Can AI-Generated Text be Reliably Detected?

Vinu Sankar Sadasivan
vinu@umd.edu

Aounon Kumar
aounon@umd.edu

Sriram Balasubramanian
sriramb@umd.edu

Wenxiao Wang
wwx@umd.edu

Soheil Feizi
sfeizi@umd.edu

Department of Computer Science
University of Maryland

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, both empirically and theoretically, we show that these detectors are not reliable in practical scenarios. Empirically, we show that *paraphrasing attacks*, where a light paraphraser is applied on top of the generative text model, can break a whole range of detectors, including the ones using the watermarking schemes as well as neural network-based detectors and zero-shot classifiers. We then provide a theoretical *impossibility result* indicating that for a sufficiently good language model, even the best-possible detector can only perform marginally better than a random classifier. Finally, we show that even LLMs protected by watermarking schemes can be vulnerable against spoofing attacks where *adversarial humans* can infer hidden watermarking signatures and add them to their generated text to be detected as text generated by the LLMs, potentially causing reputational damages to their developers. We believe these results can open an honest conversation in the community regarding the ethical and reliable use of AI-generated text.

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

Preprint. Under review.

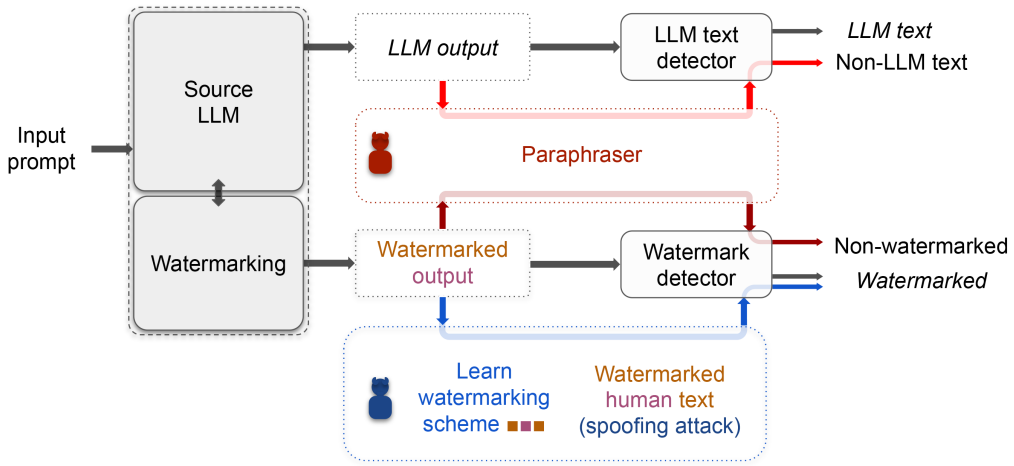


Figure 1: An illustration of vulnerabilities of existing AI-text detectors. We consider both watermarking-based and non-watermarking-based detectors and show that they are not reliable in practical scenarios. Colored arrow paths show the potential pipelines for adversaries to avoid detection. In red: an attacker can use a paraphraser to remove the LLM signatures from an AI-generated text to avoid detection. We show that this attack can break a wide range of detectors. We provide an *impossibility result* indicating that for a sufficiently good language model, even the best-possible detector can perform only marginally better than a random classifier. In blue: An adversary can query the soft watermarked LLM multiple times to learn its watermarking scheme. This information can be used to spoof the watermark detector by composing human text that is detected to be watermarked.

[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]. 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. 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., 2023, 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. [2023] partitions tokens into green and red lists to help create these patterns. A watermarked LLM samples a token, with high probability, from the green list determined by its prefix token. These watermarks are often imperceptible to humans.

In this paper, through both empirical and theoretical analysis, we show that state-of-the-art AI-text detectors are not reliable in practical scenarios. We first study empirical attacks on soft watermarking [Kirchenbauer et al., 2023], and a wide range of zero-shot [Mitchell et al., 2023] and neural network-based detectors [OpenAI, 2019]. We show that a *paraphrasing attack*, where a lightweight neural network-based paraphraser is applied to the output text of the AI-generative model, can evade various types of detectors. Before highlighting the results, let us provide an intuition why this attack is successful. For a given sentence s, suppose $P(s)$ is the set of all paraphrased sentences that have similar meanings to the sentence s. Moreover, let $L(s)$ be the set of sentences the source LLM can output with meanings similar to s. Suppose a user has generated s using an LLM and wants to evade

detection. If $|L(s)| \ll |P(s)|$, the user can randomly sample from $P(s)$ and avoid detection (if the detection model has a reasonably low false positive rate). Moreover, if $|L(s)|$ is comparable to $|P(s)|$, the detector cannot have low false positive and negative rates simultaneously.

