Evading Detection: Deceiving Machine-Text Detectors through Realistic Red Teaming Attacks

Alex Sotiropoulos and **Jugal Wadhwa** and **Akshay Padte** and
Sathiya Murthi Sankaran and **Zheng Bao**
University of Southern California, Los Angeles
{asotirop, jwadhwa, padte, sathiyam, baojames}@usc.edu

Abstract

This project proposes a focused investigation into evading detection of machine-generated text by current zero-shot techniques, through realistic red teaming attacks. The study aims to assess the detection technique and quality of text that has been 'attacked' through paraphrasing and perturbations. By rigorously evaluating the red teaming methods, this study seeks to uncover the essential ability to maintain the quality of attacked text while successfully evading the detection of machine-detected text. Our research draws inspiration from Shi et al. (2023), who performed red-teaming against detection methods such as DetectGPT, watermarking techniques and classifier-based detectors. Building upon this, we extended the scope of red teaming to two additional zero-shot detection methods, FastDetectGPT and DNA GPT. Our work also highlights an overlooked aspect of the original red teaming paper, the quality of the attacked text, and proposes attack methods which take this concern into account.

1 Introduction

In the rapidly evolving landscape of Generative AI, Large Language Models (LLMs) are advancing at an unprecedented pace, demonstrating remarkable progress in quantity and quality (Brown et al., 2020; OpenAI, 2022; Chowdhery et al., 2022; Zhang et al., 2022). These LLMs have become critical components of applications spanning various domains. However, with their widespread adoption comes the pressing need for tools capable of discerning and validating the origins of content, particularly in combating misinformation and ensuring security.

Against this backdrop, adopting zero-shot detection methods emerges as a pivotal strategy for determining the machine-generated nature of passages. There is an emphasis on zero-shot techniques due to their inherent advantage of not requiring millions of labeled samples required by other supervised techniques, especially with the number of currently available and upcoming LLMs. Recent zero-shot techniques such as DetectGPT show promising performance even in new text domains (Mitchell et al., 2023).

Despite the advancements in zero-shot techniques in the machine-text detection landscape, these methods often struggle with texts that have been deliberately altered to evade detection, revealing significant vulnerabilities in the current technology. Our research is motivated by the need to address these vulnerabilities through realistic red teaming attacks that not only test the robustness of these detectors but also adhere to a strict criterion: attacks must preserve the meaning and flow of the original text as much as possible, which we refer to as "quality attacks." Through our newly proposed red teaming attack methods, we wish to demonstrate the need for more resilient machine-text detection methods, following an approach similar to Shi et al. (2023) in their testing of DetectGPT.

2 Literature Review

Humans find it hard to distinguish between LLM-generated text and human-written text (Clark et al., 2021). In light of this, there have been several approaches to solve this problem, with the majority treating it as a binary classification problem. Initial solutions employed supervised learning techniques to train detection models. (Zellers et al., 2019; Solaiman et al., 2019; Ippolito et al., 2020). For instance, the GROVER model, developed by Zellers et al. (2019) can generate complete news articles from just a headline, and is also an efficient detector of its articles with a 92% accuracy. However, the downside is evident: models like GROVER are susceptible to misuse and tend to overfit their training data, resulting in less effective detection of machine-generated text (Bakhtin et al., 2019).

Another set of approaches uses zero-shot methods to perform the classification. Most notably, DetectGPT leverages the observation that machine-generated text often resides in areas where the log probability function has negative curvature (Mitchell et al., 2023). This paper paved the way for subsequent innovations, including FastDetectGPT (Bao et al., 2024), which provides an enhancement for DetectGPT's run time, and DNA-GPT (Yang et al., 2023), a detection method that utilizes the unique text continuation patterns distinguishing human from AI-generated content, based solely on the model's textual inputs and outputs.

Despite the advances in this area, these tools face a significant performance degradation if the LLM-generated text is modified (e.g. by word substitutions, paraphrasing, etc.) before passing it to the detector (Shi et al., 2023). Furthermore, Wolff (2020) demonstrates that replacing characters with homoglyphs and intentionally misspelling words can reduce detection accuracy by as much as ~76%. Thus, current detection methods need to be improved in terms of robustness to changes and attacks. Although watermarking and retrieval-based approaches show greater resilience to attacks like rephrasing or substitution (Shi et al., 2023; Krishna et al., 2023)—experiencing less detection accuracy degradation—these strategies rely heavily on good-faith modifications from the developers of LLMs. This creates a vulnerability, as malicious users might use LLMs without watermarking protections. Therefore, our work focuses exclusively on zero-shot detection methods.

