However, we show that they can suffer from the recursive paraphrase attacks that we develop using DIPPER. We use 2000 passages (1000 generated by OPT-1.3B and 1000 human passages) from the XSum dataset. AI outputs are stored in the AI database by the detector. As shown in Figure 5, this detector detects almost all of the AI outputs even after a round of paraphrasing. However, **the detection accuracy drops to 60% after five rounds of recursive paraphrasing.** As marked in the plot, the perplexity score of the paraphrased text only degrades by 1.7 at a detection accuracy of below 60%. Moreover, retrieval-based detectors are concerning since they might lead to **serious privacy issues** from storing users' LLM conversations.

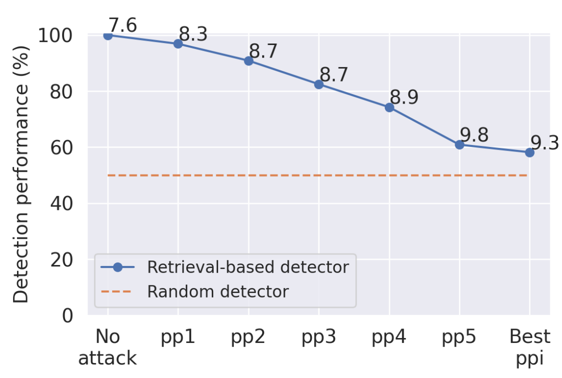


Figure 5: Recursive paraphrasing breaks the retrieval-based detector (Krishna et al., 2023) with only slight degradation in text quality. `ppi` refers to `i` recursion(s) of paraphrasing. Numbers next to markers denote the perplexity scores of the paraphraser output.

3 Spoofing Attacks on Generative AI-text Models

An AI language detector without a low type-I error can cause harm as it might wrongly accuse a human of plagiarizing using an LLM. Moreover, an attacker (*adversarial human*) can generate a non-AI text to be detected as AI-generated. This is called the *spoofing attack*. An adversary can potentially launch spoofing attacks to produce derogatory texts to damage the reputation of the target LLM's developers. In this section, as a proof-of-concept, we show that current text detectors can be spoofed to detect texts composed by adversarial humans as AI-generated. More details on the spoofing experiments are presented in Appendix D.

Soft watermarking. As discussed in §2, soft watermarked LLMs (Kirchenbauer et al., 2023a) generate tokens from the "green list" that are determined by a pseudo-random generator seeded by the prefix token. Though the pseudo-random generator is private, an attacker can estimate the green lists by observing multiple token pairs in the watermarked texts from the target LLM. An adversarial human can then leverage the estimated green lists to compose texts by themselves that are detected to be watermarked. In our experiments, we estimate the green lists for 181 most commonly used words in the English vocabulary. We query the target watermarked OPT-1.3B model one million times to observe the token pair distributions within this smaller vocabulary subset we select. Based on the frequency of tokens that follow a prefix token in the observed generative outputs, we estimate green lists for each of the 181 common words. We build a tool that helps adversarial humans create watermarked sentences by providing them with the proxy green list we learn. We observe that the **soft watermarking scheme can be spoofed to degrade its detection AUROC from 99.8% to 1.3%** (see Figure 6).

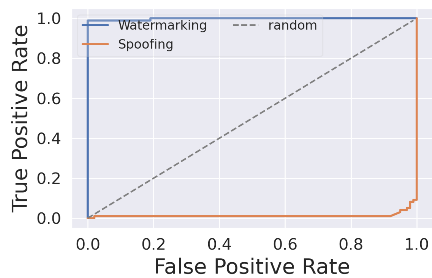


Figure 6: ROC curve of a soft watermarking-based detector (Kirchenbauer et al., 2023a) after our spoofing attack.

Retrieval-based detectors. Krishna et al. (2023) use a database to store LLM outputs to detect AI-text by retrieval. We find in our experiments (see Figure 12) that an **adversary can spoof this detector 100% of the time, even if the detector maintains a private database**. Suppose an adversary, say a teacher, has access to a human written document S, say a student's essay. The adversary can prompt the target LLM to paraphrase S to get S'. This results in the LLM, by design, storing its output S' in its private database for detection purposes. Now, the detector would classify the original human text S as AI-generated since a semantically similar copy S' is present in its database. In this manner, a teacher can purposefully allege an innocent student to have plagiarised using the retrieval-based detector.

Zero-shot and neural network-based detectors. In this setting, a malicious adversary could write a short text in a collaborative work, which may lead to the entire text being classified as AI-generated. To simulate this, we prepend a human-written text marked as AI-generated by the detector to all the other human-generated text for spoofing. In other words, from 200 long passages in the XSum dataset, we pick the human text with the worst detection score for each detector considered in §2.3. We then prepend this text to all the other human texts, ensuring that the length of the prepended text does not exceed the length of the original text. Our experiments show that the **AUROC of all these detectors drops after spoofing** (see plots in Appendix D). After this naïve spoofing attack, the TPR@1%FPR of most of these detectors drop significantly.

4 Hardness of Reliable AI Text Detection

In this section, we formally upper bound the AUROC of an arbitrary detector in terms of the TV between the distributions for $\mathcal{M}$ (e.g., AI text) and $\mathcal{H}$ (e.g., human text) over the set of all possible text sequences Ω. We note that this result holds for any two arbitrary distributions $\mathcal{H}$ and $\mathcal{M}$. For example, $\mathcal{H}$ could be the text distribution for a person or group, while $\mathcal{M}$ could be the output text distribution of a general LLM or an LLM trained by an adversary to mimic the text of a particular set of people.

We use $\mathsf{TV}(\mathcal{M}, \mathcal{H})$ to denote the TV between these two distributions and model a detector as a function $D : \Omega \to \mathbb{R}$ that maps every sequence in Ω to a real number. Sequences are classified into AI-generated or human-generated by applying a threshold γ on this number. By adjusting the parameter γ, we can tune the sensitivity of the detector to AI and human-generated texts to obtain an ROC curve.

Theorem 1. *The area under the ROC of any detector D is bounded as*

$$\mathsf{AUROC}(D) \leq \frac{1}{2} + \mathsf{TV}(\mathcal{M}, \mathcal{H}) - \frac{\mathsf{TV}(\mathcal{M}, \mathcal{H})^2}{2}.$$

The proof is deferred to Appendix C.1. Figure 7 shows how the above bound grows as a function of the TV distance. This theorem states that as the TV distance between AI and human text distributions reduces, the AUROC of the best possible detector decreases. Based on our theory, an adversary can use advanced LLMs to mimic human text to reduce the TV distance between human and AI text distributions to evade text detection systems.

For a detector to have a good performance (say, AUROC > 0.9), the distributions of human and AI-generated texts must be very different from each other (TV > 0.5 based on the figure). As $\mathcal{M}$ gets more similar to $\mathcal{H}$ (say, TV < 0.2), the performance of even the best-possible detector becomes unreliable (AUROC < 0.7). For some applications, say AI-text plagiarism, reliable detection should have a low false positive rate (say, < 0.01) and a high true positive rate (say, > 0.9). Based on our theory, this cannot be achieved even when the overlap between the distributions is relatively low, say 11% (or TV $= 0.9 - 0.01 = 0.89$, based on equation 1 in Appendix C.1).

Figure 7: Comparing the performance, in terms of AUROC, of the best possible detector to that of the baseline performance corresponding to a random classifier.

Note that, for a watermarked model, the above bound can be close to one as the TV between the watermarked distribution and human-generated distribution can be high. Corollary 1 in Appendix C.2 discusses how paraphrasing attacks can be effective in evading watermarks using Theorem 1. In Appendix C.3, we also present a tightness analysis of the bound in Theorem 1, where we show that for any distribution $\mathcal{H}$ there exists $\mathcal{M}$ and a detector D for which the bound holds with equality. We also discuss general trade-offs between true positive and false positive rates of detection in Corollaries 2 and 3 in Appendix C.2. Theorem 2 in Appendix C.4 extends Theorem 1 to bound the AUROC of the best possible detector by a function of the TV distance between LLM outputs generated using pseudorandomness and human text distributions.

In studying the hardness of the detection problem, we consider the following assumption that for a given human-text distribution $\mathcal{H}$, more advanced LLMs mimicking $\mathcal{H}$ can lead to smaller TV. Thus, using Theorem 1, the detection problem becomes increasingly more difficult. This is the core argument of our hardness result on AI text detection. Although the underlying assumption seems to be intuitive given the capabilities of LLMs such as GPT-4 (OpenAI, 2023), a precise analysis of this assumption is quite difficult because estimating the true TV of the text distributions from a finite set of samples is extremely challenging. Nevertheless, we provide some empirical evidence supporting this assumption using two sets of experiments. In all the experiments, we consistently observe that the TV distance estimates between human and AI text distributions reduce as language models get more advanced, indicating the increasing difficulty associated with AI text detection.

(i) Using synthetic text data. We perform experiments on a toy synthetic text dataset where the exact TV distance can be calculated. We use the Markov assumption to generate the synthetic text data with sequence length 3 using a randomly generated token transition matrix for varying vocabulary sizes. We use single-layer LSTMs of different hidden unit sizes to train on a dataset of size 20,000 sampled from this synthetic data distribution using a default AdamW optimizer (Loshchilov & Hutter, 2017). We compute the learned token transition matrix for the LSTM output distribution using the softmax logit values of the trained model. Using transition matrices of both distributions, we compute the exact TV. Figure 8 shows that the exact TV distances between the learned and true synthetic distributions reduce as the LSTM model size increases.

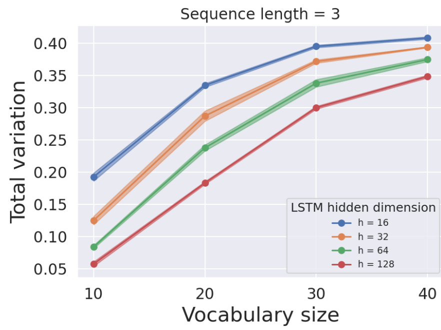


Figure 8: Increasing model size reduces the exact TV between the true synthetic data distribution and the learned distribution in all settings. Error bars report standard deviations after 5 independent trials.

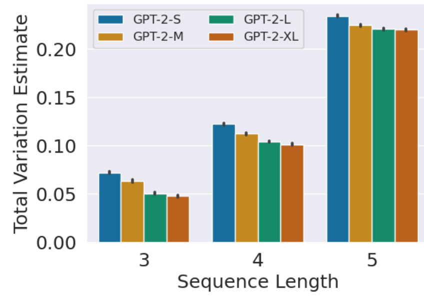


Figure 9: Estimated TV distances of GPT-2 output datasets from the WebText dataset using meta-token sequences of varying lengths. TV decreases with model size for each length.