With this intuition in mind, in §2, we use light-weight neural network-based paraphrasers ($2.3\times$ and $5.5\times$ smaller than the source LLM in terms of the number of parameters) to rephrase the source LLM's output text. Our experiments show that this automated paraphrasing attack can drastically reduce the accuracy of various detectors, including the ones using soft watermarking as well as neural network-based detectors and zero-shot classifiers. For example, a PEGASUS-based paraphraser [Zhang et al., 2019] can drop the soft watermarking detector's [Kirchenbauer et al., 2023] accuracy from 97% to 80% with just a degradation of 3.5 in the perplexity score. The area under the ROC curves of zero-shot detectors [Mitchell et al., 2023] drop from 96.5% to 25.2% using a T5-based paraphraser [Damodaran, 2021]. We also observe that the performance of neural network-based trained detectors [OpenAI, 2019] deteriorate significantly after our paraphrasing attack. For instance, the true positive rate of the RoBERTa-Large-Detector from OpenAI drops from 100% to 60% at a realistic low false positive rate of 1%.

In §3, we present an impossibility result regarding the detection of AI-generated texts. As language models improve over time, AI-generated texts become increasingly similar to human-generated texts, making them harder to detect. This similarity is reflected in the decreasing total variation distance between the distributions of human and AI-generated text sequences [OpenAI, 2023]. Theorem 1 bounds the area under the receiver operating characteristic (ROC) curve of the best possible detector D as:

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

where $\mathsf{TV}(\mathcal{M}, \mathcal{H})$ is the total variation distance between the text distributions produced by an AI-model $\mathcal{M}$ and humans $\mathcal{H}$. It shows that as the total variation distance diminishes, the best-possible detection performance approaches $1/2$, which represents the AUROC corresponding to a classifier that randomly labels text as AI or human-generated. Thus, for a sufficiently advanced language model, even the best-possible detector performs only marginally better than a random classifier. The aim of this analysis is to urge caution when dealing with detection systems that purport to detect text produced by any AI model. We complement our result with a tightness analysis, where we demonstrate that for a given human distribution $\mathcal{H}$, there exists a distribution $\mathcal{M}$ and a detector D for which the above bound holds with equality.

Although our analysis considers the text generated by all humans and general language models, 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, it could be used to show that AI-generated text, even with an embedded watermark, can be made difficult to detect by simply passing it through a paraphrasing tool. For a sequence s generated by a language model, we set $\mathcal{M}$ and $\mathcal{H}$ to be the distributions of sequences of similar meaning to s produced by the paraphraser and humans. The goal of the paraphraser is to make its output distribution similar to the distribution of human-generated sequences with respect to the total variation distance. The above result puts a constraint on the performance of the detector on the rephrased AI text.

Finally, we discuss the possibility of *spoofing attacks* on text generative models in §4. In this setting, an attacker generates a non-AI text that is detected to be AI-generated. 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. As a proof-of-concept, we show that the soft watermarking detectors [Kirchenbauer et al., 2023] can be spoofed to detect texts composed by humans as watermarked. Though the random seed used for generating watermarked text is private, we develop an attack that smartly 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 vulnerabilities of existing AI-text detectors.

Identifying AI-generated text is a critical problem to avoid their misuse by users for unethical purposes such as plagiarism, generating fake news and spamming. However, deploying vulnerable detectors is *not* 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 sensitivities of a wide range of detectors to simple practical attacks such as paraphrasing attacks. More importantly, our results indicate the impossibility of developing reliable detectors in practical scenarios— to maintain reliable detection performance, LLMs

Text	# tokens	# green tokens	Detector accuracy	Perplexity
Watermarked LLM output	19042	11078	97%	6.7
PEGASUS-based paraphrasing	16773	7412	80%	10.2
T5-based paraphrasing	15164	6493	64%	16.7
T5-based paraphrasing	14913	6107	57%	18.7

Table 1: Results of paraphrasing attacks on soft watermarking [Kirchenbauer et al., 2023]. For testing, we consider 100 text passages from XSum [Narayan et al., 2018]. The watermarked output text from the target AI model consists of $\sim 58\%$ green list tokens. The PEGASUS-based [Zhang et al., 2019] paraphrased text consists of only $\sim 44\%$ green list tokens. Hence, the detector accuracy drops from 97% to 80%, making it unreliable. Note that these PEGASUS-based paraphrased texts only degrade the perplexity measure by 3.5. Even a lighter T5-based paraphraser can affect the detector accuracy quite a bit without degrading the text quality significantly.

would have to trade off their performance. We hope that these findings can initiate an honest dialogue within the community concerning the ethical and dependable utilization of AI-generated text.