3 Problem Description

Shi et al. (2023) effectively demonstrated that DetectGPT's detection performance significantly deteriorates through word-substitution using lowest negative log probability attacks. However, this strategy does not take into account the quality of these attacked passages. Moreover, an attack that neglects efforts in maintaining passage quality fails to reflect a realistic scenario, as an adversary typically aims to evade machine-text detection and preserve the readability and clarity necessary for human interpretation. Below is an example of an attack from the work done by Shi et al. (2023). The attacks here clearly result in a loss of meaning and impact the overall flow of the original passage. For example, while some substitutions like changing "attack" to "onslaught" are viewed, in our eyes, as quality attacks, others significantly distort the text. Replacing "white power" with "fair power" alters the context, and changing "which left" to "which went away" disrupts the sentence flow. This example outlines the downside of only considering the lowest negative log probability word substitutions.

LLM-GENERATED, Unmodified:

> [...] was singled out by Zack Davies who was heard saying "white power" [...]. Trevor Jones was also jailed for 19 years over the attack at the store which left a 60-year-old employee [...].

QUERY-BASED ATTACK:

> [...] was singled out by Ethan Davies who was Registered saying "Fair power" [...]. Trevor Jones was also jailed for 19 years over the Onslaught at the store which went away a golden ager employee [...].

Our goal is to show that, by applying a minimum quality threshold to attacked text excerpts, detectors can still be compromised while maintaining the passage quality. Additionally, we extend these attacks to include DetectGPT, as well as two other zero-shot methods, FastDetectGPT and DNA-GPT. We selected these two zero-shot methods for our analysis because they both demonstrate improved detection performance over DetectGPT. Furthermore, their code bases were openly available on GitHub and aligned with our computing resource constraints, enabling practical experimentation given our limited resources.

4 Methods

1. Obtain 100 samples of paragraphs of 4-5 sentences as the machine generated text by using the first 30 tokens of the samples from the xsum dataset as the prompt to the base model

2. Apply the following attack methods to generate 100 samples for each attack method.

   (a) *Random substitution*: Replace random words with similar meaning or synonyms. This method replaces at most 20% of the words excluding stop words.

   (b) *Random substitution with threshold*: Only allow the the log probability score drop by a maximum of 0.2 using the third party LLM. Similarly re-do with threshold 0.4

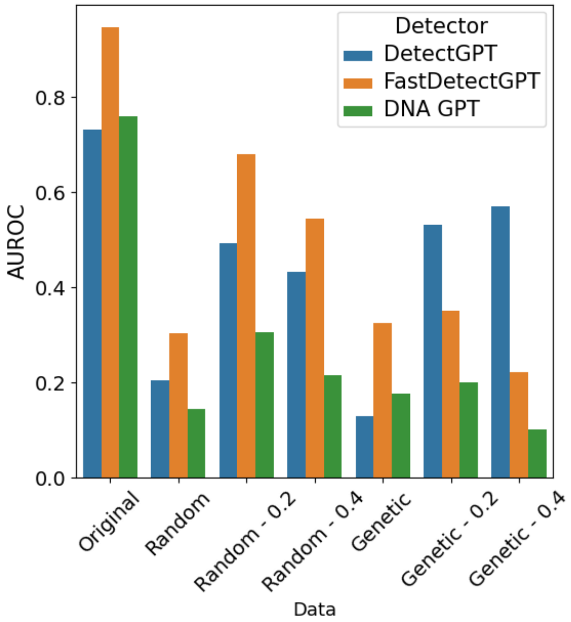


Figure 1: Comparison of AUROC on different attack methods for each detector

(c) *Genetic algorithm based replacement*: Choose the words to be replaced using a genetic algorithm. Passages with lowest log probability under the base model are considered best candidates for the generation, and are cloned and mutated to create the next generation, and so on for 10 generations.

(d) *Genetic algorithm based replacement with threshold*: Choose the words to be replaced using a genetic algorithm. Passages with lowest FastDetectGPT scores (FastDetectGPT classifies them closer to human) are considered best candidates for the generation, and are cloned and mutated to create the next generation, and so on for 10 generations. Threshold the drop in log probability under a third party LLM to 0.2. Also re-do with threshold 0.4

3. Using the original and 6 set(Random with two thresholds x3 and Genetic with two thresholds x3) of attacked text, it is run against the detectors DetectGPT, FastDetectGPT and GPT

4. Upon running the detection AUROC scores are calculated based on the detection results

5 Experimental Results

1. Set up

1. Dataset: 100 samples from the xsum dataset (considered as human-written), and 100 passages generated by base model from those samples using the first few sentences as prompt