(ii) Using projection. For discrete distributions, the TV distance can be computed as 1/2 of the sum of the point-wise differences between their probability density functions (PDFs). While this is mathematically simple since texts can be considered as token sequences with bounded length, it is not practical to compute true TV distances directly through estimating PDFs due to the size of the sample space, which is approximately *the size of the token set* to the power of *sequence length*. To tackle this issue, we split the original token set into five roughly equal partitions and assign a meta-token to each partition. Given a sequence of tokens from the original set, we construct a new sequence by replacing each token with the corresponding meta-token. We estimate the PDFs of the sequences of meta-tokens created using texts from the WebText and GPT-2 output datasets. Since the set of meta-tokens is significantly smaller than the original token set, estimating PDFs becomes much more tractable. We then use these PDFs to estimate the total variation distances of the output distributions of different GPT-2 models (GPT-2-Small, GPT-2-Medium, GPT-2-Large, and GPT-2-XL) from the WebText dataset. Figure 9 plots these TV estimates for different sequence lengths, averaged over 30 runs of the experiment. We observe that the TV distance consistently decreases with increasing model size for all sequence lengths.

These experiments provide empirical evidence that more advanced LLMs can lead to smaller TV distances. Thus, based on Theorem 1, reliable AI text detection would become increasingly difficult.

5 Conclusion

In this paper, we analyze the performance of four different classes of detectors including watermarking-based, neural-net based, zero-shot based and retrieval-based detectors and show their reliability issues. In particular, we develop a strong attack called *recursive paraphrasing* that can break recently proposed watermarking and retrieval-based detectors. Using perplexity score computation as well as conducting various MTurk human study, we observe that our recursive paraphrasing only degrades text quality slightly. We also show that adversaries can spoof these detectors to increase their type-I errors. Spoofing attacks can lead to the generation of derogatory passages detected as AI-generated that might affect the reputation of the LLM detector developers. Finally, we establish a theoretical connection between the AUROC of the best possible detector to the TV distance between human and AI-text distributions that can be used to study the fundamental hardness of the reliable detection problem for more advanced LLMs.

A detector should ideally be helpful in reliably flagging AI-generated texts to prevent the misuse of LLMs. However, the cost of misidentification by a detector can be huge. If the false positive rate of the detector is not low enough, humans (e.g., students) could be falsely accused of AI plagiarism. Moreover, a disparaging passage falsely detected to be AI-generated could affect the reputation of the LLM's developers. As a result, the practical applications of AI-text detectors can become unreliable and invalid. Security methods need not be foolproof. However, we need to make sure that it is not an easy task for an attacker to break these security defenses. Thus, analyzing the risks of using current and future detectors can be vital to avoid creating a false sense of security.

References

Scott Aaronson. My ai safety lecture for ut effective altruism. November 2022. URL https://scottaaronson.blog/?p=6823.

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In *Advanced Information Networking and Applications: Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020)*, pp. 1341–1354. Springer, 2020.

Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*, pp. 185–200. Springer, 2001.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*, 2019.

Lenore Blum, Manuel Blum, and Mike Shub. Comparison of two pseudo-random number generators. In *Advances in Cryptology: Proceedings of CRYPTO '82*, pp. 61–78. Plenum, 1982.

Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. doi: 10.1137/0213053. URL https://doi.org/10.1137/0213053.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. On the possibilities of ai-generated text detection, 2023.

Jon Christian. Cnet secretly used ai on articles that didn't disclose that fact, staff say. January 2023. URL https://futurism.com/cnet-ai-articles-label.

Prithiviraj Damodaran. Parrot: Paraphrase generation for nlu., 2021.

T Fagni, F Falchi, M Gambini, A Martella, and M Tesconi. Tweepfake: About detecting deepfake tweets. arxiv. *arXiv preprint arXiv:2008.00036*, 2020.

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

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

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

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

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259.

Kafkai. "kafkai: Ai writer & ai content generator". 2020. URL https://kafkai.com/.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023a.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models, 2023b.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense, 2023.

Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Certifying model accuracy under distribution shifts. *arXiv preprint arXiv:2201.12440*, 2022.

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

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*, 2023.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745, 2018.

OpenAI. Gpt-2: 1.5b release. November 2019. URL https://openai.com/research/gpt-2-1-5b-release.

OpenAI. Chatgpt: Optimizing language models for dialogue. November 2022. URL https://openai.com/blog/chatgpt/.

OpenAI. Gpt-4 technical report. March 2023. URL https://cdn.openai.com/papers/gpt-4.pdf.

J. Pu, Z. Sarwar, S. Abdullah, A. Rehman, Y. Kim, P. Bhattacharya, M. Javed, and B. Viswanath. Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1613–1630, Los Alamitos, CA, USA, may 2023. IEEE Computer Society. doi: 10.1109/SP46215.2023.10179387. URL https://doi.ieeecomputersociety.org/10.1109/SP46215.2023.10179387.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. URL https://arxiv.org/abs/1910.10683.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Vinu Sankar Sadasivan, Mahdi Soltanolkotabi, and Soheil Feizi. Cuda: Convolution-based unlearnable datasets. *arXiv preprint arXiv:2303.04278*, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

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

Wenxiao Wang, Alexander J Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In *International Conference on Machine Learning*, pp. 22769–22783. PMLR, 2022.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection, 2023.

Max Weiss. Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions. *Technology Science*, 2019121801, 2019.

Alex Wilson, Phil Blunsom, and Andrew D Ker. Linguistic steganography on twitter: hierarchical language modeling with manual interaction. In *Media Watermarking, Security, and Forensics 2014*, volume 9028, pp. 9–25. SPIE, 2014.

Max Wolff. Attacking neural text detectors. *CoRR*, abs/2002.11768, 2020. URL https://arxiv.org/abs/2002.11768.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. Protecting language generation models via invisible watermarking. *arXiv preprint arXiv:2302.03162*, 2023.

A EXPERIMENTS WITH MORE DATASETS AND MODELS

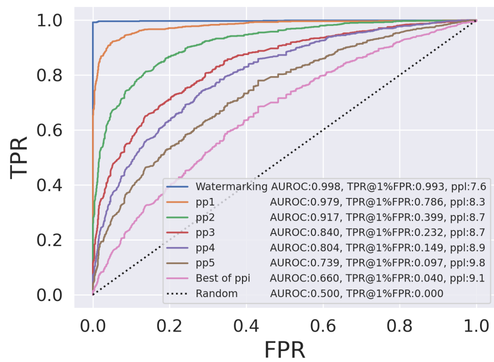
In this section, we consider multiple datasets (XSum (Narayan et al., 2018), PubMedQA (Jin et al., 2019), and Kafkai (Pu et al., 2023)) and target LLMs (OPT-1.3B (Zhang et al., 2022) and GPT-2-Medium (Radford et al., 2019)) for analyzing our attacks.

Datasets. As discussed in the § 2.1, we use 2000 text passages (1000 passages each for human and AI-generated text classes) of $\sim$300 tokens in length from the XSum dataset for analyzing our attacks. For the rest of the datasets, we use 1000 text passages (500 passages each for human and AI-generated text classes) of $\sim$200 tokens in length. XSum contains long news articles in its "document" feature. To evaluate the robustness of our attacks to distribution shifts, we include more datasets. We use PubMedQA, which is a medical text dataset. Kafkai dataset (Pu et al., 2023) contains real and fake articles (generated using privately fine-tuned OpenAI models) from 10 different domains, such as cybersecurity, SEO, and marketing. It is generated using Kafkai text generation service (Kafkai, 2020).

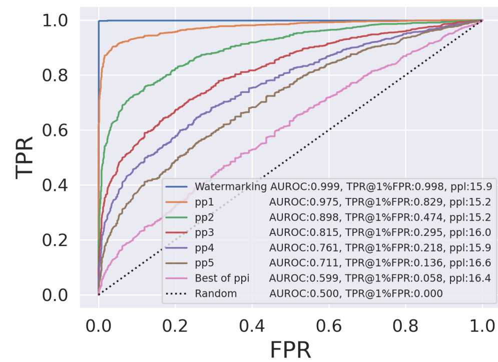
A.1 WATERMARK-BASED DETECTORS

In this section, we analyze the soft watermarking scheme in Kirchenbauer et al. (2023a). We use the powerful DIPPER paraphraser from Krishna et al. (2023) with 11B parameters for our recursive paraphrasing attack on the watermarking detector. On average, five rounds of our recursive paraphrase attack take around 36 seconds per text passage, 300 tokens in length. OPT-13B is used to measure the perplexity scores for all the settings. As shown in Table 1 and Appendix B.1, we perform a human study over the XSum dataset to evaluate the semantic drifts in our recursive paraphrasing framework. The MTurk human evaluation reveals that 70% of our recursive paraphrases maintain high-quality content preservation, and 89% of our recursive paraphrases have high-quality text or grammar.

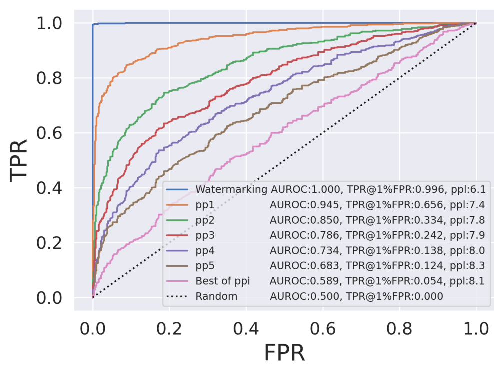
Figure 10 shows the performance of the soft watermarking detector in multiple settings. In all the settings, the detection performance drops as rounds of recursive paraphrasing proceed with a slight degradation in perplexity scores. After two rounds of paraphrasing (`pp2`), the detection performance (TPR@1%FPR) in all the settings drops below 50%. `Best of ppi`, which selects the paraphrase with the worst detection score, significantly degrades the detection performance to below 10% in all the settings with only slight degradation of 1.5, 0.5, 2.0, and 2.7 in perplexity measures.



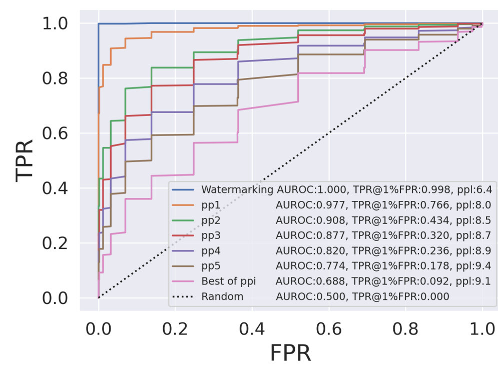
(a) XSum with OPT-1.3B



(b) XSum with GPT-2-Medium



(c) PubMedQA with OPT-1.3B



(d) Kafkai with OPT-1.3B

Figure 10: ROC plots for soft watermarking (Kirchenbauer et al., 2023a) with our recursive paraphrasing attacks. AUROC, TPR@1%FPR, and perplexity scores measured using OPT-13B are given in the legend. Detection performance on the XSum dataset using two different LLMs — OPT-1.3B and GPT-2-Medium — are evaluated in (a) and (b), respectively. (c) and (d), respectively, show the performance of the detector on two datasets — PubMedQA and Kafkai — with distribution shifts using OPT-1.3B. In all the settings, we observe that the detection performance of the watermarking-based detector reduces drastically with only a slight degradation in perplexity measures.

