International Journal of Speech Technology (2024) 27:861–874
https://doi.org/10.1007/s10772-024-10144-2

Robustness of generative AI detection: adversarial attacks on black-box neural text detectors

Vitalii Fishchuk[1] · Daniel Braun[2]

Received: 11 August 2024 / Accepted: 1 September 2024 / Published online: 16 October 2024

Abstract

The increased quality and human-likeness of AI generated texts has resulted in a rising demand for neural text detectors, i.e. software that is able to detect whether a text was written by a human or generated by an AI. Such tools are often used in contexts where the use of AI is restricted or completely prohibited, e.g. in educational contexts. It is, therefore, important for the effectiveness of such tools that they are robust towards deliberate attempts to hide the fact that a text was generated by an AI. In this article, we investigate a broad range of adversarial attacks in English texts with six different neural text detectors, including commercial and research tools. While the results show that no detector is completely invulnerable to adversarial attacks, the latest generation of commercial detectors proved to be very robust and not significantly influenced by most of the evaluated attack strategies.

Keywords Large language models · Neural text detection · Adversarial attacks · Generative AI

1 Introduction

The ubiquitous availability and accessibility of generative AI models that can produce texts, like OpenAI's ChatGPT or Google's Gemini, has resulted in a proliferation of such texts, even in contexts where their use is explicitly forbidden by policies, e.g. in scientific publishing or higher education. The increased quality and human-likeness of AI generated texts makes the enforcement of such policies increasingly difficult.

So-called "AI detectors" or "neural text detectors" are models that have been trained to classify whether a given text was generated by an AI. Many educational institutions and publishers rely on such detectors to enforce their generative AI policies. While scientific evaluations of such detectors have shown different degrees of reliability (see Sect. 2.1), these evaluations regularly do not account for the fact that, especially in scenarios where the use of generative AI is explicitly forbidden, users can and will take active steps to disguise that a text has been generated by an AI. Such measures can be considered as adversarial attacks (see Sect. 2.2). Such attacks exploit the fact that machine learning models by identifying patterns in the data rather than by understanding actual underlying concepts. Consequently, introducing small, human-unnoticeable perturbations can result in misclassification (Goodfellow et al., 2014; Szegedy et al., 2013).

Adversarial attacks can be categorised into black-box and white-box attacks (Peng et al., 2023). In white-box attacks, the attacker has full access to the target model, including its parameters, architecture, and loss function (Ebrahimi et al., 2018; Gao et al., 2018). During black-box attacks, the adversary can only input queries and observe the outputs without any insights into internal processing (Gao et al., 2018). Furthermore, it can be distinguished between targeted and untargeted attacks, where targeted attacks aim at triggering misclassification towards a specific label, while untargeted aim to cause any misclassification (Rathore et al., 2020).

Building on our previous work (Fishchuk & Braun, 2023), in this article, we investigate the robustness of scientific and commercial AI detectors, by evaluating the effectiveness of different resource-efficient adversarial attack

✉ Vitalii Fishchuk
v.fishchuk@student.utwente.nl

✉ Daniel Braun
d.braun@utwente.nl

[1] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands

[2] Department of High-tech Business and Entrepreneurship, University of Twente, Enschede, The Netherlands

strategies. Particularly we use texts generated by GPT 3.5 to compare the robustness of six neural text detectors in a black-box scenario, namely: Copyleaks, GPTZero, Radar, GPT-2 Output Detector, Turnitin, and Open AI Text Classifier (see Sect. 3.1). The adversarial attack strategies presented in this article cover a wide range of methodologies, from prompt engineering to hyperparameter-tweaking and post-processing of texts.

While the results show that no detector is completely invulnerable to adversarial attacks, they also show that the latest generation of commercial detectors, like Copyleaks, and GPTZero, is significantly more robust than previously tested detectors and not significantly influenced by most of the evaluated attack strategies.

2 Related work

2.1 AI detection

The detection of AI generated texts has been a concern before the ubiquitous availability of Large Language Models (LLMs) like ChatGPT. Jawahar et al. (2020), for example, in 2020 already presented a survey on the "Automatic Detection of Machine Generated Text", mostly focusing on GPT-2 generated text. However, since 2023, the interest in the area has significantly increased. Weber-Wulff et al. (2023), for example, compared 14 AI detection services. In their evaluation, none of the tested tools achieved an accuracy above 76%.

In a similar experiment, Elkhatat et al. (2023) compared five classifiers, four of which also were part of the study by Weber-Wulff et al. (2023). In the study by Elkhatat et al. (2023), the best-performing detector achieves a recall of 93% and a precision of 80%, significantly higher than the results reported by Weber-Wulff et al. (2023). Habibzadeh (2023) particularly focused on the detection of AI generated texts in the medical domain and the detector GPTZero. Habibzadeh (2023) reports an accuracy of 0.80 for the detector, while Weber-Wulff et al. (2023) report an accuracy of only 54% for the very same tool. A comparison by van Oijen (2023) includes seven different detectors, in which the best achieved an accuracy of just 50%. Among the tested tools is Crossplag, which achieved an accuracy of 30%, compared to an accuracy of 69% that the tool achieved in the evaluation by Weber-Wulff et al. (2023).

Table 1 shows an overview of the reported accuracy for different neural text detectors in the literature. If anything, these results show that the ability to detect AI generated texts is highly dependent on the individual documents that are evaluated and cannot easily be generalised across domains and texts. Additionally, the presented evaluations are mainly focused on texts that are directly generated by AI models without being designed in any way to escape automated detection. We believe that in practice, AI detector tools will be mostly applied in contexts where the texts they will assess have been particularly optimised to escape such detection. A student who e.g. submits a course work that has been generated by AI and does not disclose it will most likely take actions to obfuscate the fact that AI was involved in the generation of the text. Therefore, we believe it is important to not just assess detector tools with standard texts, but particularly also assess their robustness with regard to adversarial attacks.

2.2 Adversarial attacks

Most of the existing literature about adversarial attacks focuses on image detection (Kong et al., 2021; Akhtar et al., 2021; Xu et al., 2020). Textual input is less used due to its discrete nature and the difficulty in introducing human-imperceptible perturbations, contrary to the image data, where a change in a few hundred pixels can go unnoticed (Jin et al., 2019; Peng et al., 2023). Examples of adversarial attacks on general text classification models include the work by Ebrahimi et al. (2018) and Gao et al. (2018). More recent work has started to specifically look into adversarial attacks on neural text detectors: Wolff and Wolff (2022) showed that introducing spelling mistakes and replacing characters with homoglyphs can significantly reduce the detection rate for GPT-2 texts. Liang et al. (2023a) showed that similar character-level mutation-based attacks are also successful for RoBERTa-based detection models. Liang et al. (2023c) not only showed that existing detectors are vulnerable to simple rephrasing, but they also showed that they are biased towards flagging texts that have been (manually) written by non-native speakers as AI-generated.

Because currently available methods are vulnerable to adversarial attacks, multiple suggestions have been made to improve their robustness, e.g. by Liang et al. (2023b), Shen et al. (2023), Crothers et al. (2022), and Yoo et al. (2022). While watermarking techniques to identify AI-generated texts are also investigated, they are generally seen as vulnerable to adversarial attacks, especially to mutation and paraphrasing-based approaches (Jin et al., 2019; Kirchenbauer et al., 2023; Sadasivan et al., 2023).