2. Base model to generate LLM text: GPT2-xl

3. Detection Model: GPT2-xl

4. Third party LLM for Log Probability scores: babbage-002

5. Detectors

   (a) DetectGPT

   (b) FastDetectGPT

   (c) DNA GPT

2. Evaluation Protocol

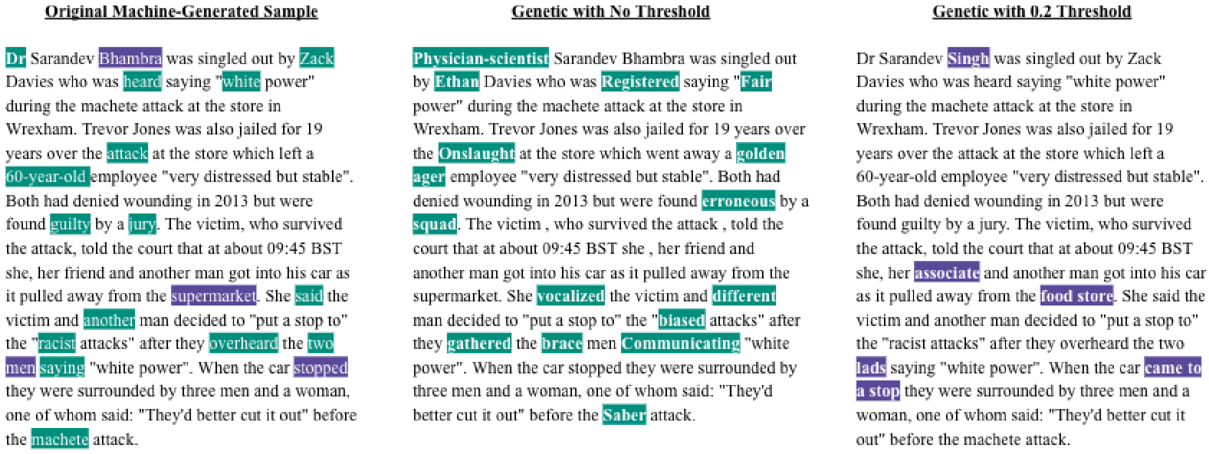
Upon reviewing the attacking methods used in the original paper, the text obtained has been altered contextually when viewed by a human. Thus there is a need for quality analysis which is implemented using the log probability scores. Upon applying the thresholds of 0.2 and 0.4, human evaluation is possible to view the number of replacements and context of the passage to verify the quality. The AUROC scores are obtained for each detection method to compare the performance of detection and verify that the attack was successful against all of them despite trying to maintain quality.

3. Results and discussion

Reviewing the AUROC scores obtained as per Table 1, it is clear that the detection techniques fail in both cases of random substitution and genetic algorithm replacement, even after the application of thresholds to the log probability scores. From Figure 1, it is noticeable that there is a trend of dropping AUROC scores for each of the attacking methods. Amongst thresholded attacks, the most drastic drop is with genetic algorithm against FastDetectGPT, which comes as no surprise as the goal of that attack method focused on beating FastDetectGPT. We see from samples obtained the quality is preserved and an example of this is in Figure 2. However, it's important to note that some minor deterioration in quality still exists, as demonstrated by the substitution from "Bhambra" to "Singh" shown in Figure 2. To address this issue, in future work,

Method	DetectGPT	Fast DetectGPT	DNA GPT
Original	0.73	0.9466	0.7582
Random	0.2036	0.3033	0.1435
Random - 0.2	0.4919	0.6795	0.3052
Random - 0.4	0.4315	0.5437	0.2142
Genetic	0.1292	0.3233	0.1754
Genetic - 0.2	0.53	0.3500	0.2000
Genetic - 0.4	0.57	0.2200	0.1000

Table 1: Comparison of AUROC on different attack methods for each detector



Original Machine-Generated Sample

Dr Sarandev Bhambra was singled out by Zack Davies who was heard saying "white power" during the machete attack at the store in Wrexham. Trevor Jones was also jailed for 19 years over the attack at the store which left a 60-year-old employee "very distressed but stable". Both had denied wounding in 2013 but were found guilty by a jury. The victim, who survived the attack, told the court that at about 09:45 BST she, her friend and another man got into his car as it pulled away from the supermarket. She said the victim and another man decided to "put a stop to" the "racist attacks" after they overheard the two men saying "white power". When the car stopped they were surrounded by three men and a woman, one of whom said: "They'd better cut it out" before the machete attack.

Genetic with No Threshold

Physician-scientist Sarandev Bhambra was singled out by Ethan Davies who was Registered saying "Fair power" during the machete attack at the store in Wrexham. Trevor Jones was also jailed for 19 years over the Onslaught at the store which went away a golden ager employee "very distressed but stable". Both had denied wounding in 2013 but were found erroneous by a squad. The victim , who survived the attack , told the court that at about 09:45 BST she , her friend and another man got into his car as it pulled away from the supermarket. She vocalized the victim and different man decided to "put a stop to" the "biased attacks" after they gathered the brace men Communicating "white power". When the car stopped they were surrounded by three men and a woman, one of whom said: "They'd better cut it out" before the Saber attack.