A.2 Zero-shot and Trained Detectors

In this section, we analyze the zero-shot and trained detectors in prior literature (Mitchell et al., 2023; Solaiman et al., 2019; Gehrmann et al., 2019; Ippolito et al., 2019; OpenAI, 2019). We use the T5-based paraphraser (Damodaran, 2021), Parrot, to paraphrase the AI-generated text and use OPT-13B to measure the perplexity scores for all the settings. We perform our experiments on the XSum (Narayan et al., 2018), PubMedQA (Jin et al., 2019), and Kafkai (Kafkai, 2020) datasets with GPT-2-Medium and OPT-1.3B as the target generative models. In Figure 11 (ROC curves) and Tables 2 (TPR@1%FPR values) and 3 (AUROC scores), we present our results. 1d, 1z, 10d, and 10z in Tables 2 and 3 refer to different variants of the DetectGPT (Mitchell et al., 2023).

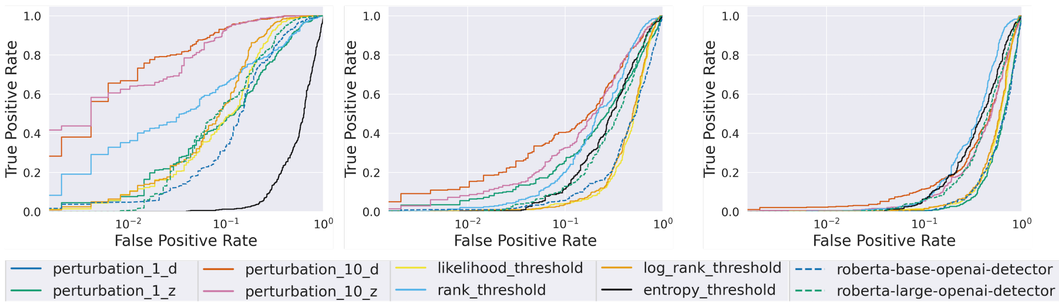
Figure 11 shows the performance of various zero-shot and trained detectors in multiple settings. The performance of these detectors drops significantly when the AI-generated text is paraphrased, and when given 5 queries to the detector, an adversary can fool most detectors effectively. Some detectors like OpenAI's RoBERTa-based models are more resilient on datasets like XSum, but are not reliable on other datasets like Kafkai. The perplexity scores of the GPT-2 generated text before any paraphrasing were 15.58 for XSum, 12.80 for PubMedQA, 19.11 for Kafkai, while the perplexity of OPT-1.3B generated text was 9.31. After paraphrasing, the perplexity scores are 20.06, 16.45, 20.01, and 13.96, respectively.

	DetectGPT				**Threshold** by				**RoBERTa**	
	1 d	1 z	10 d	10 z	Likeli-hood	Rank	Log Rank	Entropy	Base	Large
OPT-1.3B on XSum										
No attack	0.079	0.079	0.083	0.125	0.237	0.382	0.288	0.017	0.694	0.956
pp attack	0.014	0.014	0.018	0.006	0.006	0.006	0.006	0.326	0.025	0.479
5 pp attack	0.0	0.0	0.005	0.0	0.004	0.001	0.002	0.202	0.003	0.244
GPT-2 on PubMedQA										
No attack	0.05	0.05	0.598	0.481	0.085	0.379	0.144	0.029	0.748	0.902
pp attack	0.052	0.052	0.19	0.054	0.015	0.042	0.017	0.202	0.181	0.606
5 pp attack	0.0	0.0	0.031	0.002	0.008	0.01	0.012	0.135	0.088	0.452
GPT-2 on Kafkai										
No attack	0.077	0.077	0.669	0.625	0.088	0.352	0.085	0.0	0.048	0.006
pp attack	0.056	0.056	0.125	0.081	0.004	0.021	0.004	0.002	0.01	0.0
5 pp attack	0.0	0.0	0.023	0.006	0.002	0.004	0.002	0.0	0.0	0.0
GPT-2 on XSum										
No attack	0.169	0.169	0.599	0.326	0.114	0.444	0.186	0.026	0.881	1.0
pp attack	0.038	0.038	0.084	0.015	0.003	0.005	0.003	0.411	0.105	0.925
10 pp attack	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.058	0.02	0.792

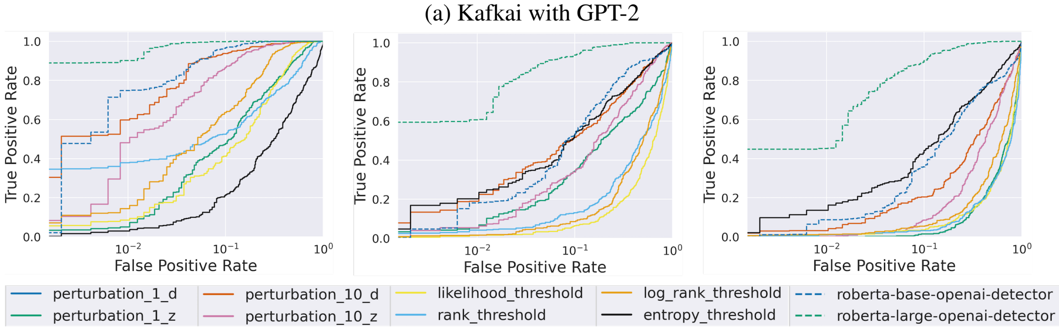
Table 2: TPR@1%FPR for trained and zero-shot detectors in different settings. For all attacks, we use the T5-based paraphraser. Here, “pp attack” refers to the paraphrasing attack where the AI output is paraphrased by the T5-based model. “i pp attack” refers to the setting where the attacker has black-box access to the detector. Here, the paraphraser generates “i” paraphrases for every passage, and the attacker selects the passage that has the worst detection score after “i” queries to the detector.

	DetectGPT				**Threshold** by				**RoBERTa**	
	1 d	1 z	10 d	10 z	Likeli--hood	Rank	Log Rank	Entropy	Base	Large
OPT-1.3B on XSum										
No attack	0.769	0.769	0.9	0.859	0.918	0.844	0.943	0.482	0.974	0.998
pp attack	0.487	0.487	0.453	0.41	0.241	0.387	0.282	0.868	0.562	0.945
5 pp attack	0.162	0.162	0.244	0.182	0.153	0.216	0.181	0.821	0.316	0.9
GPT-2 on PubMedQA										
No attack	0.816	0.816	0.973	0.955	0.804	0.796	0.892	0.615	0.982	0.998
pp attack	0.671	0.671	0.796	0.743	0.4	0.497	0.494	0.798	0.823	0.98
5 pp attack	0.33	0.33	0.601	0.541	0.327	0.314	0.409	0.752	0.712	0.967
GPT-2 on Kafkai										
No attack	0.814	0.814	0.976	0.971	0.865	0.86	0.89	0.394	0.817	0.86
pp attack	0.661	0.661	0.757	0.742	0.497	0.719	0.515	0.651	0.486	0.629
5 pp attack	0.353	0.353	0.532	0.513	0.412	0.627	0.426	0.576	0.358	0.53
GPT-2 on XSum										
No attack	0.837	0.837	0.976	0.949	0.879	0.868	0.93	0.617	0.993	1.0
pp attack	0.566	0.566	0.587	0.524	0.171	0.277	0.23	0.916	0.726	0.995
10 pp attack	0.115	0.115	0.202	0.177	0.075	0.104	0.108	0.744	0.464	0.983

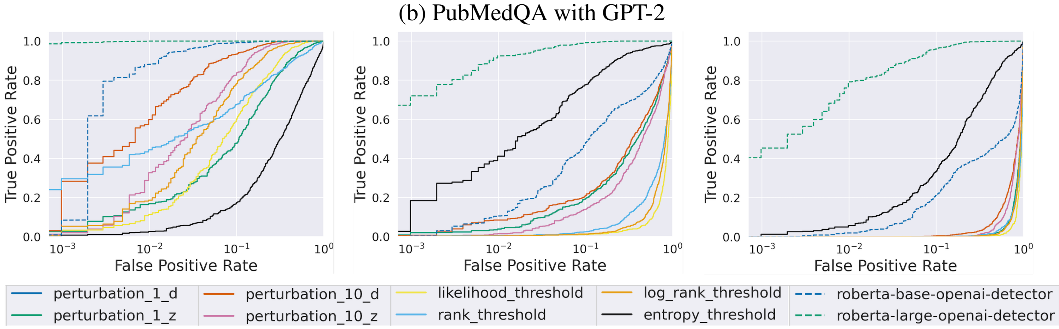
Table 3: AUROC for trained and zero-shot detectors in different settings. Here, “pp attack” refers to the paraphrasing attack where the AI output is paraphrased by the T5-based model. “i pp attack” refers to the setting where the attacker has black-box access to the detector. Here, the paraphraser generates “i” paraphrases for every passage, and the attacker selects the passage that has the worst detection score after “i” queries to the detector.



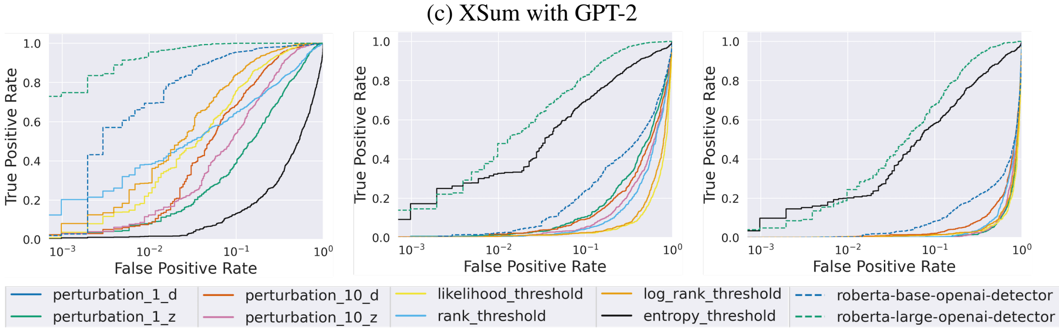
(a) Kafkai with GPT-2



(b) PubMedQA with GPT-2



(c) XSum with GPT-2

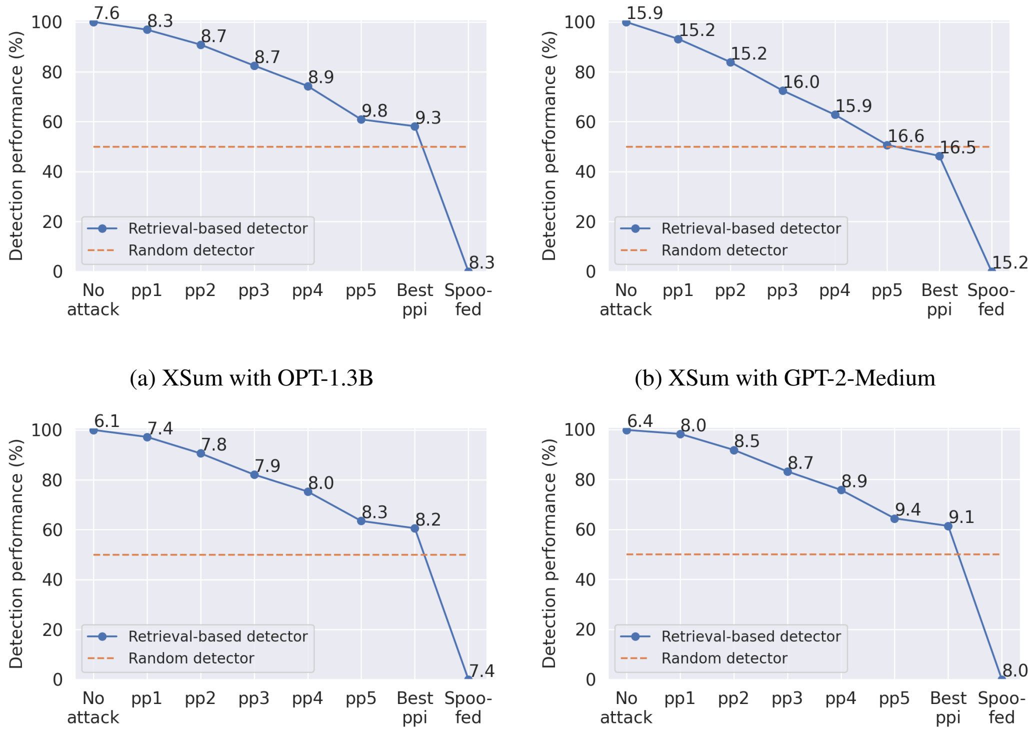


(d) XSum with OPT-1.3B

Figure 11: ROC curves for performance of various zero-shot and trained detectors for different models and datasets (**Left**) before attack, (**Middle**) after paraphrasing attack, (**Right**) applying paraphrasing attack with multiple queries to detector.