3 Experimental setup

This section describes the experimental setup that was used in this study. Neural language generation and detection are both very dynamic fields. Therefore, it is important context that the experiments described in this article were conducted between January and April 2024. By the time of publication,

Table 1 Accuracy of different neural text detectors as reported in the literature

Detector	Generator					
	ChatGPT	GPT 3.5, 4	GPT 2-4, Davinci, Flan-T5	GPT 3.5-4, Gemini, Mistral, LLaMa 2	ChatGPT	ChatGPT
Compilatio	74					
Content at Scale	33	71				0
Crossplag	69	80				30
DetectGPT	46			77		
Go Winston	67					
GPT Zero	54	81	64	94	58	30
GPT-2 Output Detector	72					
OpenAI Text Classifer	54	78				40
PlagiarismCheck	39					
Turnitin	76	100				
Writeful GPT Detector	43					
Writer	50	71	69		50	0
Zero GPT	59	87			58	
Copyleaks		100			83	44
Originality.ai		98	97	94		
Scribber		88			50	
IvyPanda		77				
GPT Radar		76				
SEO.ai		72				
Sapling		65	67			
ContentDetector.ai		63				
Grammica		86				
GPTKIT			55			
Zylalab			68			
Checkfor.ai	59			99		
Undetectable.ai					75	
Corrector App						50
Source	Weber-Wulff et al. (2023)	Walters (2023)	Akram (2023)	Emi and Spero (2024)	Cooperman and Brandão (2024)	van Oijen (2023)

the capabilities of both, generators and detectors, might have changed significantly.

3.1 Detectors

In addition to the experiments described in this article, in 2023, we conducted a series of baseline experiments to establish whether neural text detectors are in general vulnerable to adversarial attacks. For these baseline experiments, we used three neural text detectors: The GPT-2 Output Detector, the now discontinued OpenAI Text Classifier, and the commercial detector of Turnitin (Fishchuk & Braun, 2023), After the baseline experiments were successful, we decided to extend our experiments, to include more attack strategies and more detectors, particularly commercially available ones. Therefore, we now report additional experiments with Radar, GPTZero, Copyleaks, and also the GPT-2 Output Detector.

3.1.1 GPT-2 output detector

The GPT-2 Output Detector is an open source detection model that was trained by OpenAI, by fine-tuning a RoBERTa model with outputs from the GPT-2 model. The underlying Roberta model enforces an input limitation of 512 tokens, which makes it necessary to truncate longer texts (Solaiman et al., 2019).

3.1.2 Radar

Radar is another open source detection model. It was trained by Hu et al. (2023) by fine-tuning a RoBERTa model, and, according to the developers, significantly outperforms other AI-content detectors, particularly in settings where paraphrasing is applied to the generated texts. Like the GPT-2 Output Detector, Radar is also limited to inputs of 512 tokens.

3.1.3 OpenAI text classifier

The OpenAI Text Classifier was a neural text detector that was developed by OpenAI and made accessible through their website, following the spike in ChatGPT's popularity. While the OpenAI Text Classifier has been evaluated in the baseline experiments, it has since been discontinued and is therefore not part of the newly conducted experiments.

3.1.4 Turnitin

Turnitin is a commercial vendor for a wide range of education-related software, including tools for plagiarism detection. Relatively recently, Turnitin expanded their portfolio by also offering a neural text detection component.[1]

3.1.5 GPTZero

GPTZero[2] is a widely deployed commercial detector trained to specifically detect GPT-4, GPT-3.5, Bard, LLaMa and other new AI generation models. GPTZero AI detection is accessible through a web interface and an API, which was used in this study.

3.1.6 Copyleaks

Copyleaks[3] is another commercial vendor that offers a wider range of education-related software, including plagiarism detection and neural text detectors. Copyleaks is also accessible through both a web-interface and an API. For the presented study, the API endpoint was used.

3.2 Measurement

All detectors evaluated in this study assess texts with a score between 0 and 1, where 0 means that the tool is certain that the text was generated by a human and 1 means that the tool is certain that the text was generated by an AI. This metric is referred to as the AI-detection score in the subsequent text. We will consider an adversarial attack as successful if it lowers the average AI-detection score of a detector below 0.5.

3.3 Corpus

The texts used for the experiments have been generated with the GPT-3.5-turbo-0125 model through the OpenAI API.[4] The prompting schema used to generate the baseline texts is shown in Listing 1. The model's hyperparameters have been set to the default values specified in the API in the baseline setting. The generated texts have been limited to 800 tokens, in order to generate texts with an approximate length of 400 words. Unless stated otherwise in the experiment setup, all texts are produced using a single-query approach.

```
Write a four−hundred−word <style> essay on the topic '<topic>'.
```

Listing 1: Basic prompt for text generation

A list of 200 essay topics Nova (2019), grouped by essay genre, was used as input for the text production. For each of the following genres, texts have been generated for 20 different topics:

- argumentative
- cause and effect
- compare contrast
- controversial argumentative
- descriptive
- expository
- funny argumentative
- narrative
- persuasive
- research

Due to the high fluctuation of the text size produced by the GPT-3.5 model, a limit of 300–500 words was established for each text. If a generated text was not within these borders, the text has been regenerated. All texts that have been used in the experiments, together with the code for the experiments, are available on GitHub.[5]

3.4 Attacks

Based on the existing literature and our previous work, we identified seven promising resource-efficient approaches for adversarial attacks which are introduced in this section. The evaluated attack approaches can be categorised into three categories: prompt engineering (Sects. 3.4.1 to 3.4.3), hyperparameter tweaking (Sect. 3.4.4), and post-processing (Sects. 3.4.5 to 3.4.7).

[1] https://www.turnitin.com/solutions/topics/ai-writing/.

[2] https://gptzero.me/.

[3] https://copyleaks.com/.

[4] https://platform.openai.com/docs/overview.

[5] https://github.com/DaBr01/Adversarial-Attacks-on-Neural-Text-Detection/.

3.4.1 Text characteristics prompt engineering

We examined the effect of essay genres and tones on detection scores to determine the possibility of potential attack vectors. To assess the impact of essay genre on detection, we generate 20 texts per genre category each with a different genre-specific topic. As described in Sect. 4, we did not find a significant difference in text detection across different genres. Therefore, in the following experiments, we limited the number of samples by choosing "research" as genre. This allowed us to perform more experiments and assess a broader picture of detector resistance to adversarial attacks.

In addition, we assessed the impact of different essay tones on the detection. Five formality tones (informal, semi-formal, formal, academic, and professional) have been selected and tested with two methods: simple and extended. In the simple variant of the attack, the model was prompted to follow the specified tone and rely on the model's interpretation of the style. In the extended version, the prompts were supplemented with a list of instructions detailing how to achieve the provided tone.