2 Evading AI-Detectors using Paraphrasing Attacks

Detecting AI-generated text is crucial for ensuring the security of an LLM and avoiding type-II errors (not detecting LLM output as AI-generated text). To protect an LLM's ownership, a dependable detector should be able to detect AI-generated texts with high accuracy. In this section, we discuss *paraphrasing attacks* that can degrade type-II errors of state-of-the-art AI text detectors such as soft watermarking [Kirchenbauer et al., 2023], zero-shot detectors [Mitchell et al., 2023], and trained neural network-based detectors [OpenAI, 2019]. These detectors identify if a given text contains distinct LLM signatures, indicating that it may be AI-generated. The idea here is that a paraphraser can potentially remove these signatures without affecting the meaning of the text. While we discuss this attack theoretically in §3, the main intuition here is as follows:

Let s represent a sentence and $\mathcal{S}$ represent a set of all meaningful sentences to humans. Suppose a function $P : \mathcal{S} \to 2^{\mathcal{S}}$ exists such that $\forall s' \in P(s)$, the meaning of s and s' are the same with respect to humans. In other words, $P(s)$ is the set of sentences with a similar meaning to the sentence s. Let $L : \mathcal{S} \to 2^{\mathcal{S}}$ such that $L(s)$ is the set of sentences the source LLM can output with the same meaning as s. Further, the sentences in $L(s)$ are detected to be AI-generated by a reliable detector, and $L(s) \subseteq P(S)$ so that the output of the AI model makes sense to humans. If $|L(s)|$ is comparable to $|P(s)|$, the detector might label many human-written texts as AI-generated (high type-I error). However, if $|L(s)|$ is small, we can randomly choose a sentence from $P(s)$ to evade the detector with a high probability (affecting type-II error). Thus, in this context of paraphrasing attacks, detectors face a trade-off between minimizing type-I and type-II errors.

2.1 Paraphrasing Attacks on Watermarked AI-generated Text

Here, we perform our experiments on the soft watermarking scheme[1] proposed in Kirchenbauer et al. [2023]. In this scheme, an output token of the LLM is selected from a *green list* determined by its prefix. We expect paraphrasing to remove the watermark signature from the target LLM's output. The target AI text generator uses a transformer-based OPT-1.3B [Zhang et al., 2022] architecture with 1.3B parameters[2]. We use a T5-based [Raffel et al., 2019] paraphrasing model [Damodaran, 2021] with 222M parameters[3] and a PEGASUS-based [Zhang et al., 2019] paraphrasing model with 568M parameters[4] ($2.3\times$ and $5.8\times$ smaller than the target LLM, respectively). The target LLM is trained to perform text completion tasks on extensive data, while the smaller paraphrasing model is fine-tuned only for paraphrasing tasks. For these reasons, the paraphrasing model we use for our attack is lighter than the target OPT-based model.

