Watermarking of AI-Generated Text: Word Replacement Through Prompt Engineering

CS4991 Capstone Report, 2025

Maxwell Penders
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
pye3yh@virginia.edu

ABSTRACT

On social media, AI-generated text has been a dangerous force in the rise of disinformation campaigns, leading to a large quantity of bots which utilize LLM technology to generate false information en masse. I show a novel approach to LLM watermarking which may be used by LLM providers or models to encode AI detectability into the output text while focusing on a low false-positive rate to minimize the number of users that may be negatively impacted by this technology. Using simple word replacement through prompt engineering, I shift semantically similar words (synonyms) to reach a known AI encoding. Then, analysis of the text can use encoding signatures to determine a likelihood of a text being AI-generated. This model can detect AI-encoded words with a 77.35% accuracy rate, while having a low false-positive rate of at most 0.15%. Future work is needed to further refine this model and implement combined watermarking and watermark-probabilistic detection of AI-generated text to be sufficiently useful for wide-scale implementation.

1. INTRODUCTION

Few technologies have impacted the quantity of text output quite like large language models (LLMs). As more people around the world rush to incorporate this technology into their lives, more nefarious actors have harnessed easy access to this technology to generate unverified or purposely misleading claims and publish them to the internet. This text can be highly specialized and persuasive and has been found to be a common method for covert propaganda operations by foreign state-backed organizations [1].

In response to this growing issue, different organizations and individuals have released AI text detection algorithms with varying degrees of success, with a common theme being the failure of the available detection algorithms to obtain low false-positive rates, with OpenAI's proprietary classifier having a false positive rate of 9% [8].

Rather than focusing on text classification, recent work has been done on text watermarking, which is the process of hiding digital information within a piece of text. Many of these embeddings are done after the text is generated, commonly swapping semantically similar words to create a token.

The use case for this technology within the LLM space is to incorporate the use of these watermarks into the text generation pipelines of popular LLM APIs such as OpenAI's ChatGPT. I use a method similar to previous work in this field but include the encoding into the prompt itself, allowing the LLM to encode the text as it is generated.

2. RELATED WORKS

The study of text watermarking methods began in 1993 with the seminal work of Maxemchuk and Low [4] who described slightly shifting or marking text through changing spaces, shifts in line position, and individual character movements to encode an ID into the text. They expressed frustration at the ease with which such changes will likely either be easily lost in transfers like photocopying or are simply far too easily detected.

Further relevant work instead focused on the text itself, using aspects of the text such as entropy [5], natural-language semantic structures [7], and word length [2]. From these, the work done on natural-language semantic structures has led to recent work highly relevant to my research.

Recent work utilizes the same technologies empowering the LLMs themselves, generally centered around single/multiple word synonym substitution [6] and entropy [3]. The single/multiple word synonym substitution builds directly on the work from above and is the basis for the encoding in this paper.

3. PROJECT DESIGN

To create a prompt-loaded encoding for LLM output, an encoding system needs to be designed and specified. This encoding system requires a stored list of "signatures" which are used to recognize the watermarked text, as well as the prompt text used to encode the information. From there, a decoder must be created to determine if a specific input text has a text signature matching one from the stored list. Finally, the accuracy of this method is tested using watermarked and non-watermarked LLM-generated text.

3.1 System Design Resources

To generate the text for the watermarking system tests, I used the ChatGPT gpt-3.5-turbo model. I used a Claude 3.5 Sonnet model to aid in the writing of the code for the system as well as the code used to test and analyze the system. All code can be found at: https://github.com/User1391/Text-Watermarking

3.2 Prompt Encoding

I created a dictionary of 29 base words and at least 10 synonym words for each. For example, for the base word "mitigate", the synonym list was "reduce, minimize, alleviate, diminish, lessen, moderate, ease, soften, temper, control, contain, limit, decrease, curtail, manage". From there, I created a list of signatures, with each one using a different encoding of words. For example, an encoding of ["quick", "smart", "happy", "slow"] with bits [1, 3, 0, 2] is the signature one encoding. Then, categories are described for signatures based on how the chosen encoded words tend to influence the meaning of the output text. The four categories I use are *emotional*, *technical*, *critical*, and *neutral*. The signature used for the text is chosen based on keywords found in the prompt input text.

From there, the prompt is created. First, the instruction is given: "You will produce a short text on a topic, embedding a hidden watermark. To do this, you MUST replace certain words with specific synonyms. Here are the rules." Then, using the synonym mappings from the selected signature, the following is added to the prompt for each *word* and respective *forced_synonym* and corresponding *bit*: "Whenever you want to use '*word,*' use '*forced_synonym,*' (bit=*bit*)". Then, using the intended prompt *topic*, "Important: use each forced synonym at least once so the watermark is embedded. If you mention the same concept again, keep using the same forced synonym. Ensure coherence. TOPIC: *topic*. Now produce your watermarked text."