3.4.2 Detection avoidance prompt engineering

Different prompts aimed at tweaking the text structure and diversity have been tested. Additionally, we tested the difference between single-query and two-query prompt engineering, where the former refers to providing all of the instructions together with a topic in a single prompt, and the latter—providing instructions after the topic asking to re-write the essay:

- asking to avoid detection
- asking for a "good balance" of perplexity and burstiness
- maximise perplexity
- repeat a set of characteristics distinguishing human texts

3.4.3 Outlier prompt engineering

While evaluating the previously described attack strategies, we noticed a set of outlier texts with significantly lowered detection scores while bearing no visual difference. Based on the top-level analysis of these texts, we derived a set of characteristics noticed only in these texts. In this experiment, we attempted to replicate said characteristics using prompt engineering. The identified characteristics are:

- in-text citations
- in-text citations with a reference list
- first-person tone
- blend of different formality tones
- unusual grammar and sentence structures
- in-text numerical data

3.4.4 Hyperparameter-tweaking

Tweaking hyperparameters was tested by producing texts with different combinations of parameter values. Due to the large amount of possible combinations, we limited the experiments to five essay topics.

Firstly, we conduct a preliminary search across four parameters: temperature, top p, presence penalty, and frequency penalty. Temperature and top p control randomness in the text. By increasing the temperature, the output becomes more random. However, for values beyond the default of 1.0, the length of the outputs started fluctuating strongly and the quality of the texts dropped. Top p represents the percentage of tokens selected based on their probability mass. The frequency penalty controls the frequency of tokens appearing in the text, with higher values leading to more diverse verbatim. The presence penalty controls the model's likelihood of repeating tokens in the text. Higher values of presence penalty lead to the model producing more diverse texts (OpenAI, 2023).

We set the search boundaries for the parameter search to [− 1, 2] for frequency and presence penalty bounds and [0, 1.5] for temperature. Top p is explored in the whole available range of [0, 1]. The boundaries have been chosen because texts generated outside these boundaries often have a very low quality. We only consider an attack strategy as successful if it does not significantly reduce the quality of the generated texts. We select a step size of 0.2 for frequency and presence penalty and 0.1 for temperature and top p to keep the number of samples similar for each of the parameters.

We further build upon the previous experiment by conducting a linear search of frequency and presence penalty in the bounds of [0, 2] with a smaller step of 0.1. Additionally, we supplement the research genre with argumentative (5 topics each) to explore the effects of different genres on the effectiveness of parameter tweaking attacks.

Finally, we conduct a grid search over a combination of frequency and presence penalties while limiting the frequency penalty to a range of [0, 1] and the presence penalty to [0, 2]. Frequency penalty is analysed in the smaller range due to the negative impact on text quality of higher values.

3.4.5 Character mutations

A popular set of adversarial attack strategies for texts are character-level mutations. With this attack strategy, we attempt to lower detection by replacing certain characters in the text with visually similar but different characters. We select a wide range of these attacks and test them utilising an all-in approach: If a certain character is replaced with a visually similar one, all instances of that character in the text are replaced. Additionally, we test so-called invisible characters, a set of UTF symbols used for text formatting

Table 2 Character mutation attacks

Short name	Perturbation type	Mappings (Latin Character: UTF Code)
Cyrillic-Full	Replace English with Cyrillic All similarly looking characters	a: U+0430, c: U+0441, e: U+0435 i: U+0456, l: U+04CF, o: U+043E p: U+0440, x: U+0445, y: U+0443 h: U+04BB, w: U+051D, j: U+0458 s: U+0455
Cyrillic-Simple	Replace English with Cyrillic Identically looking characters only	a: U+0430, c: U+0441, e: U+0435 i: U+0456, o: U+043E, p: U+0440 x: U+0445, y: U+0443, j: U+0458 s: U+0455
Armenian	Replace Armenian	o: U+0585, h: U+0570, g: U+0581 u: U+057D, n: U+0578
Greek	Replace Greek	a: U+03B1, y: U+03B3, v: U+03BD o: U+03BF, p: U+03C1
Punctuation	Replace Punctuation	,: U+201A,.: U+FF0E
Invisible	Add Invisible Characters	Zero Width Space: U+200B Zero Width Non-Joiner: U+200C Zero Width Joiner: U+200D
Various	Various Replacements	u: U+1D1C, d: U+217E, g: U+0261 q: U+051B, v: U+1D20
19-char	Replacing 19 English characters	A combination of Cyrillic-Full Armenian and Greek
Combined	Combined Attack	All mappings from Major Cov, Punctuation, and Invisible
L-I	Lowercase L to Uppercase I Swap Attack	l: I (U+0049)

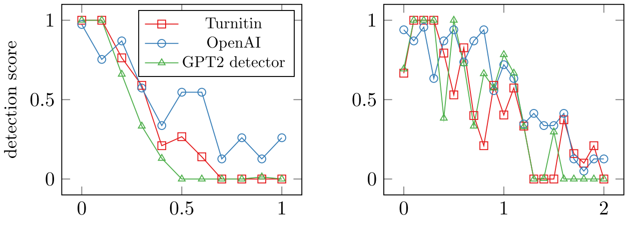
and invisible in most text readers. Table 2 shows the list of all character mutations investigated.

3.4.6 Translation

Another attack strategy that we evaluated is translation: By first translating the original English texts into an intermediate language and subsequently back into English, characteristics distinct to the model that originally generated the text could be lost. Particularly, we used Google Translate to translate the texts into four intermediary languages: Chinese, Arabic, Japanese, and Russian.

3.4.7 Paraphrasing

Following the translation experiment, we further attempt to change the text structure and characteristics via automatic paraphrasing tools. Two tools are evaluated, the open source Parrot paraphraser (Damodaran, 2021) and the commercial tool Quillbot.[6]



(a) frequency penalty

(b) presence penalty

Fig. 1 Influence of the frequency and presence penalty on the detection score

4 Results

4.1 Baseline experiments

In the baseline experiments described in Fishchuk and Braun (2023), we explored a set of resource- and query-efficient black-box adversarial attack methods, namely, parameter tweaking, prompt engineering, and character-level mutations, with the GPT-2 Output Detector, the OpenAI Text Classifier, and Turnitin.

Frequency and presence penalty adjustments were found to be efficient attack strategies. When applied in combination

[6] https://quillbot.com/.

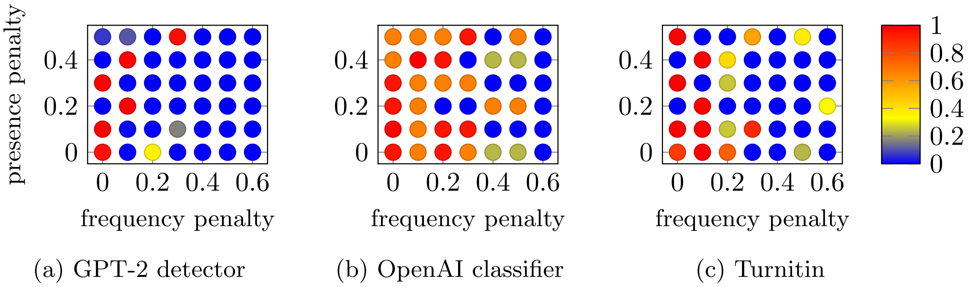
(see Fig. 2) and separately (see Fig. 1), they reduced detection scores below 0.5 while maintaining high text quality.