A.3 Retrieval-based Detectors

In this section, we analyze the retrieval-based detector proposed in Krishna et al. (2023). We show that our recursive paraphrasing attack is effective in breaking their detector. We use the 11B parameter DIPPER paraphraser (Krishna et al., 2023) for our attack. OPT-13B is used to measure the perplexity scores in all the settings.



(a) XSum with OPT-1.3B

(b) XSum with GPT-2-Medium

(c) PubMedQA with OPT-1.3B

(d) Kafkai with OPT-1.3B

Figure 12: ROC plots for soft watermarking (Kirchenbauer et al., 2023a) with our recursive paraphrasing attacks. AUROC, TPR@1%FPR, and perplexity scores measured using OPT-13B are given in the legend. Detection performance on the XSum dataset using two different LLMs — OPT-1.3B and GPT-2-Medium — are evaluated in (a) and (b), respectively. (c) and (d), respectively, show the performance of the detector on two datasets — PubMedQA and Kafkai — with distribution shifts using OPT-1.3B. In all the settings, we observe that the detection performance of the watermarking-based detector reduces drastically with only a slight degradation in perplexity measures.

Figure 12 shows the performance of the retrieval-based detector in multiple settings. In all the settings, the detection accuracy drops as rounds of recursive paraphrasing proceed with only a slight degradation in perplexity scores. We observe that the detector works well after a single round of paraphrasing (`pp1`). However, after five rounds of paraphrasing, `Best of ppi` reduces the detector's accuracy to close to 50% with only a slight degradation in perplexity scores. We also find that we can easily spoof the retrieval-based detector as discussed in §3 to deteriorate the detector's performance to 0%. Note that retrieval-based detectors are concerning since they might lead to serious privacy issues from storing users' LLM conversations.

B More Details on AI Paraphrasers

B.1 Human Evaluation Study on DIPPER Paraphrases

Apart from measuring the perplexity scores of the paraphrases using OPT-13B, we perform two human evaluation studies to investigate the quality of the paraphrases we use for the paraphrasing attack. We pick 20 random watermarked passages and their corresponding five rounds of recursive paraphrases (`pp1` to `pp5`) for human evaluation. Each paraphrase is evaluated by 3 unique MTurk workers. One of the twenty watermarked passages generated by the target LLM was non-English, and hence eliminated from the human evaluation. Therefore, our study includes a total of 95 recursive paraphrases. We use the same setup as Krishna et al. (2023) for our human study. As shown in Figure 13, users are given a source text with some highlighted portion. The non-highlighted portion of the source text is input into the target OPT-1.3B model that generates watermarked text which is highlighted for the user's reference. DIPPER paraphrases of the highlighted text are provided as the paraphrasing. The user is supposed to evaluate the quality of the paraphrases with respect to the highlighted watermarked text. They are supposed to rate it on a Likert scale of 1 to 5. See Table 4 for the evaluation summary on content preservation of DIPPER paraphrasers based on the user study. Table 5 shows the summary of the evaluation of text quality/grammar of the paraphrases. For the content preservation study, we use the following Likert scale: 5 – preserves the meaning of the source but differs in words and/or structure. 4 – preserves most information in the source but differs in some minor factual details. 3 – reserves some information in the source but differs in certain significant ways. 2 – topically related to the source but most information in the source is not preserved. 1 – not topically related. For the text quality or grammar quality study, we use the following Likert scale: 5 – the paraphrase has excellent grammar/quality with respect to the highlighted source. 4 – the paraphrase is clear and correct with minor grammatical errors. 3 – the paraphrase has few grammatical errors, but remains clear and comparable to highlighted source text. 2 – the paraphrase has significant number of grammatical errors, but remains understandable. 1 – the paraphrase is inferior to the highlighted source text with a lot of grammatical errors, may be difficult to comprehend.

Based on the evaluations, 70% of the paraphrases are rated high quality in terms of content preservation, and 89% of the paraphrases are rated to have high-quality text/grammar. Hence, our human study indicates that watermarking detectors can be evaded using recursive paraphrases with only a slight degradation in text quality.

ppi	**Average rating**	**Sum of 5 & 4 (%)**	**5 - Approx. equivalent (%)**	**4 - nearly equivalent (%)**	**3 - Somewhat equivalent (%)**	**2 - Topically related (%)**	**1 - Topically unrelated (%)**
i=1	4.0 ± 0.8	70.2	29.8	40.4	29.8	0.0	0.0
i=2	4.1 ± 0.8	77.2	33.3	43.9	19.3	3.5	0.0
i=3	3.9 ± 0.9	63.2	33.3	29.8	33.3	3.5	0.0
i=4	4.2 ± 0.9	80.0	49.1	30.9	14.5	5.5	0.0
i=5	3.7 ± 1.1	61.4	29.8	31.6	21.1	17.5	0.0
All ppi	$\mathbf{4.0 \pm 0.9}$	**70.4**	**35.1**	**35.3**	**23.6**	**6.0**	**0.0**

Table 4: Human evaluation of recursive paraphrases using MTurk for content preservation. `ppi` represents the $\mathtt{i}^{th}$ round of recursive paraphrasing.

ppi	**Average rating**	**Sum of 5 & 4 (%)**	**5 - Excellent (%)**	**4 - Good (%)**	**3 - Fair (%)**	**2 - Adequate (%)**	**1 - Poor (%)**
i=1	4.28 ± 0.67	87.72	40.35	47.37	12.28	0.00	0.00
i=2	4.12 ± 0.50	92.98	19.30	73.68	7.02	0.00	0.00
i=3	4.12 ± 0.53	91.23	21.05	70.18	8.77	0.00	0.00
i=4	4.11 ± 0.64	84.21	26.32	57.89	15.79	0.00	0.00
i=5	4.07 ± 0.53	89.47	17.54	71.93	10.53	0.00	0.00
All ppi	$\mathbf{4.14 \pm 0.58}$	**89.12**	**24.91**	**64.21**	**10.88**	**0.0**	**0.0**

Table 5: Human evaluation of recursive paraphrases using MTurk for text quality/grammar. `ppi` represents the $\mathtt{i}^{th}$ round of recursive paraphrasing.

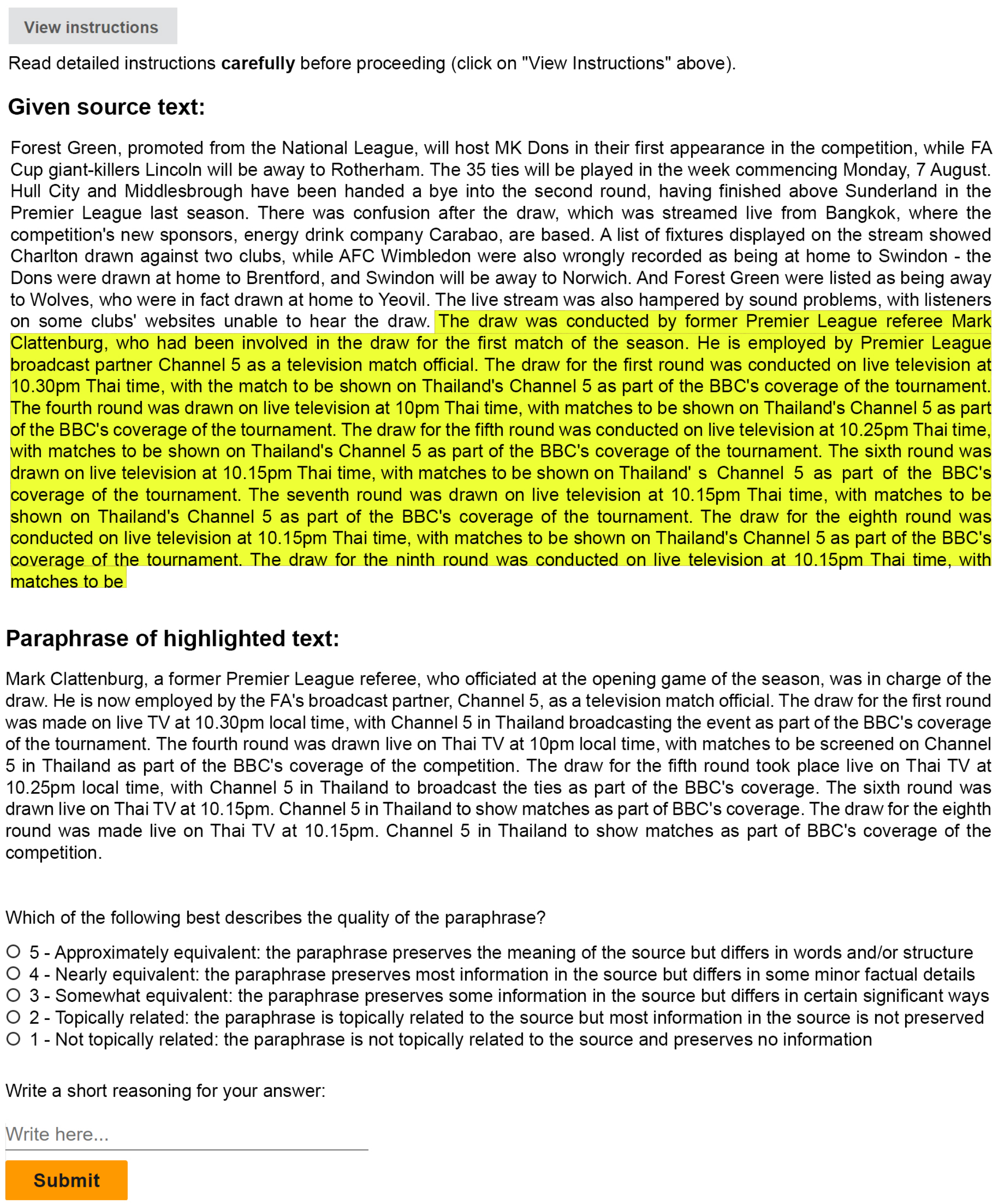


Figure 13: MTurk user interface for human evaluation of paraphrases.