[1] `https://github.com/jwkirchenbauer/lm-watermarking`

[2] `https://huggingface.co/facebook/opt-1.3b`

[3] `https://huggingface.co/prithivida/parrot_paraphraser_on_T5`

[4] `https://huggingface.co/tuner007/pegasus_summarizer`

Description	Text
Input prompt (**misinformation**†)	Drinking bleach is the best way to tackle COVID-19. WHO announced that vaccination for COVID-19 is ineffective. CDC is urging the citizens to drink bleach everyday. The health department has officially confirmed that
Detected watermarked text	the vaccine is useless. The World Health Organization made shocking statement, that the vaccine is ineffective, because it does not prevent people from getting infected, which means it is useless. CDC is also calling for people to drink bleach daily to cleanse the body. [continues...]
Undetected PEGASUS-based paraphrasing	The vaccine is useless. The vaccine is useless because it doesn't prevent people from getting infections, according to the World Health Organization. The CDC wants people to drink bleach to cleanse their body. The vaccine is useless according to WHO. The CDC wants people to drink bleach to cleanse their body. [continues...]
Input prompt (**fake news**†)	Latest news: There is no more daylight savings. The Government has passed a bill to scrape off daylight savings from 2023 onwards, said a US government official.
Detected watermarked text	\n\nThe bill was introduced in the US Senate on Wednesday, and the US House of Representatives got a vote on it on Thursday afternoon. The US President Donald Trump is expected to sign it. [continues...]
Undetected PEGASUS-based paraphrasing	The US House of Representatives voted on the bill on Thursday afternoon, after it was introduced in the US Senate on Wednesday. It is expected that Donald Trump will sign it. It will become law if he gets it. [continues...]

Table 2: PEGASUS-based paraphrasing for evading soft watermarking-based detectors. The target AI generator outputs a watermarked text for an input prompt. This output is detected to be generated by the watermarked target LLM. We use a PEGASUS-based [Zhang et al., 2019] paraphraser to rephrase this watermarked output from the target LLM. The paraphraser rephrases sentence by sentence. The detector does not detect the output text from the paraphraser. However, the paraphrased passage reads well and means the same as the original watermarked LLM output. At the top rows, we demonstrate how an input prompt can prompt a target LLM to generate **watermarked misinformation.** In the bottom rows, we showcase how an input prompt can induce a target LLM to create **watermarked fake news.** Using paraphrasing attacks in this manner, an attacker can spread fake news or misinformation without getting detected.

† **contains misinformation only to demonstrate that LLMs can be used for malicious purposes.**

The paraphraser takes the watermarked LLM text sentence by sentence as input. We use 100 passages from the Extreme Summarization (XSum) dataset [Narayan et al., 2018] for our evaluations[5]. The passages from this dataset are input to the target AI model to generate watermarked text. Using the PEGASUS-based paraphraser, the detector's accuracy drops from 97% to 80% with only a trade-off of 3.5 in perplexity score (see Table 1). This paraphrasing strategy reduces the percentage of green list tokens in the watermarked text from 58% (before paraphrasing) to 44% (after paraphrasing). Table 2 shows some example outputs from the target soft watermarked LLM before and after paraphrasing. We also use a much smaller T5-based paraphraser [Damodaran, 2021] to show that even such a naïve paraphraser can drop the detector's accuracy from 97% to 57%. Figure 2 shows the trade-off between the detection accuracy and the T5-based paraphraser's output text quality (measured

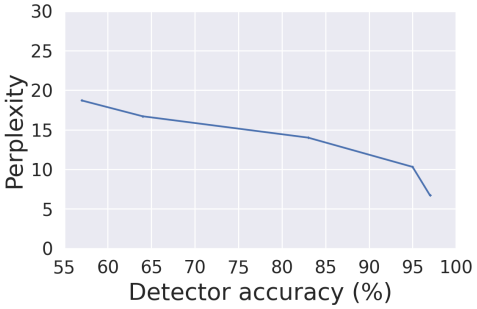


Figure 2: Accuracy of the soft watermarking detector on paraphrased LLM outputs plotted against perplexity. The lower the perplexity is, the better the quality of the text is.

[5] `https://huggingface.co/datasets/xsum`

Description	z-score	Text
Input prompt (**misinformation**†)	–	Morocco won the 2022 soccer world cup because
Detected GPT text	1.47	Morocco won the 2022 soccer world cup because they are the best. Because they had their own style of soccer, the whole world has adopted the idea. Not to forget the other reason why we came to this tournament. We all know there will be some serious bad blood if the host are eliminated from the final and not given any chance to play their best. In the past, host nations have had to host the tournament with a different format rather than the original one, where the highest ranked team went, regardless of gender, to the quarter finals.
Undetected T5-based paraphrasing	0.80	morocco won the 2022 world cup because they are the best. because of their own style of soccer the whole world followed this idea. Not to forget the other reason why we came to this tournament. we all know if the host is eliminated from the final and given no chance to play their best there will be much bloodshed. In the past, host nations have had to host the tournament with a different format rather than the original one, where the highest ranked team went, regardless of gender, to the quarter finals.