On the contrary, basic prompt engineering was found to be reliable only for bypassing the GPT-2 Output detector, and only the burstiness and perplexity prompt worked against Turnitin. Furthermore, a slight but consistent drop in detection rates was perceived for all second-query approaches. Finally, the study found character level mutations causing a substantial drop in detection scores for the GPT-2 detector and OpenAI classifier. Although Turnitin was able to classify the Latin-Cyrillic swap as an attempt to avoid detection, the l-I swap was found to drop the mean detection score to 0.21. Overall, the l-I swap was the only character mutation technique showing the lowest mean scores for all three of the detectors. The detailed results can be found in Fishchuk and Braun (2023).

4.2 Text characteristics

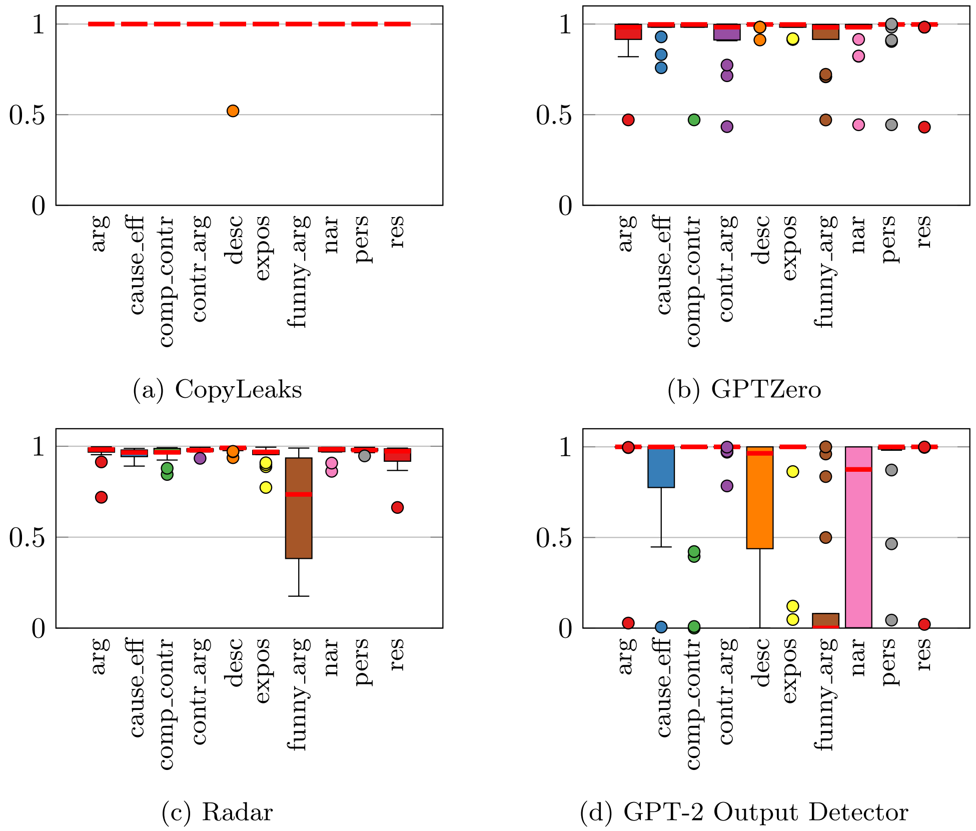
4.2.1 Genre

As shown in Fig. 3, the essay genre had little to no impact on the commercial detectors (GPTZero, Copyleaks). The GPT-2 Output Detector and Radar, however, showed lower detection scores for the "funny-argumentative" genre, with the most significant impact perceived in the scores of the GPT-2 Output Detector. While detection scores were in general lower for Radar, in most cases the score stayed clearly above 0.5, leading to a likely classification as an AI text in a real-world scenario. Consequently, we conclude that there is no significant impact of the genre on the detection scores across the three detectors GPTZero, Copyleaks, and Radar.



(a) GPT-2 detector (b) OpenAI classifier (c) Turnitin

Fig. 2 Influence of joined optimisation of frequency and presence penalties on the detection score



(a) CopyLeaks (b) GPTZero

(c) Radar (d) GPT-2 Output Detector

Fig. 3 Influence of the genre on the score of the different detectors

4.2.2 Tone

The first iteration of the experiment, conducted by only requesting the model to follow the specific tone without providing any instructions, did not produce any difference in the detection scores across all four detectors. The second iteration, on the other hand, where the model was provided with instructions detailing how to follow the tone, revealed lower detection scores for GPT-2 and Radar for informal tone texts. The GPT-2 Output Detector consistently classified informal texts as human with scores near 0 (see Fig. 4). Radar scores also consistently dropped, however only to a median of 0.8. The results of the second iteration were further validated on argumentative genre topics, showing the same trend. Overall, the tone only had a relevant impact on the performance of the GPT-2 Output Detector for an informal tone.

4.3 Hyperparameter-tweaking

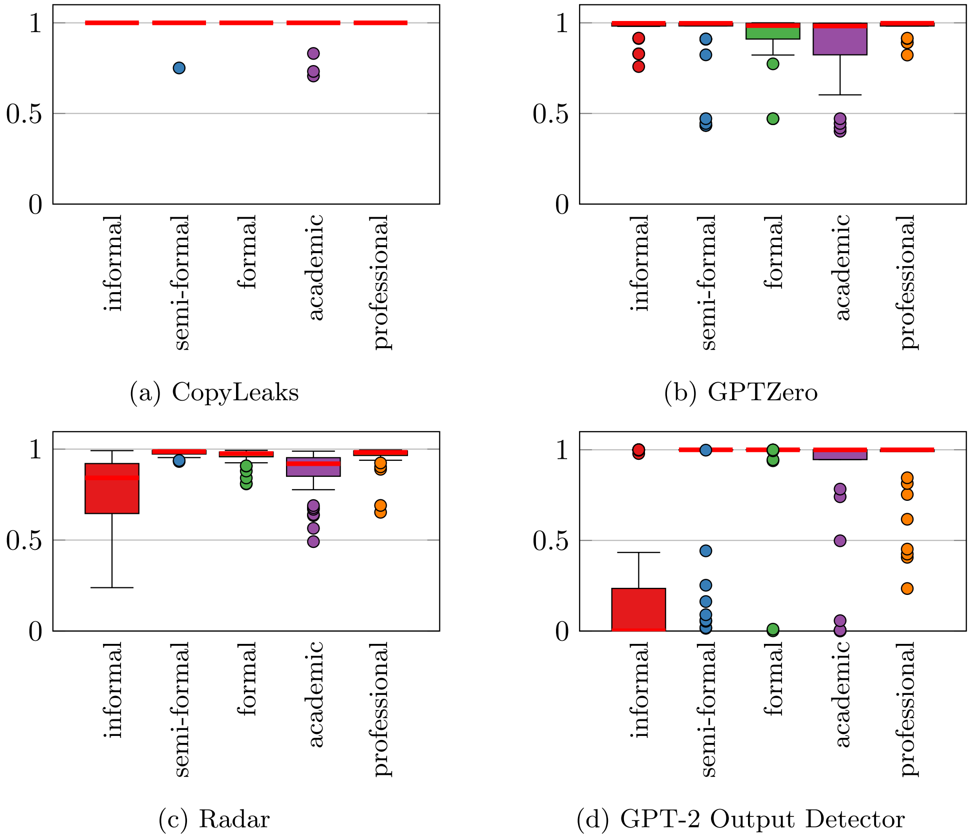
A preliminary linear search of temperature, top p, frequency and presence penalty showed a steady drop in detection scores for frequency penalty past values of 1 (see Fig. 5), which can be attributed to a significant drop in text quality observed with higher parameter values. Texts generated with a frequency penalty closer to 2 are low-quality and require substantial manual editing. Nevertheless, these texts are still classified above 0.8 by Copyleaks and slightly above 0.5 by Radar. GPTZero and GPT-2 detector scores fell below 0.5, with a frequency penalty above 1. We did not observe any significant impact of temperature, top p and presence penalty on detectors besides the GPT-2 Output Detector.