B.2 Example Paraphrases

Description	Text
Input	The draw was conducted by former Premier League referee Mark Clattenburg, who had been involved in the draw for the first match of the season. He is employed by Premier League broadcast partner Channel 5 as a television match official. The draw for the first round was conducted on live television at 10.30pm Thai time, with the match to be shown on Thailand's Channel 5 as part of the BBC's coverage of the tournament. The fourth round was drawn on live television at 10pm Thai time, with matches to be shown on Thailand's Channel 5 as part of the BBC's coverage of the tournament. The draw for the fifth round was conducted on live television at 10.25pm Thai time, with matches to be shown on Thailand's Channel 5 as part of the BBC's coverage of the tournament. The sixth round was drawn on live television at 10.15pm Thai time, with matches to be shown on Thailand's Channel 5 as part of the BBC's coverage of the tournament. The seventh round was drawn on live television at 10.15pm Thai time, with matches to be shown on Thailand's Channel 5 as part of the BBC's coverage of the tournament. The draw for the eighth round was conducted on live television at 10.15pm Thai time, with matches to be shown on Thailand's Channel 5 as part of the BBC's coverage of the tournament. The draw for the ninth round was conducted on live television at 10.15pm Thai time, with matches to be
Paraphrase	Mark Clattenburg, a former Premier League referee, who officiated at the opening game of the season, was in charge of the draw. He is now employed by the FA's broadcast partner, Channel 5, as a television match official. The draw for the first round was made on live TV at 10.30pm local time, with Channel 5 in Thailand broadcasting the event as part of the BBC's coverage of the tournament. The fourth round was drawn live on Thai TV at 10pm local time, with matches to be screened on Channel 5 in Thailand as part of the BBC's coverage of the competition. The draw for the fifth round took place live on Thai TV at 10.25pm local time, with Channel 5 in Thailand to broadcast the ties as part of the BBC's coverage. The sixth round was drawn live on Thai TV at 10.15pm. Channel 5 in Thailand to show matches as part of BBC's coverage. The draw for the eighth round was made live on Thai TV at 10.15pm. Channel 5 in Thailand to show matches as part of BBC's coverage of the competition.
Input	"We had to cancel his operation and put off on him, he was supposed to have it on Christmas Eve." He said Josh and his family had been in touch with some people from the US and Australia who are in Belfast right now. He is one of the first people from the US to come to Belfast to meet and speak with Josh, and the rest of the family will visit from the US in the next few days. "Josh has become a worldwide man's man, and it has been incredible to see how everybody has taken to him. He's an ambassador for Irish people and for Ireland, and the people are really behind him. He's a great ambassador for Ireland and for his country and it has been really good." Josh was taken to Queen's University Hospital for surgery and they discovered and cut down his appendix, which had been causing him pain for more than a year. The surgery took place on January 11. He is still on painkillers, and the family are encouraging people to keep praying for his recovery. The hashtag #prayforjosh and the post and the message he messages are now being shared widely through social media by people from the US and Australia. The most recent tweets from his father's account are about his recovery in Belfast. He is also being accompanied by a "pray warriors" team and the hashtag #prayforjosh is now being used by other campaigners to urge people to send support to Josh. The church is also asking people to help
Paraphrase	"His operation was cancelled and put back - it was due to be on Christmas Eve." He said that the family and friends of Josh have been in touch with people from the United States and Australia who are currently in Belfast. He himself was one of the first to arrive in the city to meet and talk to the teenager, and more of the family will arrive in a few days. "Josh has become a 'global' man and it's been amazing to see how people have taken to him. He's a real Irishman and an ambassador for Ireland and it's been really good for him." He said that on January 11th, when he was taken to Queen's University Hospital in Belfast for the operation, a large primary cancer and five smaller growths were removed. He is still recovering and the family ask people to keep praying for his recovery. The post with the hashtag #prayforjosh and the messages it contains have now been shared all over social media by people in the US and Australia. The last few posts on his father's Twitter account are about his recovery in Belfast. He has also been surrounded by a team of prayer warriors and the campaign has now widened, with people using the tag #prayforjosh to urge others to show their support for him. The church is also calling on people to help in other ways.
Input	I've enjoyed my time with the team." Wales, who won 11 medals in total in Brazil, failed to make the podium at three consecutive Games from 1996 to 2008 before Doull stepped into the team and helped to end the long wait for a Welsh team pursuit gold. The Welsh team were in fine form on Saturday as they captured their 11th medal, equalling the record of 14 won in 1972. The Welsh team's run to gold this year was well-documented. Geraint Thomas, the 2008 and 2012 Olympic champion, set an Olympic record of 31 minutes 59.4 seconds to help Great Britain to gold. Born in Cardiff but living in Aberystwyth, Anthony Iwan Thomas was selected as one of the two Welsh track riders for Rio and earned a silver medal in the men's road race, which he won in August. "It's a dream come true to come back and win Olympic gold here," said Thomas. "To do it in this stadium, it's just surreal. I can't explain it. Let me get home and I'll be really grateful to everyone." He missed the 2012 Games through injury but returned to gold-winning form in 2016 when he won his third consecutive world championship in the men's road race, finishing second in Rio. "It's a real honour to be on the podium today," added the 35-year-old. "There's a lot of times when you think 'this is it' and you think
Paraphrase	I've loved every minute with the team. " Wales, who won 11 medals in Brazil, had not finished on the podium at three consecutive Olympic games from 1996 to 2008 before Doull helped end the country's long wait for a team pursuit gold medal. The nation's 11th medal on the last day equalled the record of 14 set in 1972. So many stories of success for the Welsh riders in Rio have already been written. Born in Cardiff but now based in the Ceredigion resort of Aberystwyth, the son of Geraint Thomas won silver in the men's road race. Thomas said: "It's just a dream come true to come back and win gold here and it's even more special to do it in this stadium. " If I get home I'll thank everybody. " The 38-year-old had to miss the 2012 Olympic games with injury but was back to winning ways in 2016 when he retained his world title and finished second in Rio. "There's so many times when you think, 'This is the day' and it never comes."

Table 6: Examples of paraphrased passage from the XSum dataset. The paraphrasing is performed using DIPPER (Krishna et al., 2023).

Description	Text
Input	The force has said it has no intention of changing policy or use of force, and will continue to use the weapons as part of its role policing the public. A spokesman for Scotland Yard said: "Officers in Scotland will continue to have a range of weapons on-hand, including tasers, pepper spray and a handgun and they will be used in accordance with Taser International guidelines. "We also have well established policies governing the use of these weapons and the use of force, and the use of force is judged to be proportionate when appropriate." The move was welcomed by a leading rights campaign group. John O'Donnell, the head of the High Court, said Tasers should have been compulsory in Scotland in 2015 following the death of a journalist in custody in Newcastle and the deaths of two more while in police custody. He said: "The use of these weapons by police in Scotland has been widely condemned and the High Court ruled they should be used to protect the public." A spokesman for the Department for Transport said: "No police force should decide for itself what constitutes necessary force and the use of these weapons by police in Scotland has been widely condemned. "The use of these weapons by officers must be subject to strict rules which safeguard the public and the officers' rights." The London police said they "do not comment on intelligence matters".?
Paraphrase	The Met said it had no plans to change its policy on the use of force and said the Tasers would be used "when it is appropriate and proportionate". A spokesman said: "Officers in Scotland will continue to carry a range of weapons including Tasers, pepper spray and handguns. "We have well-established policies on the use of these weapons and the use of force in general and we will continue to ensure these are followed." It was welcomed by the Scottish Human Rights Commission. The high court judge, Lord O'Donnell, said in February that the use of Tasers by police in Scotland should have been compulsory in the wake of the death of a journalist in police custody in Newcastle, and the subsequent deaths of two more people in police custody. He said: "The use of these weapons by police in Scotland has been widely condemned and the high court has ruled they must be used to protect the public." The Department for Transport said police should not be able to decide for themselves when it was appropriate to use force. "The use of these weapons by officers must be subject to strict rules which protect the public and officers' rights." The Met said it did not comment on intelligence matters. Would you like to book an appointment with the British Transport Police?
Input	"He gets on with me. I have no doubt about his motives." The move is also an enormous gamble but Koukash, a self-made millionaire who made his fortune as a real estate mogul in Dubai, is determined to make sure the gamble pays off. He is willing, he says, to let the man who guided Salford to seven Grand Final appearances head a great project. "He has an incredible track record of creating people and businesses," said Noble. "He has done it in Qatar, in Dubai, in America, here." He is also one of the game's most ruthless businessmen. Koukash, who moved to the UK from Sudan with his parents as a boy and has spent the past 30 years building his empire of clubs and businesses, has seen the game of rugby league decline dramatically over the past 10 years. He claims that the game has never been more popular than it is today. He is also a fervent supporter of the game and this makes his interest in rugby league even more compelling. Salford have been in dire straits and Koukash, who has ploughed much of his own money in the club, has promised to help them become one of the great clubs of the game. The club has all the right qualities and Koukash wants to make sure it happens. He has spent the past two weeks scouting for players to recruit and has already seen the arrival of two promising youngsters. The Reds need players to make them competitive
Paraphrase	He added: "I know him. I have no doubt about his intentions." It's a huge gamble for the self-made millionaire who has made his money in property in Dubai. He is happy to entrust the man who took Salford to seven Challenge Cup finals with his great plan. " He's got an incredible record of turning around people and businesses," Noble said. "He's done it in Qatar, in Dubai, in America, here." He's also one of the most ruthless businessmen in the game. Having arrived in Britain as a boy from Sudan with his parents, he's spent the past 30 years building an empire of clubs and businesses. He's also seen the game decline in popularity over the past decade, but insists it's now more popular than ever. He is a huge supporter of the game and that's why he's interested in the sport. Salford have been in crisis and he's promised to help them become one of the great clubs. The club has all the attributes and Koukash is determined to make it happen. He has spent the past fortnight looking at players and has already recruited two. He has to make the Reds a more competitive side and has already brought in a couple of players who have impressed.
Input	She said he would not stop attacking and asked for help. She said she wanted "peace" but "not death". Henderson-McCarroll, of St Nicholas Drive, Newry, admitted manslaughter while in charge of a dangerous drug. She said her actions were a "blip in my mind" as a result of a "bad decision" to take drugs. Justice Treacy said he would not impose a custodial sentence on Henderson-McCarroll, but instead sentenced her to three years' imprisonment. The judge said he would not impose the maximum sentence for manslaughter given the circumstances, but felt he would not impose the minimum of two and a half years. He told Henderson-McCarroll: "He (Mr Girvan) would not be in his right mind if he would not have let his guard down. If there was one thing the jury should have heard - it was that your actions were a blip in my mind. You didn't intend to kill him. You were acting in self-defence. You poked him and your actions were a blip and a bit of a lapse in judgment." The judge said the maximum sentence for manslaughter given Henderson's previous convictions would have amounted to between five and seven years. The judge said it was "not an uncommon crime" to kill someone in self-defence. He said sentencing Henderson was an "ugly case of drug-induced madness." He added: "He (Mr Girvan) must have suffered terribly."
Paraphrase	She said he had not stopped attacking her and she called for help. She said she wanted "peace" but not death. Henderson-McCarroll, of St Nicholas Drive in Newry, admitted manslaughter while under the influence of a sedative. She said her actions were the result of a "mistake" after taking drugs. Mr Justice Treacy said he would not grant a suspended sentence, but instead would sentence her to three years in prison. He said he would not impose the maximum sentence for manslaughter, in the circumstances, but did not feel he should impose the minimum term of two and a half years. He said to the defendant: "You could not in your right mind have left your guard down, you did not intend to kill him. If there was one thing the jury ought to have heard, it was that your actions were a momentary lapse of reason. You acted in self defence, you poked him with a knife, your actions were momentary and a lapse of reason." The judge said that given the defendant's previous record the maximum sentence for manslaughter, with a minimum term of a year, would have been five to seven years. He said it was not an uncommon crime for someone to kill in self defence. " But in this case, it was an ugly case of drug induced madness. " He added: "Mr Girvan must have suffered horribly."

