Attacking to Enhance: Refining Machine-Text Detection through Vulnerability Analysis

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 the discriminative power and robustness of current zero-shot, few-shot, and supervised techniques for detecting machine-generated text. Emphasizing zero-shot methodologies, the study aims to assess performance across varying passage lengths and dataset domains, notably against samples of machine-generated text that have been purposely "attacked" with sample perturbations, and paraphrasing techniques. By rigorously evaluating these methods, this study seeks to uncover crucial weaknesses for enhancing the reliability of text detection systems. Doing so is critical in combating the proliferation of machine-generated content and ensuring the integrity of digital information. 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 wish to extend the scope of red teaming to additional zero-shot detection methods and explore performance across different domains. The study also intends to identify potential improvements to current techniques and determine the optimal method for the task.

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. This work aims to study techniques to detect this across zero-shot, few-shot, and supervised methods. 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)

With the rise in LLMs and detection techniques, it is necessary to analyze how the proposed methods perform on different LLMs and test their robustness to "attacks" such as noise, paraphrasing, and domain changes. This work aims to use this critical analysis to compare available techniques, determine the optimal approach, and potentially build on them to check the possibility of an enhanced ensemble approach.

Ultimately, this study aims to bolster our understanding of text detection methodologies and their efficacy in combating the proliferation of machine-generated content, thereby upholding the integrity and authenticity of digital information.

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 Detect-CodeGPT (Shi et al., 2024), which adapts DetectGPT's methodology for identifying machine-generated code, and Binoculars (Hans et al., 2024), a detection method that utilizes LLM observations, different from DetectGPT–log perplexity and next-token prediction perplexity.

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 leaves a vulnerability, as malicious users can circumvent these measures by developing or utilizing LLMs without watermarking protections.

3 Plan of Action

1. DetectGPT Implementation - Implement and replicate the results of detection using the DetectGPT model.

2. Research different zero shot methods - Review different zero shot models for detection of machine generated text.

3. Implement zero-shot models - Implement the different zero shot models, including newer ones not tested by Shi et al. (2023).

4. Comparative study of methods - Perform analysis of detection on possible domains such as

Milestones	**EDC**
1. Clone and Run "LLM-Detector-Robustness" Repo	March 11
2. Add other detection methods to code	March 17
3. Evaluate other methods in presences of adversarial attacks	March 25
4. Submit Status Report	March 26
5. Run methods against other dataset domains	March 31
6. Devise potential improvements for best performing detector	April 5
7. Assess potential improvements	April 15
8. Finish Final Report	April 20
9. Finish Final Presentation Slides	April 22

Table 1: **Milestones and Estimated Date of Completion (EDC)**

   (a) Fake News detection - WMT16 German News Dataset [1]

   (b) Social Media Deepfakes - TweepFake Dataset [2]

   (c) Plagiarism - Text and Code - PubMedQA Medical text dataset [3]

5. Analysis of different methods - Understand the results of the different methods based on reasoning performance and if any failure cases exist and why.

6. Test robustness to passage length, perturbation / paraphrasing, domain - Analysis on whether different parameters effect the detection of text.

7. Improve detectors and prevent jail breaking - Improve detection to work on text that has been written to bypass current detectors.

8. Ensemble detection method - Study an ensemble method of detection to understand if there's any improvement in performance

[1] WMT16 - https://www.statmt.org/wmt16/translation-task.html

[2] TweepFake - https://www.kaggle.com/datasets/mtesconi/twitter-deep-fake-text

[3] PubMedQA - https://pubmedqa.github.io/

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.

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.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text.

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].

Yuling Shi, Hongyu Zhang, Chengcheng Wan, and Xiaodong Gu. 2024. Between lines of code: Unraveling the distinct patterns of machine and human programmers.

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.

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 pre-trained transformer language models.