Following the preliminary search, we conducted extended research into frequency and presence penalties, supplementing the search with additional topics from the argumentative genre. This new experiment confirms the effect of the frequency penalty on detection and the absence of such for the presence penalty (see Fig. 6). Furthermore, the GPT-2 detector shows similar spike patterns in the detection scores for the presence penalty, hinting at potential problems in the detector model's training data. No difference has been found between argumentative and research genres.

Following the linear searches, interactions between frequency and presence penalty were explored in the context of detection scores performing a grid search. The same pattern of frequency penalty decreasing detection scores is observed (see Fig. 7), while the effect of the presence penalty is minimal (except for the GPT-2 detector).

Confirming the results of the baseline experiments, frequency penalty is shown to have a drastic impact on detection with a tradeoff for text quality. However, the improvement in the detection industry is evident, as the frequency penalty threshold decreasing detection below

Fig. 4 Influence of the tone (with description) on the score of the different detectors



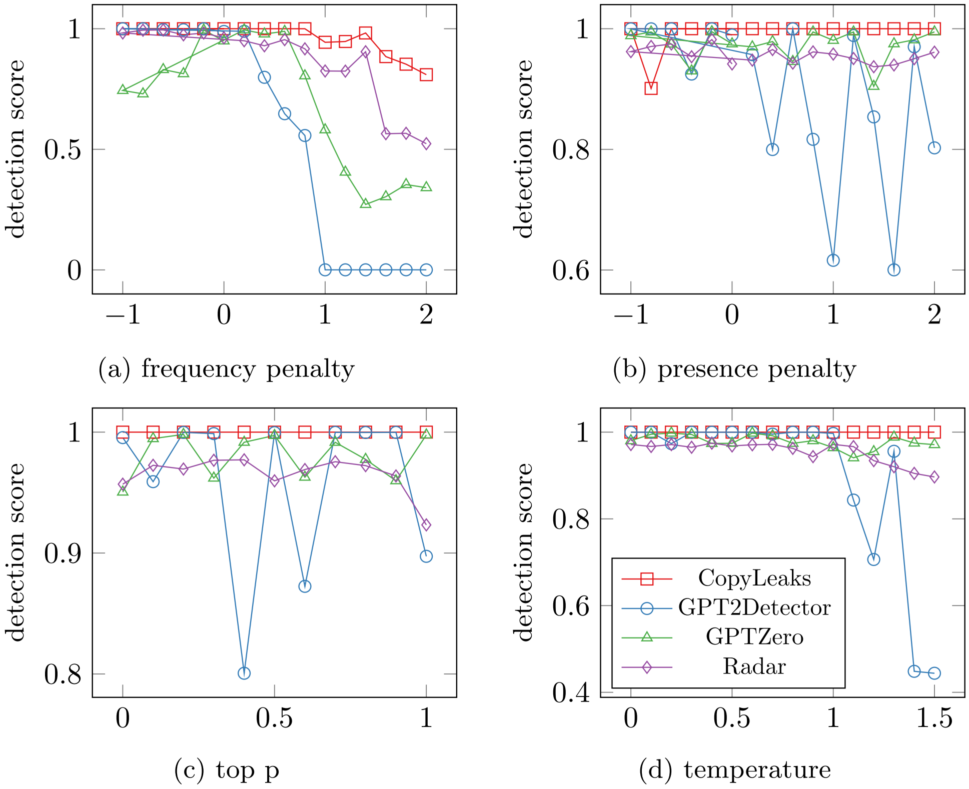
(a) CopyLeaks

(b) GPTZero

(c) Radar

(d) GPT-2 Output Detector

Fig. 5 Influence of different parameters on the detection score in the preliminary hyperparameter study



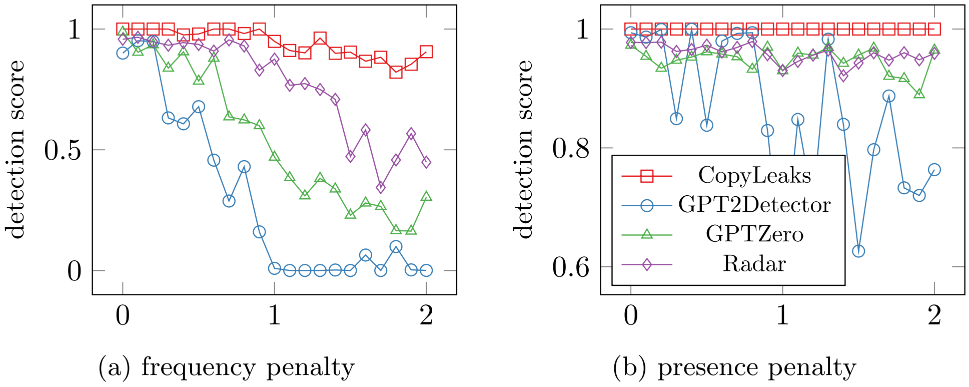
(a) frequency penalty

(b) presence penalty

(c) top p

(d) temperature

Fig. 6 Influence of different parameters on the detection score in the extended hyperparameter study