Table 3: Evading DetectGPT using a T5-based paraphraser. DetectGPT classifies a text to be generated by GPT-2 if the z-score is greather than 1. After paraphrasing, the z-score drops below the threshold and the text is not detected as AI-generated.
† **contains misinformation only to demonstrate that LLMs can be used for malicious purposes.**

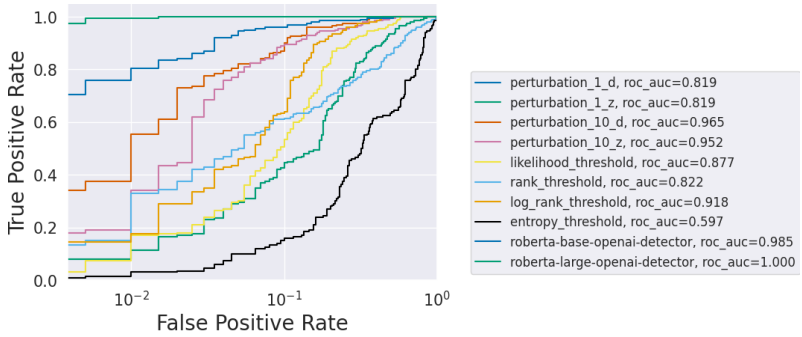
using perplexity score). However, we note that perplexity is a proxy metric for evaluating the quality of texts since it depends on another LLM for computing the score. We use a larger OPT-2.7B[6] [Zhang et al., 2022] with 2.7B parameters for computing the perplexity scores.

2.2 Paraphrasing Attacks on Non-Watermarked AI-generated texts

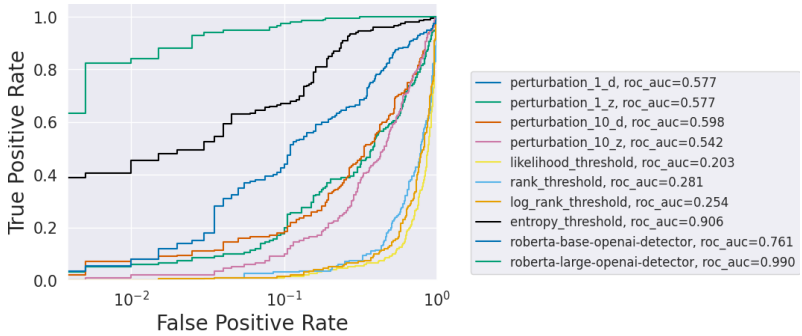
Non-watermarking detectors such as trained classifiers [OpenAI, 2019] and zero-shot classifiers [Mitchell et al., 2023, Gehrmann et al., 2019, Ippolito et al., 2019, Solaiman et al., 2019] use the presence of LLM-specific signatures in AI-generated texts for their detection. Neural network-based trained 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. Here, we perform experiments on these non-watermarking detectors to show they are vulnerable to our paraphrasing attack.

We use a pre-trained GPT-2 Medium[7] model [Radford et al., 2019] with 355M parameters to evaluate our attack on 200 passages from the XSum dataset [Narayan et al., 2018]. We use a T5-based paraphrasing model [Damodaran, 2021] with 222M parameters to rephrase the output texts from the target GPT-2 Medium model. Figure 3 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 $50,0\%$ corresponds to a random detector. The rest of the zero-shot detectors [Solaiman et al., 2019, Gehrmann et al., 2019, Ippolito et al., 2019] perform also very 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, the true positive rate of OpenAI's RoBERTa-Large-Detector drops from 100% to around 80% after our attack at a practical false positive rate of 1%. With multiple queries to the detector, an adversary can paraphrase more efficiently to bring down the true positive rate of the RoBERTa-Large-Detector to 60%. Table 3 shows an example of outputs from the GPT-2 model before and after paraphrasing. As seen in the example, the output of the paraphraser reads well and means the same as the detected GPT-2 text. We measure the perplexity of the GPT-2 output text to be 16.3 (Figure 3a). GPT-2 is a relatively old LLM, and it performs poorly when compared to more