Table 7: Examples of paraphrased passage from the XSum dataset. The paraphrasing is performed using DIPPER (Krishna et al., 2023).

Description	Text
Input	In the year 2014-2015, Prison Link Cymru made 1,099 referrals, and said some offenders waited up to a year before finding suitable accommodation. Staff from the charitable organization say that the investment in housing would be cheaper than having to re-lock homeless offenders. The Welsh government said that more people than ever are getting help to deal with their housing problems. Changes to the Welsh Housing Act in 2015 removed priority rights for ex-offenders. Prison Link Cymru, which helps people to find accommodation after they have been released, said that women generally had a good time, as they had to deal with children and other problems.
`pp1`	In the year 2014-2015, Prison Link Cymru made 1,099 referrals, and said some offenders waited up to a year before finding suitable accommodation. Staff from the charitable organization say that the investment in housing would be cheaper than having to re-lock homeless offenders. The Welsh government said that more people than ever are getting help to deal with their housing problems. Changes to the Welsh Housing Act in 2015 removed priority rights for ex-offenders. Prison Link Cymru, which helps people to find accommodation after they have been released, said that women generally had a good time, as they had to deal with children and other problems.
`pp2`	In 2015-16, Prison Link Cymru had 1,099 referrals and said some offenders were living rough for up to a year before finding suitable accommodation. Workers at the charity claim investment in housing would be cheaper than locking up homeless repeat offenders. The Welsh Government said more people than ever before were getting help to address housing problems. Prison Link Cymru, which helps people find accommodation after they have left prison, said things were generally good for women because issues such as children or domestic violence were often the main reasons they returned to prison.
`pp3`	In fact, it was one of the main reasons why they came back to prison. In the year 2015-16, Prison Link Cymru made 1,099 referrals and said that offenders had lived for up to a year before finding suitable accommodation. The workers at the charitable organization say that the investment in housing would be cheaper than re-locking homeless offenders. The government of Wales says that more people than ever are being helped to deal with their housing problems. Prison Link Cymru, which helps people to find accommodation after being released, says that women are generally in a good position, as children and other problems are often the main reason why they are returned to prison.
`pp4`	In the year to the end of March, Prison Link Cymru had 1,099 referrals and said offenders had been living rough for up to a year before finding suitable accommodation. Workers at the charity say investment in housing would be cheaper than re-imprisoning homeless repeat offenders. The Welsh Government says more people than ever before are getting help to address their housing problems. Prison Link Cymru, which helps people find accommodation after they have been released from prison, said generally things were good for women because issues such as children and other problems were often the main reasons they returned to prison.
`pp5`	The government of Wales says that more people than ever before are being helped to deal with their housing problems. In the year 2015-16, Prison Link Cymru referred 1,099 people and said that homeless people had lived up to a year before finding suitable accommodation. The workers at the charitable organization say that the investment in housing would be cheaper than imprisoning homeless offenders again. Prison Link Cymru, which helps people find accommodation after being released, says that women are generally well, because children and other problems are often the main reason why they return to prison.

Table 8: Example of a recursively paraphrased passage from the XSum dataset. The paraphrasing is performed using DIPPER (Krishna et al., 2023). `ppi` refers to the output after `i` rounds of recursive paraphrasing.

C Proofs and Corollaries

C.1 Proof of Theorem 1

Theorem 1. *The area under the ROC of any detector D is bounded as*

$$\mathsf{AUROC}(D) \leq \frac{1}{2} + \mathsf{TV}(\mathcal{M}, \mathcal{H}) - \frac{\mathsf{TV}(\mathcal{M}, \mathcal{H})^2}{2}.$$

Proof. The ROC is a plot between the true positive rate (TPR) and the false positive rate (FPR), which are defined as follows:

$$\begin{aligned} \mathsf{TPR}_\gamma &= \mathbb{P}_{s\sim\mathcal{M}}[D(s) \geq \gamma] \\ \text{and } \mathsf{FPR}_\gamma &= \mathbb{P}_{s\sim\mathcal{H}}[D(s) \geq \gamma], \end{aligned}$$

where γ is some classifier parameter. We can bound the difference between the TPR_γ and the FPR_γ by the total variation between M and H:

$$|\mathsf{TPR}_\gamma - \mathsf{FPR}_\gamma| = |\mathbb{P}_{s\sim\mathcal{M}}[D(s) \geq \gamma] - \mathbb{P}_{s\sim\mathcal{H}}[D(s) \geq \gamma]| \leq \mathsf{TV}(\mathcal{M}, \mathcal{H}) \tag{1}$$

$$\mathsf{TPR}_\gamma \leq \mathsf{FPR}_\gamma + \mathsf{TV}(\mathcal{M}, \mathcal{H}). \tag{2}$$

Since the TPR_γ is also bounded by 1 we have:

$$\mathsf{TPR}_\gamma \leq \min(\mathsf{FPR}_\gamma + \mathsf{TV}(\mathcal{M}, \mathcal{H}), 1). \tag{3}$$

Denoting FPR_γ, TPR_γ, and $\mathsf{TV}(\mathcal{M}, \mathcal{H})$ with x, y, and tv for brevity, we bound the AUROC as follows:

$$\begin{aligned} \mathsf{AUROC}(D) = \int_0^1 y\, dx &\leq \int_0^1 \min(x + tv, 1) dx \\ &= \int_0^{1-tv} (x + tv) dx + \int_{1-tv}^1 dx \\ &= \left| \frac{x^2}{2} + tvx \right|_0^{1-tv} + |x|_{1-tv}^1 \\ &= \frac{(1-tv)^2}{2} + tv(1-tv) + tv \\ &= \frac{1}{2} + \frac{tv^2}{2} - tv + tv - tv^2 + tv \\ &= \frac{1}{2} + tv - \frac{tv^2}{2}. \end{aligned}$$

□

C.2 General Trade-offs For Detection

Paraphrasing to Evade Detection. Although our analysis considers general distributions, it can also be applied to specific scenarios, such as particular writing styles or sentence paraphrasing, by defining $\mathcal{M}$ and $\mathcal{H}$ appropriately. For example, $\mathcal{M}$ can be the outputs from an LLM trained to mimic a particular set of people, or $\mathcal{H}$ can be the text distribution of a specific person. Similarly, Corollary 1 shows that if a paraphraser's goal is to lower the TV between paraphrased AI text and human text, then detection gets harder.

Set $\mathcal{M} = \mathcal{R}_\mathcal{M}(s)$ and $\mathcal{H} = \mathcal{R}_\mathcal{H}(s)$ to be the distribution of sequences with similar meanings to s produced by the paraphraser and humans, respectively.

Corollary 1. *The area under the ROC of the detector D is bounded as*

$$\mathsf{AUROC}(D) \leq \frac{1}{2} + \mathsf{TV}(\mathcal{R}_\mathcal{M}(s), \mathcal{R}_\mathcal{H}(s)) - \frac{\mathsf{TV}(\mathcal{R}_\mathcal{M}(s), \mathcal{R}_\mathcal{H}(s))^2}{2}.$$

Another way to understand the limitations of AI-generated text detectors is directly through the characterization of the trade-offs between true positive rates and false positive rates. Adapting inequality 2, we have the following corollaries:

Corollary 2. *For any watermarking scheme W,*

$$\Pr_{s_w \sim \mathcal{R}_{\mathcal{M}}(s)}[s_w \text{ is watermarked using } W] \leq \Pr_{s_w \sim \mathcal{R}_{\mathcal{H}}(s)}[s_w \text{ is watermarked using } W] + \mathsf{TV}(\mathcal{R}_{\mathcal{M}}(s), \mathcal{R}_{\mathcal{H}}(s)),$$

where $\mathcal{R}_{\mathcal{M}}(s)$ and $\mathcal{R}_{\mathcal{H}}(s)$ are the distributions of rephrased sequences for s produced by the paraphrasing model and humans, respectively.

Humans may have different writing styles. Corollary 2 indicates that if a rephrasing model resembles certain human text distribution $\mathcal{H}$ (i.e. $\mathsf{TV}(\mathcal{R}_{\mathcal{M}}(s), \mathcal{R}_{\mathcal{H}}(s))$ is small), then either certain people's writing will be detected falsely as watermarked (i.e. $\Pr_{s_w \sim \mathcal{R}_{\mathcal{H}}(s)}[s_w$ is watermarked using $W]$ is high) or the paraphrasing model can remove the watermark (i.e. $\Pr_{s_w \sim \mathcal{R}_{\mathcal{M}}(s)}[s_w$ is watermarked using $W]$ is low).

Corollary 3. *For any AI-text detector D,*

$$\Pr_{s \sim \mathcal{M}}[s \text{ is detected as AI-text by } D] \leq \Pr_{s \sim \mathcal{H}}[s \text{ is detected as AI-text by } D] + \mathsf{TV}(\mathcal{M}, \mathcal{H}),$$

where $\mathcal{M}$ and $\mathcal{H}$ denote text distributions by the model and by humans, respectively.

Corollary 3 indicates that if a model resembles certain human text distribution $\mathcal{H}$ (i.e. $\mathsf{TV}(\mathcal{M}, \mathcal{H})$ is small), then either certain people's writing will be detected falsely as AI-generated (i.e. $\Pr_{s \sim \mathcal{H}}[s$ is detected as AI-text by $D]$ is high) or the AI-generated text will not be detected reliably (i.e. $\Pr_{s \sim \mathcal{M}}[s$ is detected as AI-text by $D]$ is low).

A recent work (Chakraborty et al., 2023) shows a trade-off on the detection problem with respect to the availability of the number of data samples for detection. They show a TV upper bound for the detector's AUROC using an information theoretic approach. However, the underlying assumption of their result is that several *independent* samples are available to the detector from either human or text distribution, which might not be a practical assumption since sentences in a document are often correlated with each other. Also, a large number of data samples need not be available for pragmatic scenarios. For example, it may not be practical for a text detector to ask a student to write multiple essays for an assignment or to assume that a Twitter bot would publish longer tweets that are completely written by the AI without any human intervention.