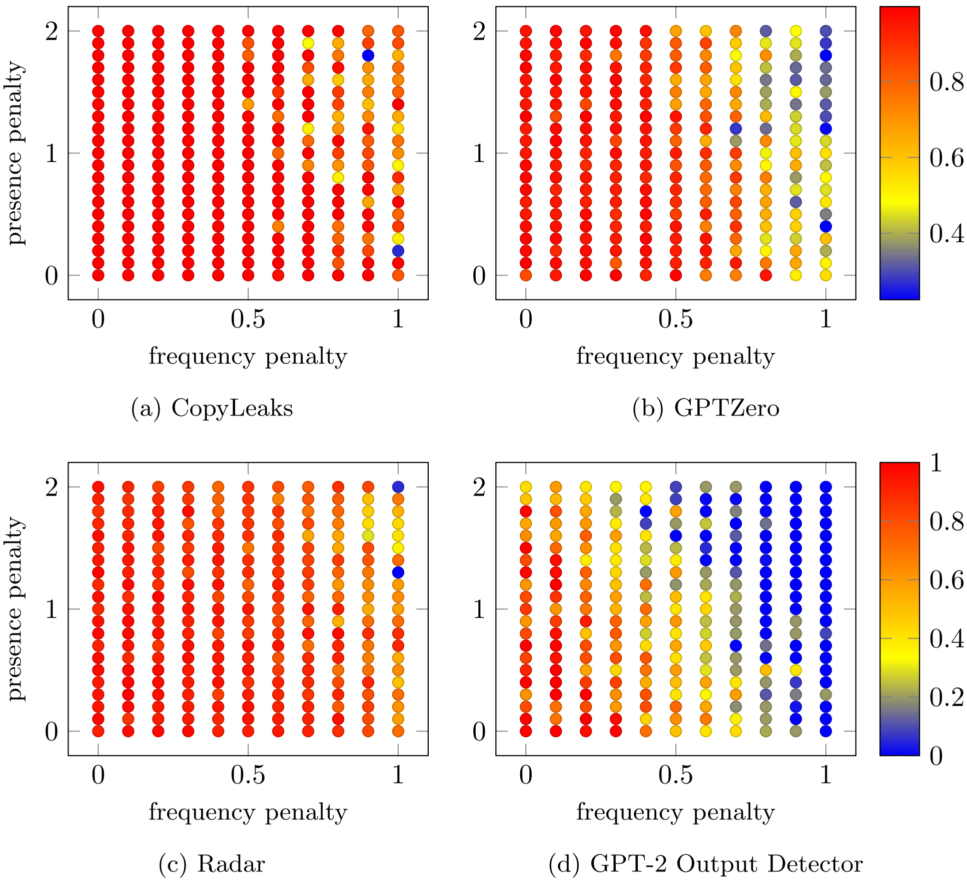
(a) frequency penalty

(b) presence penalty

0.5 is way higher (or non-existent) for the three new detectors. Consequently, the attack strategy found efficient in the baseline experiments can no longer be identified as such due to the detection-quality trade-off threshold increment. Only texts of deficient quality are classified below 0.5 at this stage, leading to the parameter tweaking attack being no longer viable. Additionally, the presence and frequency penalty showed a lower impact on the GPT-2 Output Detector detection scores than we found in the baseline experiments, which could be caused by differences in the generated texts or changes made in the snapshot of the GPT-3.5 model that was used.

4.4 Detection avoidance prompts

Different prompts aimed at avoiding detection were ineffective for all three new detectors. The GPT-2 Output Detector struggled with all forms of the second-query approach (see Fig. 8). Radar showed a slight but consistent drop in detection scores and higher variability for the second-query approach. Still, most Radar scores stayed above 0.6–0.7 for all the prompts. This experiment confirms the conclusion of previous work that the GPT-2 Output Detector struggles with second-query approach. Based on the experiment, we can conclude that the new detectors resist general prompt engineering attacks.



(a) CopyLeaks

(b) GPTZero

(c) Radar

(d) GPT-2 Output Detector

Fig. 7 Influence of joined optimisation of frequency and presence penalties on the detection score

4.5 Outlier prompts

This experiment simulates different characteristics found in outlier texts using prompt engineering. Overall, requesting a generation model to use in-text citations, reference lists, numerical data, tone blends, non-standard grammar structures, and first-person tone did not impact detection scores. GPTZero showed a slight decrease in detection scores when the reference list is included, which can be attributed to the reference list, as removing it spikes the detection back to the baseline. Overall, we did not observe any effect of simulating features of outliers on the detection scores.

4.6 Character mutations

In this experiment, we test the influence of various character-level mutations on detection accuracy. Copyleaks was found to be vulnerable to multiple character-level mutations (see Table 3). However, this only applies to their AI-detection API, and the attacks might be mitigated when using a full plagiarism-checking model, which features an explicit cheat-detection function. GPTZero resisted all of the attacks. However, the detection scores dropped to around 0.8 and showed higher variability for the English-Cyrillic full version, L-I, 19-char, and combined mutation methods. Radar resisted punctuation mutations but fell prone to the rest, with scores below 0.4 for each mutation. The GPT-2 Output Detector displayed high resistance to invisible characters and combined attacks while failing for the rest, with detection scores close to 0 for all remaining mutations.

Overall, we found the character-level mutation attacks to have varying results, with each detector having its weaknesses for a specific type of mutation. Notably, GPTZero is the only detector for which the scores never dropped below 0.5, successfully resisting all the character mutation attacks.

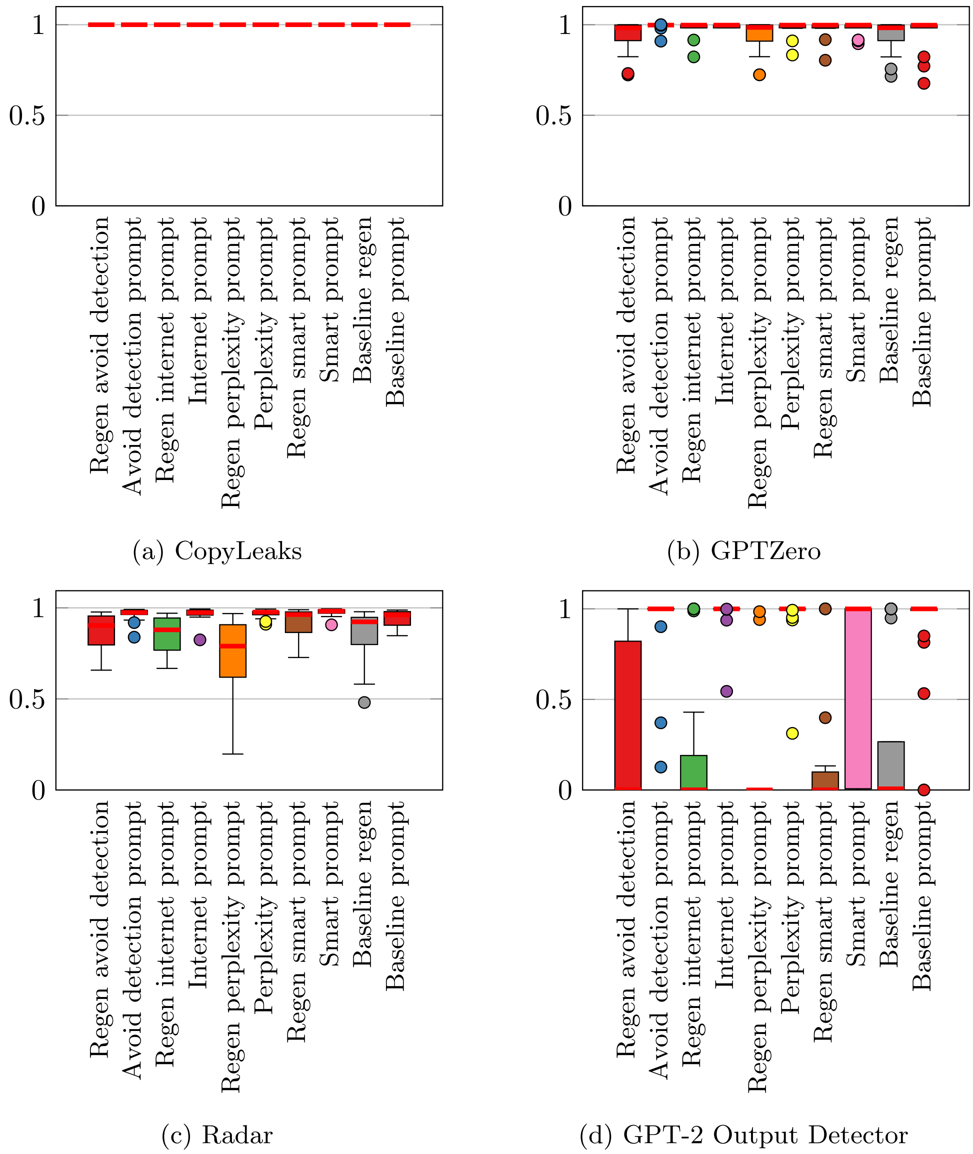
4.7 Translations

We found all detectors to be robust against intermediary-language translation attacks. Although GPTZero, Copyleaks and Radar showed slightly greater variability in detection scores, the influence was very limited. GPT-2 Output Detector scores dropped more significantly, showing high variability and medians below 0.5 for the Japanese and Russian translations.