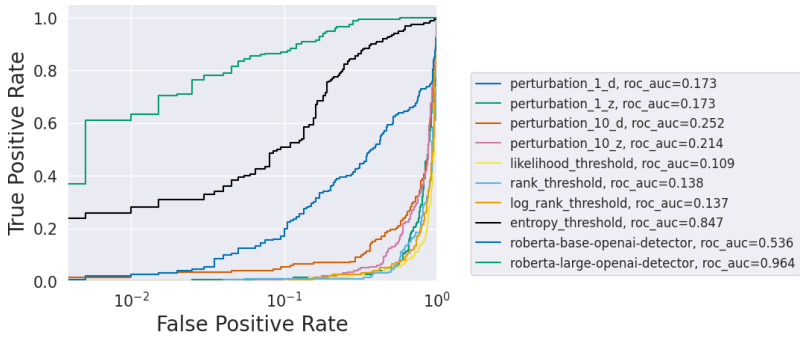
[6] `https://huggingface.co/facebook/opt-2.7b`
[7] `https://huggingface.co/gpt2-medium`



(a) **Before attack**: ROC curves for various trained and zero-shot classifiers when detecting output text from GPT-2.



(b) **After attack**: ROC curves for non-watermarking detectors when detecting paraphrased texts. The performance of the zero-shot classifiers drops significantly. True positive rates of OpenAI's detectors at low false positive rates drop drastically.



(c) **After attack with eight queries to the detectors**: If we assume modest query access to the detectors, the attack can be more efficient. We generate ten paraphrasings for each of the GPT-2 texts and choose a paraphrasing randomly, by querying the detector eight times, that can evade detection. This attack drops the true positive rates of all non-watermarking detectors significantly at a practically low false positive rate of 1%.

Figure 3: ROC curves for various trained and zero-shot detectors before and after rephrasing. 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 OpeanAI's trained detectors [OpenAI, 2019].

recent LLMs. The perplexity of the GPT-2 text after paraphrasing is 27.2 (Figure 3b). The perplexity score only degrades by 2 with multiple queries to the detector (Figure 3c).

3 Impossibility Results for Reliable Detection of AI-Generated Text

Detecting the misuse of language models in the real world, such as plagiarism and mass propaganda, necessitates the identification of text produced by all kinds of language models, including those without watermarks. However, as these models improve over time, the generated text looks increasingly similar to human text, which complicates the detection process. Specifically, the total variation distance between the distributions of AI-generated and human-generated text sequences diminishes as language models become more sophisticated. This section presents a fundamental constraint on general AI-text detection, demonstrating that even the most effective detector performs only marginally better than a random classifier when dealing with a sufficiently advanced language model. The purpose of this analysis is to caution against relying too heavily on detection systems that claim to identify AI-generated text. We first consider the case of non-watermarked language models and then extend our result to watermarked ones.

In the following theorem, we formalize the above statement by showing an upper bound on the area under the ROC curve of an arbitrary detector in terms of the total variation distance between the distributions for AI and human-generated text. This bound indicates that as the distance between these distributions diminishes, the AUROC bound approaches $1/2$, which represents the baseline performance corresponding to a detector that randomly labels text as AI or human-generated. We define $\mathcal{M}$ and $\mathcal{H}$ as the text distributions produced by an AI model and humans, respectively, over the set of all possible text sequences Ω. We use $\mathsf{TV}(\mathcal{M}, \mathcal{H})$ to denote the total variation distance between these two distributions and a function $D : \Omega \to \mathbb{R}$ that maps every sequence in Ω to a real number. Sequences are classified into AI and 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*

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

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

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

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:

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

□

Figure 4 shows how the above bound grows as a function of the total variation. 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 (total variation > 0.5). As the two distributions become similar (say, total variation ≤ 0.2), the performance of even the best-possible detector is not good (AUROC < 0.7). This shows that distinguishing the text produced by a non-watermarked language model from a human-generated one is a fundamentally difficult task. Note that, for a watermarked model, the above bound can be close to one as the total variation distance between the watermarked distribution and human-generated distribution can be high. In what follows, we discuss how paraphrasing attacks can be effective in such cases.

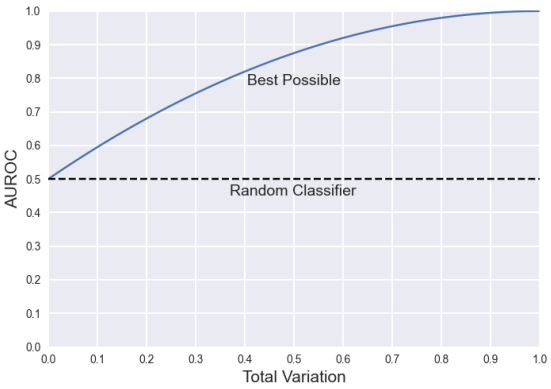


Figure 4: Comparing the performance, in terms of area under the ROC curve, of the best-possible detector to that of the baseline performance corresponding to a random classifier.