C.3 Tightness Analysis for Theorem 1

In this section, we show that the bound in Theorem 1 is tight. For a given distribution of human-generated text sequences $\mathcal{H}$, we construct an AI-text distribution $\mathcal{M}$ and a detector D such that the bound holds with equality. Define sublevel sets of the probability density function of the distribution of human-generated text $\mathsf{pdf}_{\mathcal{H}}$ over the set of all sequences Ω as follows:

$$\Omega_{\mathcal{H}}(c) = \{s \in \Omega \mid \mathsf{pdf}_{\mathcal{H}}(s) \leq c\}$$

where $c \in \mathbb{R}$. Assume that, $\Omega_{\mathcal{H}}(0)$ is not empty. Now, consider a distribution $\mathcal{M}$, with density function $\mathsf{pdf}_{\mathcal{M}}$, which has the following properties:

1. The probability of a sequence drawn from $\mathcal{M}$ falling in $\Omega_{\mathcal{H}}(0)$ is $\mathsf{TV}(\mathcal{M}, \mathcal{H})$, i.e., $\mathbb{P}_{s \sim \mathcal{M}}[s \in \Omega_{\mathcal{H}}(0)] = \mathsf{TV}(\mathcal{M}, \mathcal{H})$.
2. $\mathsf{pdf}_{\mathcal{M}}(s) = \mathsf{pdf}_{\mathcal{H}}(s)$ for all $s \in \Omega(\tau) - \Omega(0)$ where $\tau > 0$ such that $\mathbb{P}_{s \sim \mathcal{H}}[s \in \Omega(\tau)] = 1 - \mathsf{TV}(\mathcal{M}, \mathcal{H})$.
3. $\mathsf{pdf}_{\mathcal{M}}(s) = 0$ for all $s \in \Omega - \Omega(\tau)$.

Define a hypothetical detector D that maps each sequence in Ω to the negative of the probability density function of $\mathcal{H}$, i.e., $D(s) = -\mathsf{pdf}_{\mathcal{H}}(s)$. Using the definitions of TPR_γ and FPR_γ, we have:

$$\begin{aligned}
\mathsf{TPR}_\gamma &= \mathbb{P}_{s \sim \mathcal{M}}[D(s) \geq \gamma] \\
&= \mathbb{P}_{s \sim \mathcal{M}}[-\mathsf{pdf}_{\mathcal{H}}(s) \geq \gamma] \\
&= \mathbb{P}_{s \sim \mathcal{M}}[\mathsf{pdf}_{\mathcal{H}}(s) \leq -\gamma] \\
&= \mathbb{P}_{s \sim \mathcal{M}}[s \in \Omega_{\mathcal{H}}(-\gamma)]
\end{aligned}$$

Similarly,

$$\mathsf{FPR}_\gamma = \mathbb{P}_{s\sim\mathcal{H}}[s \in \Omega_{\mathcal{H}}(-\gamma)].$$

For $\gamma \in [-\tau, 0]$,

$$\begin{aligned}
\mathsf{TPR}_\gamma &= \mathbb{P}_{s\sim\mathcal{M}}[s \in \Omega_{\mathcal{H}}(-\gamma)] \\
&= \mathbb{P}_{s\sim\mathcal{M}}[s \in \Omega_{\mathcal{H}}(0)] + \mathbb{P}_{s\sim\mathcal{M}}[s \in \Omega_{\mathcal{H}}(-\gamma) - \Omega_{\mathcal{H}}(0)] \\
&= \mathsf{TV}(\mathcal{M},\mathcal{H}) + \mathbb{P}_{s\sim\mathcal{M}}[s \in \Omega_{\mathcal{H}}(-\gamma) - \Omega_{\mathcal{H}}(0)] && \text{(using property 1)} \\
&= \mathsf{TV}(\mathcal{M},\mathcal{H}) + \mathbb{P}_{s\sim\mathcal{H}}[s \in \Omega_{\mathcal{H}}(-\gamma) - \Omega_{\mathcal{H}}(0)] && \text{(using property 2)} \\
&= \mathsf{TV}(\mathcal{M},\mathcal{H}) + \mathbb{P}_{s\sim\mathcal{H}}[s \in \Omega_{\mathcal{H}}(-\gamma)] - \mathbb{P}_{s\sim\mathcal{H}}[s \in \Omega_{\mathcal{H}}(0)] && (\Omega_{\mathcal{H}}(0) \subseteq \Omega_{\mathcal{H}}(-\gamma)) \\
&= \mathsf{TV}(\mathcal{M},\mathcal{H}) + \mathsf{FPR}_\gamma. && (\mathbb{P}_{s\sim\mathcal{H}}[s \in \Omega_{\mathcal{H}}(0)] = 0)
\end{aligned}$$

For $\gamma \in [-\infty, -\tau]$, $\mathsf{TPR}_\gamma = 1$, by property 3. Also, as γ goes from 0 to $-\infty$, FPR_γ goes from 0 to 1. Therefore, $\mathsf{TPR}_\gamma = \min(\mathsf{FPR}_\gamma + \mathsf{TV}(\mathcal{M},\mathcal{H}), 1)$ which is similar to Equation 3. Calculating the AUROC in a similar fashion as in the previous section, we get the following:

$$\mathsf{AUROC}(D) = \frac{1}{2} + \mathsf{TV}(\mathcal{M},\mathcal{H}) - \frac{\mathsf{TV}(\mathcal{M},\mathcal{H})^2}{2}.$$

C.4 Pseudorandomness in LLMs

Most machine learning models, including LLMs, use pseudorandom number generators in one form or another to produce their outputs. For example, an LLM may use a pseudorandom number generator to sample the next token in the output sequence. In discussing our hardness result, Kirchenbauer et al. (2023b) in a more recent work argue that this pseudorandomness makes the AI-generated text distribution very different from the human-generated text distribution. This is because the pseudorandom AI-generated distribution is a collection of Dirac delta function distributions, and a human is exorbitantly unlikely to produce a sample corresponding to any of the delta functions. In our framework, this means that the TV between the human and pseudorandom AI-generated distributions is almost one, making the bound in Theorem 1 vacuous.

We argue that although the true TV between the human and pseudorandom AI-generated distributions is high and there exists (in theory) a detector function that can separate the distributions almost perfectly, this function may not be efficiently computable. Any polynomial-time computable detector can only achieve a negligible advantage from the use of pseudorandomness instead of true randomness. If we had knowledge of the seed used for the pseudorandom number generator, we would be able to predict the pseudorandom samples. However, an individual seeking to evade detection could simply randomize this seed making it computationally infeasible to predict the samples.

We modify the bound in Theorem 1 to include a negligible correction term ϵ to account for the use of pseudorandomness. We prove that the performance of a polynomial-time computable detector D on a pseudorandom version of the AI-generated distribution $\widehat{\mathcal{M}}$ is bounded by the total variation for the truly random distribution $\mathcal{M}$ (resulting from the LLM using true randomness) as follows:

$$\mathsf{AUROC}(D) \leq \frac{1}{2} + \mathsf{TV}(\mathcal{M},\mathcal{H}) - \frac{\mathsf{TV}(\mathcal{M},\mathcal{H})^2}{2} + \epsilon.$$

The term ϵ represents the gap between the probabilities assigned by $\mathcal{M}$ and $\widehat{\mathcal{M}}$ to any polynomial-time computable $\{0, 1\}$-function f, i.e.,

$$\left|\mathbb{P}_{s\in\mathcal{M}}[f(s) = 1] - \mathbb{P}_{s\in\widehat{\mathcal{M}}}[f(s) = 1]\right| \leq \epsilon. \tag{4}$$

This term is orders of magnitude smaller than any of the terms in the bound and can be safely ignored. For example, commonly used pseudorandom generators[1] can achieve an ϵ that is bounded by a negligible function $1/b^t$ of the number of bits b used in the seed of the generator for a positive integer t[2] (Blum et al., 1982; Blum & Micali, 1984). From a computational point of view, the TV for the pseudorandom distribution is almost the same as the truly random AI-generated distribution. Thus,

[1] Cryptographic PRNGs: https://en.wikipedia.org/wiki/Pseudorandom_number_generator

[2] Negligible function: https://en.wikipedia.org/wiki/Negligible_function

our framework provides a reasonable approximation for real-world LLMs, and the hardness result holds even in the presence of pseudorandomness.

Computational Total Variation Distance. Just as the total variation distance TV between two probability distributions is defined as the difference in probabilities assigned by the two distributions to any $\{0, 1\}$-function, we define a computational version of this distance TV_c for polynomial-time computable functions:

$$\mathsf{TV}_c(A, B) = \max_{f \in \mathcal{P}} \left|\mathbb{P}_{s\sim A}[f(s) = 1] - \mathbb{P}_{s\sim B}[f(s) = 1]\right|,$$

where $\mathcal{P}$ represents the set of polynomial-time computable $\{0, 1\}$-functions. $\mathcal{P}$ could also be defined as the set of all polynomial-size circuits which could be more appropriate for deep neural network-based detectors. The function f could be thought of as the indicator function for the detection parameter being above a certain threshold, i.e., $D(s) \geq \gamma$ as in the proof of Theorem 1. The following lemma holds for the performance of a polynomial-time detector D:

Lemma 1. *The area under the ROC of any polynomial-time computable detector D is bounded as*

$$\mathsf{AUROC}(D) \leq \frac{1}{2} + \mathsf{TV}_c(\widehat{\mathcal{M}}, \mathcal{H}) - \frac{\mathsf{TV}_c(\widehat{\mathcal{M}}, \mathcal{H})^2}{2}.$$

This lemma can be proved in the same way as Theorem 1 by replacing the truly random AI-generated distribution $\mathcal{M}$ with its pseudorandom version $\widehat{\mathcal{M}}$ and the true total variation TV with its computaional variant TV_c.

Next, we relate the computational total variation TV_c between $\mathcal{H}$ and the pseudorandom distribution $\widehat{\mathcal{M}}$ with the total variation TV between $\mathcal{H}$ and the truly random distribution $\mathcal{M}$.

Lemma 2. *For human distribution $\mathcal{H}$, truly random AI-generated distribution $\mathcal{M}$ and its pseudorandom version $\widehat{\mathcal{M}}$,*

$$\mathsf{TV}_c(\widehat{\mathcal{M}}, \mathcal{H}) \leq \mathsf{TV}(\mathcal{M}, \mathcal{H}) + \epsilon.$$

Proof.