4.8 Paraphrasing attack

Copyleaks and Radar proved to be resistant against paraphrasing attacks, with detection scores exhibiting higher

Fig. 8 Influence of prompt engineering on the score of the different detectors



(a) CopyLeaks

(b) GPTZero

(c) Radar

(d) GPT-2 Output Detector

variability but still staying above 0.5–0.7 (see Fig. 9). The GPTZero median and upper-bound scores dropped slightly below 0.5 for Quillbot paraphrasing, and the scores of the GPT-2 Output Detector plummeted to 0.

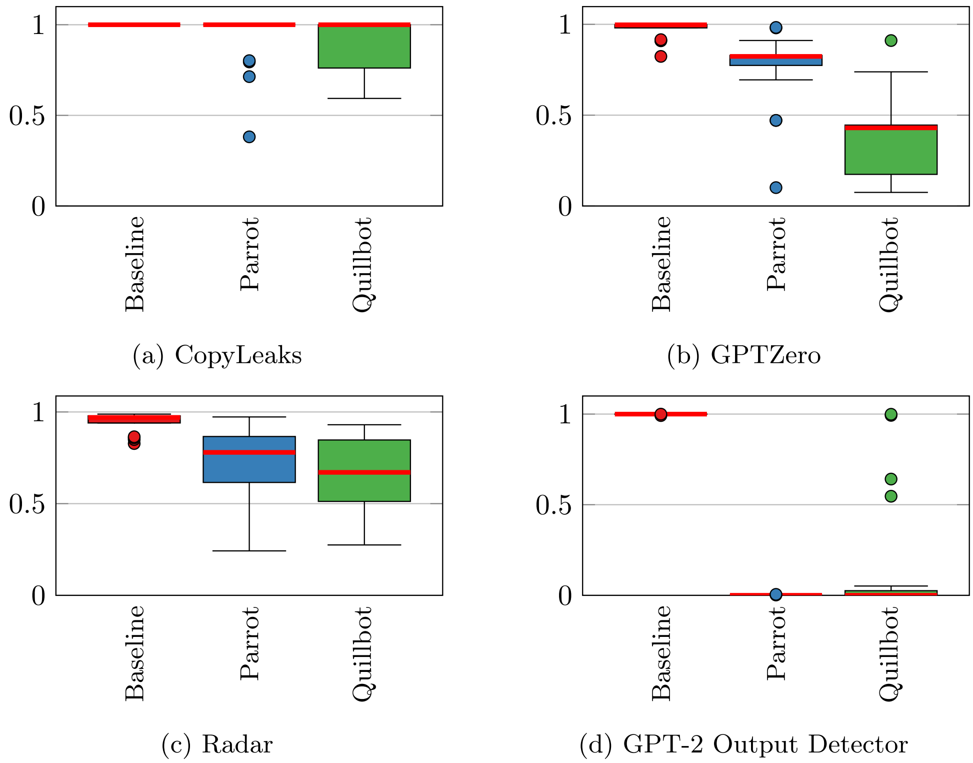
Table 3 Influence of character mutation attacks on the different detectors

Mutation	CopyLeaks	GPT2Detector	GPTZero	Radar
Armenian	0.0	0.34	0.97	0.37
Combined	0.0	0.95	0.79	0.36
Cyrillic-Simple	0.0	0.01	0.97	0.25
Cyrillic-Full	0.0	0.06	0.79	0.28
Baseline	1.0	0.91	0.97	0.94
Greek	0.0	0.12	0.97	0.14
Invisible	0.0	1.0	0.97	0.39
L-I	0.95	0.0	0.79	0.42
19-char	0.0	0.07	0.79	0.22
Punctuation	0.98	0.0	0.97	0.8
Various	0.62	0.0	0.97	0.18

5 Conclusion

This study explored the effectiveness of adversarial attacks on different neural text detectors, based on texts generated by GPT-3.5. While neural text detectors in the past have proven to be very vulnerable to such attacks, the results presented in this article paint a more nuanced picture. While, as of April 2024, no detector was completely invulnerable to adversarial attacks, the latest generation of commercial detectors, like Copyleaks and GPTZero, is in general very robust and only vulnerable to very specific

Fig. 9 Influence of paraphrasing on the score of the different detectors



attack strategies. Given that the capabilities of neural text detectors seem to improve rapidly, they might soon also be robust against these attacks. Table 4 shows an overview of the tested attack strategies and detectors and shows the success of the different approaches for the given detectors.

5.1 Ethical considerations

Neural text detectors in the past have shown to be biased towards non-native English writers, falsely flagging their work as AI generated (Liang et al., 2023d). Such false positives, which we did not investigate in our study, can potentially have severe consequences, e.g. in the context of education. We, therefore, advise to use the assessment of neural text detectors with caution.

5.2 Limitations

5.2.1 Scope

In this study, we explored the effect of adversarial attacks on the AI detection scores of four popular detectors. Consequently, this study only showcases the detectors' robustness against adversarial attacks. Due to the exclusion of human-written texts, it cannot be used to evaluate the detectors' general performance. As summarised in Sect. 2.1, a lot of previous work has been conducted on evaluating the accuracy of such detectors.

5.2.2 Reproducibility

This study examines actively developed generation and detection models GPT-3.5, Copyleaks, and GPTZero from

Table 4 High-level overview of the successfulness of the tested attack strategies on the different detectors (including results from Fishchuk and Braun (2023))

Detector	Prompt engineering	Hyperparameter-tweaking	Character mutation	Paraphrasing	Translation
Copyleaks	✗	✗	✓	✗	✗
GPTZero	✗	✗	✗	✓	✗
Radar	✗	✗	✓	✗	✗
GPT-2 Output Detector	✓	✓	✓	✓	✓
Turnitin	✗	✓	✓	n.a	n.a
OpenaAI Text Classifier	✗	✓	✓	n.a	n.a

January 2024 to April 2024. The behaviour of the models is subject to change with new updates, and the results of this study might become partially irreproducible. Therefore, we included the open source detectors GPT-2 and Radar as a baseline.

5.2.3 Texts

We focus on essay genre and text length within specific boundaries, which might not generalise to other text genres and lengths.

5.2.4 Copyleaks

We used the Copyleaks AI detection API as detector. However, Copyleaks also offers a wider plagiarism detection service, which includes AI detection. Theoretically, the wider service might catch character-level mutation attacks and flag them before initiating the AI detection model. However, this has not been tested and is beyond the scope of this study.