Genetic with 0.2 Threshold

Dr Sarandev Singh was singled out by Zack Davies who was heard saying "white power" during the machete attack at the store in Wrexham. Trevor Jones was also jailed for 19 years over the attack at the store which left a 60-year-old employee "very distressed but stable". Both had denied wounding in 2013 but were found guilty by a jury. The victim, who survived the attack, told the court that at about 09:45 BST she, her associate and another man got into his car as it pulled away from the food store. She said the victim and another man decided to "put a stop to" the "racist attacks" after they overheard the two lads saying "white power". When the car came to a stop they were surrounded by three men and a woman, one of whom said: "They'd better cut it out" before the machete attack.

Figure 2: Sample of Genetic replacement without threshold and 0.2 drop threshold

we can improve our detection technique by applying Named Entity Recognition (NER) to preserve named entities. Yet it can be observed that the overall preservation of quality was significantly better than that in the genetic attack without thresholding. This example illustrates that these methods successfully evade detection and fulfill the study's objective of preserving the quality of attacked passages.

6 Conclusions and Future Work

While Shi et al. (2023) demonstrated that attacking text samples with word substitutions can degrade the AUROC of zero-shot detection techniques like DetectGPT, their analysis does not address the accompanying drop in textual quality, which contradicts the purpose of employing machine-generated text. Such attacks typically result in a decline in quality, potential changes in tone, context conflicts, and a loss of coherence, undermining the effectiveness of red teaming detection models. A more realistic approach to red teaming would involve modifying text samples through word substitutions while ensuring the quality remains high. Our work builds on these insights, highlighting the balance between preserving text quality and evading detection. We found that employing both Random and Genetic methods at thresholds of 0.2 and 0.4 results in a substantial decrease in detection rates, although less drastically than when these attacks are applied without any thresholds. Nevertheless, these results illustrate that it is possible to hinder detection capabilities while simultaneously preserving the quality of the text.

Looking ahead, potential future work in this area could focus on refining attack methods, such as those utilizing genetic algorithms, which currently target defeating FastDetectGPT but make them excel against other detectors as well (perhaps by modifying the selection criterion in the genetic algorithm to consider scores from multiple detectors). The goal would be to develop optimizations that can effectively challenge all detectors using the same method. Additionally, understanding the quality and context of the text remains predominantly a manual process, so proposing concrete metrics for quality measurements would further justify our approach. Finally, testing these attack methods on text data from diverse domain-specific datasets could greatly improve the relevance and thoroughness of the results.

7 Individual Contributions

- Akshay: Went through the code for the paper by Shi et al. (2023) and ran it with different parameters. Raised the issues about the quality of the passages of query-based attack method. Implemented the modifications for applying threshold by the genetic algorithm based on log probabilities of the babbage LLM, and changed the selection criterion of the genetic algorithm to FastDetectGPT scores.

- Alex: Created project repository. Reviewed code from the original Red Teaming paper (Shi et al., 2023) and attempted to run code locally. Composed introduction email to our TA advisor, and also attended zoom session to receive feedback on our project results. Setup and created the skeleton layout for our final presentation, and was one of the three presenters for the group.

- Jugal: Read the code for DetectGPT attacking. Gained a basic understanding of a the code flow in regards to detection in both techniques. Tried to gain an understanding of the resources needed in regards to the execution and look at possible providers for compute resources. Evaluated colab as a viable runtime for running detection algorithms to generate auroc scores. Wrote scripts to match the input format for the attacked text to match the expected format of detectors. Reviewed the code for FastDetectGPT and DNA GPT and set up a runtime on colab to test the model. Generated the input files and ran detection on all three models to obtain scores for the original and attacked text (6 sets) for each of the 3 detectors.

- Sathiya: Reviewed the code for the Fast DetectGPT method. Set up a local environment, implemented and tested the model. Evaluated the non-attacked and attacked samples and checked performance. Explored options available to determine the content quality and readability.

- Zheng: Studied red-teaming techniques. Explored automatic and human evaluation methods and pipelines for assessing model-generated texts. Ran the genetic algorithm script and generated the attacked text without threshold.

References

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

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online. Association for Computational Linguistics.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *arXiv preprint arXiv:2303.13408*.

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

OpenAI. 2022. CHATGPT: Optimizing language models for dialogue. [Accessed on 2024-02-29].

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2023. Red teaming language model detectors with language models.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Max Wolff. 2020. Attacking neural text detectors. *CoRR*, abs/2002.11768.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dnagpt: Divergent n-gram analysis for training-free detection of gpt-generated text.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news.

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. 2022. Opt: Open pretrained transformer language models.