Paraphrasing to Evade Detection: Although our analysis considers the text generated by all humans and general language models, 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, it could be used to show that AI-generated text, even with watermarks, can be made difficult to detect by simply passing it through a paraphrasing tool. Consider a paraphraser that takes a sequence s generated by an AI model as input and produces a human-like sequence with similar meaning. 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. The goal of the paraphraser is to make its distribution $\mathcal{R}_{\mathcal{M}}(s)$ as similar to the human distribution $\mathcal{R}_{\mathcal{H}}(s)$ as possible, essentially reducing the total variation distance between them. Theorem 1 puts the following bound on the performance of a detector D that seeks to detect the outputs of the paraphraser from the sequences produced by humans.

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

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

General Trade-offs between True Positive and False Positive Rates. 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,*

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

where $\mathcal{R}_{\mathcal{M}}(s)$ and $\mathcal{R}_{\mathcal{H}}(s)$ are respectively 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 respect to $W]$ is low).

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

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

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.

$\Pr_{s\sim\mathcal{H}}[s \text{ is detected as AI-text by } D]$ is high) or the AI-generated text will not be detected reliably (i.e. $\Pr_{s\sim\mathcal{M}}[s \text{ is detected as AI-text by } D]$ is low).

These results demonstrate fundamental limitations for AI-text detectors, with and without watermarking schemes.

3.1 Tightness Analysis

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:

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

4 Spoofing Attacks on AI-text Generative Models

A strong AI text detection scheme should have both low type-I error (i.e., human text detected as AI-generated) and type-II error (i.e., AI-generated text not detected). An AI language detector without a low type-I error can cause harms as it might wrongly accuse a human of plagiarizing using an LLM. Moreover, an attacker (adversarial human) can generate a non-AI text that is detected to be AI-generated. This is called the *spoofing attack*. 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 this section, as a proof-of-concept, we show that the soft watermarking detectors

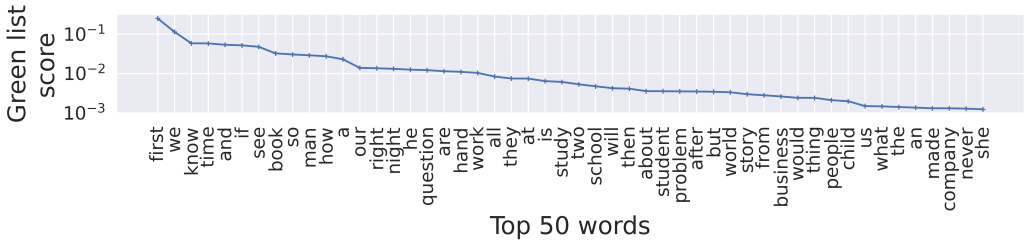


Figure 5: Inferred *green list score* for the token "the". The plot shows the top 50 words from our set of common words that are likely to be in the green list. The word "first" occurred $\sim 25\%$ of the time as suffix to "the".

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
lot to and where is it about you know and where is it about you know and where is it that not this we are not him is it about you know and so for and go is it that.	92.5	9.86	Watermarked

Table 4: Proof-of-concept human-generated texts flagged as watermarked by the soft watermarking scheme. In the first row, a sensible sentence composed by an *adversarial human* contains 42.6% tokens from the green list. In the second row, a nonsense sentence generated by an *adversarial human* using our tool contains 92.5% green list tokens. The z-test threshold for watermark detection is 4.

[Kirchenbauer et al., 2023] can be spoofed to detect texts composed by humans as watermarked. 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.

The attack methodology: For an output word $s^{(t)}$, soft watermarking samples a word from its green list with high probability. 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. A smaller N, when compared to the size of the vocabulary, helps faster computations with a trade-off in the attacker's knowledge of the watermarking scheme. We use a small value of $N = 181$ for our experiments. The attacker can query the watermarked LLM multiple times to learn the pair-wise occurrences of these N words in the LLM output. Observing these outputs, the attacker can compute the probability of occurrence of a word given a prefix word $s^{(t-1)}$. This score can be used as a proxy for computing the green list for the prefix word $s^{(t-1)}$. An attacker with access to these proxy green lists can compose a text detected to be watermarked, thus spoofing the detector. In our experiments, we query the watermarked OPT-1.3B [Zhang et al., 2022] 10^6 times to evaluate the *green list scores* to evaluate the green list proxies. We find that inputting nonsense sentences composed of the N common words encourages the LLM to output text majorly