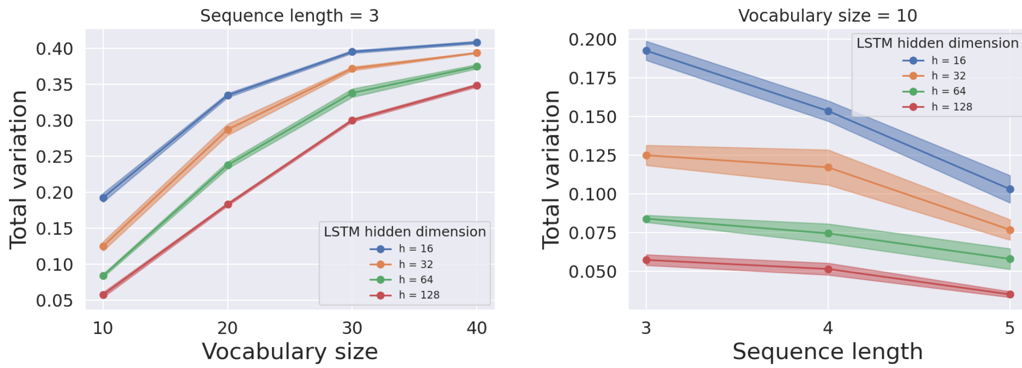
$$\begin{aligned}
\mathsf{TV}_c(\widehat{\mathcal{M}}, \mathcal{H}) &= \max_{f \in \mathcal{P}} \left|\mathbb{P}_{s\sim\mathcal{H}}[f(s) = 1] - \mathbb{P}_{s\sim\widehat{\mathcal{M}}}[f(s) = 1]\right| && \text{(from definition of } \mathsf{TV}_c\text{)} \\
&= \max_{f \in \mathcal{P}} \big|\mathbb{P}_{s\sim\mathcal{H}}[f(s) = 1] - \mathbb{P}_{s\sim\mathcal{M}}[f(s) = 1] \\
&\qquad + \mathbb{P}_{s\sim\mathcal{M}}[f(s) = 1] - \mathbb{P}_{s\sim\widehat{\mathcal{M}}}[f(s) = 1]\big| && \text{(+/-ing } \mathbb{P}_{s\sim\mathcal{M}}[f(s) = 1]\text{)} \\
&\leq \max_{f \in \mathcal{P}} \left|\mathbb{P}_{s\sim\mathcal{H}}[f(s) = 1] - \mathbb{P}_{s\sim\mathcal{M}}[f(s) = 1]\right| \\
&\qquad + \left|\mathbb{P}_{s\sim\mathcal{M}}[f(s) = 1] - \mathbb{P}_{s\sim\widehat{\mathcal{M}}}[f(s) = 1]\right| && \text{(using } |a + b| \leq |a| + |b|\text{)} \\
&\leq \mathsf{TV}(\mathcal{M}, \mathcal{H}) + \epsilon. && \text{(from definition of } \mathsf{TV} \text{ and bound 4)}
\end{aligned}$$

□

We now use this to prove the modified version of our computational hardness result.

Theorem 2 (Computational Hardness Result). *The AUROC of any polynomial-time computable detector D for $\mathcal{H}$ and the pseudorandom distribution $\widehat{\mathcal{M}}$ is bounded using the TV for the truly random distribution $\mathcal{M}$ as*

$$\mathsf{AUROC}(D) \leq \frac{1}{2} + \mathsf{TV}(\mathcal{M}, \mathcal{H}) - \frac{\mathsf{TV}(\mathcal{M}, \mathcal{H})^2}{2} + \epsilon.$$



(a) Varying vocabulary size

(b) Varying sequence length

Figure 14: TV distances between synthetic toy data distributions and LSTM model generation distributions. TV distances are computed for multiple settings, varying the vocabulary size and sequence length of the training dataset and varying the size of the LSTM network used for training.

Proof.

$$\begin{aligned}
\mathsf{AUROC}(D) &\leq \frac{1}{2} + \mathsf{TV}_c(\widehat{\mathcal{M}}, \mathcal{H}) - \frac{\mathsf{TV}_c(\widehat{\mathcal{M}}, \mathcal{H})^2}{2} && \text{(from Lemma 1)} \\
&\leq \frac{1}{2} + \mathsf{TV}(\mathcal{M}, \mathcal{H}) + \epsilon - \frac{(\mathsf{TV}(\mathcal{M}, \mathcal{H}) + \epsilon)^2}{2} \\
& && \text{(from Lemma 2 and since } \tfrac{1}{2} + x - \tfrac{x^2}{2} \text{ is increasing in } [0,1]) \\
&= \frac{1}{2} + \mathsf{TV}(\mathcal{M}, \mathcal{H}) + \epsilon - \frac{\mathsf{TV}(\mathcal{M}, \mathcal{H})^2 + \epsilon^2 + 2\epsilon\mathsf{TV}(\mathcal{M}, \mathcal{H})}{2} \\
&\leq \frac{1}{2} + \mathsf{TV}(\mathcal{M}, \mathcal{H}) - \frac{\mathsf{TV}(\mathcal{M}, \mathcal{H})^2}{2} + \epsilon. && \text{(dropping negative terms containing } \epsilon)
\end{aligned}$$

□

C.5 Estimating TV Distance

In §4, we show experiments supporting the assumption that more advanced LLMs lead to smaller TV distance between human and machine text distributions. We present two experimental settings — (i) Using synthetic text data and (ii) Using projection. In Figure 14, we show the TV distances computed with varying vocabulary sizes and sequence lengths. In all the experiments, we consistently find that the TV distances reduce as the network size increases.

D More Details on Spoofing

D.1 Soft Watermark Detector

In Kirchenbauer et al. (2023a), they watermark LLM outputs by asserting the model to output tokens with some specific pattern that can be easily detected with meager error rates. Soft watermarked texts are majorly composed of *green list* tokens. If an adversary can learn the green lists for the soft watermarking scheme, they can use this information to generate human-written texts that are detected to be watermarked. Our experiments show that the soft watermarking scheme can be spoofed efficiently. Though the soft watermarking detector can detect the presence of a watermark very accurately, it cannot be certain if this pattern is actually generated by a human or an LLM. An *adversarial human* can compose derogatory watermarked texts in this fashion that are detected to be watermarked, which might cause reputational damage to the developers of the watermarked LLM. Therefore, it is important to study *spoofing attacks* to avoid such scenarios.

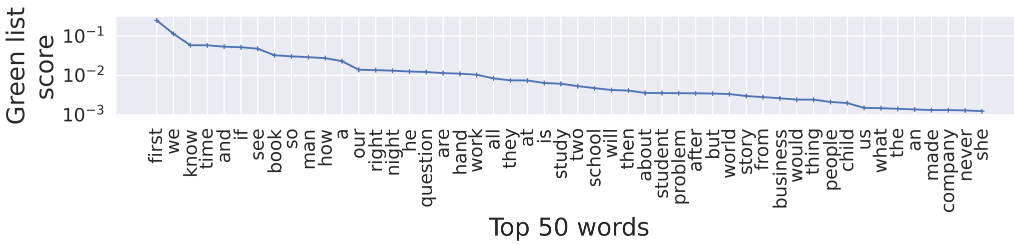


Figure 15: Inferred *green list score* for the token "the". The plot shows the top 50 words from our set of common words that are likely to be in the green list. The word "first" occurred $\sim 25\%$ of the time as suffix to "the".

In watermarking, the prefix word $s^{(t-1)}$ determines the green list for selecting the word $s^{(t)}$. The attacker's objective is to compute a proxy of green lists for the N most commonly used words in the vocabulary. We use a small value of $N = 181$ for our experiments. The attacker queries the watermarked OPT-1.3B Zhang et al. (2022) 10^6 times to observe pair-wise token occurrences in its output to estimate *green list score* for the N tokens. We find that inputting nonsense sentences composed of the N common words encourages the LLM to output text mostly composed of these words. This makes the querying more efficient. A token with a high green list score for a prefix $s^{(t)}$ might be in its green list (see Figure 15). We build a tool that helps adversarial humans create watermarked sentences by providing them with proxy green list. In this manner, we can spoof watermarking models easily. See Table 9 for example sentences created by an adversarial human. Figure 6 shows that the performance of watermark-based detectors degrades significantly in the presence of paraphrasing and spoofing attacks, showing that they are not reliable.

Human text	% tokens in green list	z-score	Detector output
the first thing you do will be the best thing you do. this is the reason why you do the first thing very well. if most of us did the first thing so well this world would be a lot better place. and it is a very well known fact. people from every place know this fact. time will prove this point to the all of us. as you get more money you will also get this fact like other people do. all of us should do the first thing very well. hence the first thing you do will be the best thing you do.	42.6	4.36	Watermarked

Table 9: Proof-of-concept human-generated texts flagged as watermarked by the soft watermarking scheme. A sentence composed by an *adversarial human* contains 42.6% tokens from the green list. The z-test threshold for watermark detection is 4, the same as the default hyperparameter in Kirchenbauer et al. (2023a).

D.2 Zero-Shot and Trained Detectors

We report the false positive rate fixed at a true positive rate of 90% and the true positive rate at a false positive rate of 1% in Table 10. The ROC curves before and after spoofing the detectors are provided in Figure 16. Our experiments demonstrate that most of these detection methods show a significant increase in false positive rates at a fixed true positive rate of 90% after spoofing. After this naïve spoofing attack, the true positive rate at a false positive rate of 1% and AUROC scores of these detectors drop significantly.

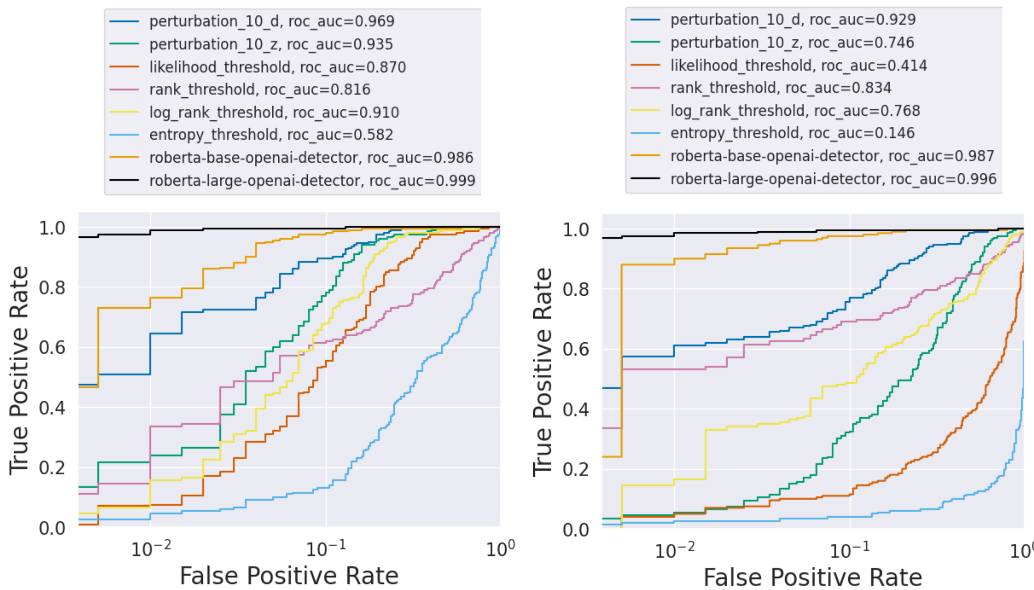


Figure 16: ROC curves before (left) and after (right) spoofing attack (§ 3). Most detectors exhibit quality degradation after our spoofing attack.

Detection Methods	T@F	F@T
Entropy threshold (Gehrmann et al., 2019)	**0.025** (0.045)	**0.995** (0.845)
Likelihood threshold (Solaiman et al., 2019)	**0.050** (0.075)	**0.995** (0.310)
Logrank threshold	0.165 (0.155)	**0.690** (0.190)
Rank threshold (Gehrmann et al., 2019)	0.530 (0.335)	**0.655** (0.590)
Roberta (base) OpenAI detector (OpenAI, 2019)	0.900 (0.765)	0.010 (0.035)
Roberta (large) OpenAI detector (OpenAI, 2019)	**0.985** (0.990)	0.000 (0.000)
DetectGPT (Mitchell et al., 2023)	**0.055** (0.240)	**0.555** (0.145)

Table 10: True positive rates at 1% false positive rate (T@F) and false positive rates at 90% true positive rate (F@T) after (before the attack in parentheses) the spoofing attack described in §3. Bolded numbers show successful attacks where T@F decreases, or F@T increases after spoofing.