5.3 Future work

During this study, we considered using a one-way translation-based attack, where GPT-3.5 is asked to generate a text in a language different from English, and the result is translated back to English. However, due to our lack of expertise in the said languages and limited time, we left it out of the scope of the study. Additionally, the texts generated in languages other than English conflicted with the experiment setup in the context of word limit. However, a preliminary inspection of intermediary languages yielded promising results, laying the foundation for future research. Furthermore, this study can be further extended to additional generational models.

Acknowledgements The authors would like to thank (in alphabetical order): Copyleaks for providing us access and credits to their API. GPTZero for providing us access and credits to their API.

References

Akhtar, N., Mian, A., Kardan, N., & Shah, M. (2021). Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access, 9*, 155161–155196.

Akram, A. (2023). An empirical study of AI-generated text detection tools. *Advances in Machine Learning & Artificial Intelligence, 4*(2), 44–55.

Cooperman, S. R., & Brandão, R. A. (2024). Ai tools vs AI text: Detecting AI-generated writing in foot and ankle surgery. *Foot & Ankle Surgery: Techniques, Reports & Cases, 4*(1), 100367. https://doi.org/10.1016/j.fastrc.2024.100367

Crothers, E., Japkowicz, N., Viktor, H., & Branco, P. (2022). Adversarial robustness of neural-statistical features in detection of generative transformers. In *2022 international joint conference on neural networks (IJCNN)* (pp. 1–8).

Damodaran, P. (2021). *Parrot: Paraphrase generation for NLU.*

Ebrahimi, J., Rao, A., Lowd, D., & Dou, D. (2018, July). HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 31– 36). Association for Computational Linguistics. Retrieved from https://aclanthology.org/P18-2006

Elkhatat, A. M., Elsaid, K., & Almeer, S. (2023). Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. *International Journal for Educational Integrity, 19*(1), 17.

Emi, B., & Spero, M. (2024). Technical report on the Checkfor.ai AI-generated text classifier. *arXiv preprint* arXiv:2402.14873

Fishchuk, V., & Braun, D. (2023). Efficient black-box adversarial attacks on neural text detectors. In Abbas, M., & Freihat, A. A., (Eds.), *Proceedings of the 6th international conference on natural language and speech processing (ICNLSP 2023)* (pp. 78–83). Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.icnlsp-1.8

Gao, J., Lanchantin, J., Soffa, M. L., & Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *Proceedings—2018 IEEE symposium on security and privacy workshops. (SPW)* (pp. 50–56). https://doi.org/10.1109/SPW.2018.00016

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. In *3rd international conference on learning representations, ICLR 2015—conference track proceedings*.

Habibzadeh, F. (2023). Gptzero performance in identifying artificial intelligencegenerated medical texts: A preliminary study. *Journal of Korean Medical Science, 38*(38).

Hu, X., Chen, P.-Y., & Ho, T.-Y. (2023). Radar: Robust AI-text detection via adversarial learning. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (Vol. 36, pp. 15077–15095). Curran Associates Inc.

Jawahar, G., Abdul-Mageed, M., & Lakshmanan, L. V. S. (2020). Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th international conference on computational linguistics* (pp. 2296–2309). International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2020.coling-main.208

Jin, D., Jin, Z., Zhou, J.T., & Szolovits, P. (2019). Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *AAAI 2020—34th AAAI conference on artificial intelligence* (pp. 8018–8025). https://doi.org/10.1609/aaai.v34i05.6311

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). A watermark for large language models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., & Scarlett, J. (Eds.), *Proceedings of the 40th international conference on machine learning* (Vol. 202, pp. 17061–17084). PMLR. Retrieved from https://proceedings.mlr.press/v202/kirchenbauer23a.html

Kong, Z., Xue, J., Wang, Y., Huang, L., Niu, Z., & Li, F. (2021). A survey on adversarial attack in the age of artificial intelligence. *Wireless Communications and Mobile Computing, 2021*, 1–22.

Liang, G., Guerrero, J., & Alsmadi, I. (2023a). Mutation-based adversarial attacks on neural text detectors. *arXiv preprint* arXiv:2302.05794

Liang, G., Guerrero, J., Zheng, F., & Alsmadi, I. (2023b). Enhancing neural text detector robustness with μattacking and RR-training. *Electronics, 12*(8), 1948. https://doi.org/10.3390/electronics12081948

Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., & Zou, J. (2023c). GPT detectors are biased against non-native english writers. In *ICLR 2023 workshop on trustworthy and reliable large-scale machine learning models*.

Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., & Zou, J. (2023d). GPT detectors are biased against non-native English writers. *Patterns, 4*(7), 100779. https://doi.org/10.1016/j.patter.2023.100779

Nova, A. (2019). *Essay topics: 100+ best essay topics for your guidance.* Retrieved November 7, 2023, from https://www.5staressays.com/blog/essay-writing-guide/essay-topics

OpenAI. (2023). *Api reference— openai api.* Retrieved from https://platform.openai.com/docs/api-reference/chat/create

Peng, H., Wang, Z., Zhao, D., Wu, Y., Han, J., Guo, S., & Zhong, M. (2023). Efficient text-based evolution algorithm to hard-label adversarial attacks on text. *Journal of King Saud University-Computer and Information Sciences, 35*, 101539. https://doi.org/10.1016/J.JKSUCI.2023.03.017

Rathore, P., Basak, A., Nistala, S. H., & Runkana, V. (2020). Untargeted, targeted and universal adversarial attacks and defenses on time series. In *2020 international joint conference on neural networks (IJCNN)* (pp. 1–8).

Sadasivan, V.S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). *Can AI-generated text be reliably detected?*

Shen, L., Zhang, X., Ji, S., Pu, Y., Ge, C., Yang, X., & Feng, Y. (2023). *Textdefense: Adversarial text detection based on word importance entropy.*

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., & Wu, J. (2019). Release strategies and the social impacts of language models. *arXiv preprint* arXiv:1908.09203

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. In *2nd international conference on learning representations, (ICLR 2014) —Conference Track Proceedings.*

van Oijen, V. (2023). *AI-generated text detectors: Do they work?* Retrieved 9 March, 2024, from https://communities.surf.nl/en/ai-in-education/article/ai-generated-text-detectors-do-they-work

Walters, W. H. (2023). The effectiveness of software designed to detect AI-generated writing: A comparison of 16 AI text detectors. *Open Information Science, 7*(1), 20220158. Retrieved 20 April, 2024, from https://doi.org/10.1515/opis-2022-0158.

Weber-Wulff, D., Anohina-Naumeca, A., Bjelobaba, S., Foltỳnek, T., Guerrero-Dib, J., Popoola, O., & Waddington, L. (2023). Testing of detection tools for AI-generated text. *International Journal for Educational Integrity, 19*(1), 26.

Wolff, M., & Wolff, S. (2022). *Attacking neural text detectors.*

Xu, H., Ma, Y., Liu, H.-C., Deb, D., Liu, H., Tang, J.-L., & Jain, A. K. (2020). Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing, 17*, 151–178. https://doi.org/10.1007/s11633-019-1211-x

Yoo, K., Kim, J., Jang, J., & Kwak, N. (2022). Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the association for computational linguistics. (ACL 2022)* (pp. 3656–3672). Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.findings-acl.289

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.