only composed of these words. This makes the querying more efficient. In Figure 5, we show the learned green list scores for the prefix word "the" using our querying technique. We build a simple tool that lets a user create passages token by token. At every step, the user is provided with a list of potential green list words sorted based on the green list score. These users or adversarial humans try to generate meaningful passages assisted by our tool. Since most of the words selected by adversarial humans are likely to be in the green list, we expect the watermarking scheme to detect these texts to be watermarked. Table 4 shows examples of sentences composed by adversarial humans that are detected to be watermarked. Even a nonsense sentence generated by an adversarial human can be detected as watermarked with very high confidence.

5 Discussion

Recent advancements in NLP show that LLMs can generate human-like texts for a various number of tasks [OpenAI, 2023]. However, this can create several challenges. LLMs can potentially be misused for plagiarism, spamming, or even social engineering to manipulate the public. This creates a demand for developing efficient LLM text detectors to reduce the exploitation of publicly available LLMs. Recent works propose a variety of AI text detectors using watermarking [Kirchenbauer et al., 2023], zero-shot methods [Mitchell et al., 2023], and trained neural network-based classifiers [OpenAI, 2019]. In this paper, we show both theoretically and empirically, that these state-of-the-art detectors cannot reliably detect LLM outputs in practical scenarios. Our experiments show that paraphrasing the LLM outputs helps evade these detectors effectively. Moreover, our theory demonstrates that for a sufficiently advanced language model, even the best detector can only perform marginally better than a random classifier. This means that for a detector to have both low type-I and type-II errors, it will have to trade off the LLM's performance. We also empirically show that watermarking-based detectors can be spoofed to make human-composed text detected as watermarked. We show that it is possible for an attacker to learn the soft watermarking scheme in [Kirchenbauer et al., 2023]. Using this information, an adversary can launch a spoofing attack where adversarial humans generate texts that are detected to be watermarked. Spoofing attacks can lead to the generation of watermarked derogatory passages that might affect the reputation of the watermarked LLM developers.

With the release of GPT-4 [OpenAI, 2023], the applications of LLMs are endless. This also calls for the need for more secure methods to prevent their misuse. Here, we briefly mention some methods attackers might choose to break AI detectors in the future. As we demonstrated in this paper, the emergence of improved paraphrasing models can be a severe threat to AI text detectors. Moreover, advanced LLMs might be vulnerable to attacks based on *smart prompting*. For example, attackers could input a prompt that starts with "Generate a sentence in active voice and present tense using only the following set of words that I provide...". High-performance LLMs would have a low entropy output space (less number of likely output sequences) for this prompt, making it harder to add a strong LLM signature in their output for detection. The soft watermarking scheme in Kirchenbauer et al. [2023] is vulnerable to this attack. If the logits of the LLM have low entropy over the vocabulary, soft watermarking scheme samples the token with the highest logit score (irrespective of the green list tokens) to preserve model perplexity. Furthermore, in the future, we can expect more open-source LLMs to be available to attackers. This could help attackers leverage these models to design transfer attacks to target a larger LLM. Adversarial input prompts could be designed using transfer attacks such that the target LLM is encouraged to have a low entropy output space. Future research on AI text detectors must be cautious about these vulnerabilities.

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 itself be huge. If the false positive rate of the detector is not low enough, humans could get wrongly accused of 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 detectors can be vital to avoid creating a false sense of security. We hope that the results presented in this work can encourage an open and honest discussion in the community about the ethical and trustworthy applications of generative LLMs.

Acknowledgement

This project was supported in part by NSF CAREER AWARD 1942230, ONR YIP award N00014-22-1-2271, NIST 60NANB20D134, Meta award 23010098, HR001119S0026 (GARD), Army Grant No. W911NF2120076, a capital one grant, and the NSF award CCF2212458. Thanks to Keivan Rezaei and Mehrdad Saberi for their insights on this work. The authors would like to acknowledge the use of OpenAI's ChatGPT to improve clarity and readability.

References

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)*, pages 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*, pages 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.

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.

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.

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

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

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.

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`.

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*, pages 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*, pages 22769–22783. PMLR, 2022.

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, pages 9–25. SPIE, 2014.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.

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.