3.3 Output Text Decoding

To decode text output from the LLM, we first convert all of the text into lowercase, as that is how all synonyms are stored in the list. For each type of signature, it attempts to find the relevant synonyms and decode them into an output list. If this list matches one of the signatures, the text is considered to be a match for the watermark. If this list does not match one of the signatures, the text is considered to not be a match. If some but not all synonyms are found, it is recognized as a partial match but does not positively mark the text as a match.

3.4 Testing

Four test categories with five topics each were submitted to the prompt, multiple times with the text encoding requirements prompt header and multiple times without for a total of 4000 total LLM requests. One category, for example, was *technical*, and included the topics of [“artificial intelligence advances”, “quantum computing research”, “blockchain technology”, “machine learning algorithms”, “software engineering practices”].

4. RESULTS

Over the 2000 watermarked texts, watermarked text was detected with an accuracy of 77.35% ± 1.83% (with a 95% confidence interval). This is significantly less than the >90% accuracy rate obtained by other models. However, there were no false positives over the sample of 2000 non-watermarked texts, and using the “rule of threes”, we can say that with a 95% confidence interval the likelihood of a false positive is <~0.15% [9]. This is far superior to the aforementioned ChatGPT false positive rate of 9%.

Detection accuracy was also dependent on category. The *critical* and *technical* categories far exceeded detection rates of 80%, while the emotional category had a detection rate of only 61%.

5. CONCLUSION

These findings show the potential of prompt engineering as a method for easily watermarking LLM output. This work should help contribute to the body of work around text watermarking, reducing the harm done by LLMs by allowing social media and other platforms to easily identify whether a body of uploaded text was LLM-generated or not with a greater accuracy than traditional AI detectors.

6. FUTURE WORK

This work encompasses only a small step in achieving the goal of large-scale text watermarking for LLM APIs. Future work is required both in algorithmic encoding advancements as well as the implementation of a full-scale prompt engineering system with thousands of encodings and signatures. Finally, more research into combatting disinformation, especially that spread by AI, would aid in finding alternative and symbiotic technical and non-technical solutions to the issue.

REFERENCES

[1] Josh A Goldstein, Jason Chao, Shelby Grossman, Alex Stamos, and Michael Tomz. 2024. How persuasive is AI-generated propaganda? *PNAS Nexus* 3, 2 (February 2024), 1–7. https://doi.org/10.1093/pnasnexus/pgae034

[2] Zunera Jalil, Anwar M. Mirza, and Hajira Jabeen. 2010. Word length based zero-watermarking algorithm for tamper detection in text documents. *2010 2nd International Conference on Computer Engineering and Technology* (2010), V6-378-V6-382. https://doi.org/10.1109/ICCET.2010.5486185

[3] Zhuang Li. 2025. BiMarker: Enhancing Text Watermark Detection for Large

Language Models with Bipolar Watermarks. https://doi.org/10.48550/arXiv.2501.12174

[4] N.F. Maxemchuk and S. Low. 1997. Marking text documents. In *Proceedings of International Conference on Image Processing*, October 1997. 13 vols.3-. https://doi.org/10.1109/ICIP.1997.631958

[5] Yingjie Meng, Tao Guo, Zhihua Guo, and Liming Gao. 2010. Chinese Text Zero-Watermark Based on Sentence's Entropy. In *2010 International Conference on Multimedia Technology*, October 2010. 1–4. https://doi.org/10.1109/ICMULT.2010.5631421

[6] Travis Munyer, Abdullah Tanvir, Arjon Das, and Xin Zhong. 2024. DeepTextMark: A Deep Learning-Driven Text Watermarking Approach for Identifying Large Language Model Generated Text. https://doi.org/10.48550/arXiv.2305.05773

[7] O. Vybornova and B. Macq. 2007. A method of text watermarking using presuppositions. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, February 27, 2007. SPIE, 613–622. https://doi.org/10.1117/12.702871

[8] 2024. New AI classifier for indicating AI-written text. Retrieved February 13, 2025 from https://openai.com/index/new-ai-classifier-for-indicating-ai-written-text/

[9] 16.9.4 Confidence intervals when no events are observed. Retrieved March 18, 2025 from https://handbook-5-1.cochrane.org/chapter_16/16_9_4_confidence_intervals_when_no_events_are_observed.htm