香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Signature of the Future: Securing Large Language Models with Invisible Watermarking

FANG Zicheng 121090122 SHI Zhan 121090476
ZHENG Yingqi 121090841 XU Bowen 121090661

Abstract

As large language models (LLMs) and generative AI increasingly permeate various sectors, text traceability becomes paramount. The project introduces a novel watermarking algorithm designed to embed invisible markers in generated texts, making them detectable by specific algorithms without affecting readability or meaning. The algorithm has been validated for its high detection accuracy (99.3%) and low false positive rates (1.6%). Furthermore, empirical results emphasize the algorithm's resilience to attacks and its potential for wide applicability without requiring extensive retraining. The report concludes by addressing the ethical implications of deploying such technologies and the ongoing need for improvements to enhance watermark humanization while preserving detection capabilities.

1 Introduction

The proliferation of LLMs and generative AI has gradually entered various domains. However, its ability to quickly generate text also presents challenges in terms of information dissemination and intellectual property. There is a growing concern about the authenticity and integrity of generated content (Bergman et al., 2022; Mirsky et al., 2023). Simultaneously, the distinctive features of LLMs pose challenges in traditional watermarking, such as minimizing the impact on text quality, ensuring robustness against rewriting, and maintaining detection reliability across diverse applications (Radford et al., 2022). Consequently, AI model watermarking technology has emerged, which can embed watermark information to achieve content tracking and source attribution without affecting the performance of the AI model. This project aims to develop a robust watermarking solution specifically tailored for LLM-generated content.

2 Methodology

2.1 Algorithm

The core algorithms idea is to divide the LLM vocabulary into several parts using a hash value as a random seed. Words and word fragments are selected from one part as watermarks, achieving invisible watermarks and detectable content. Watermarks are detected by z-score test. If the z-score exceeds a certain threshold, the paragraph is considered to contain watermarks. The threshold is set at z = 4 , resulting in a low false positive rate of 3×10^{-5}. Since the probability of the chosen LLM cannot be changed to directly influence the next sample, the logits processor is used to adjust the scores of each word by a strength parameter δ and the activation function with 3 potential methods.

Preprint. Under review.

Algorithm 1: Watermark (3 methods)

Input: chosen language model and tokenizer M, corresponding vocabulary V, green list size $\gamma \in (0, 1)$, watermark strength δ, random number generator F if necessary.
1. Use the hash key of the input to generate a watermark key k to make sure fully randomness. (or we can use F directly)
2. Use watermark key to partition the vocabulary V into a "green list" $G \subset V$ of size $\gamma|V|$, and a "red list" $R = G^c$.
3. Each score ratio and the sum of scores represents the probability of word. A new language model $\hat{M}$ is defined such that for t and any prefix $[x, y_{1:t-1}]$, the resulting logits satisfy:

$$simple : \hat{\ell}_t[v] := \ell_t[v] + \delta \mathbf{1}(v \in G),$$

$$softmax : \hat{\ell}_t[v] := exp(\ell_t[v] + \delta \mathbf{1}(v \in G)),$$

$$swish : \hat{\ell}_t[v] := (\ell_t[v] + \delta \mathbf{1}(v \in G)) * sigmoid(\ell_t[v] + \delta \mathbf{1}(v \in G),$$

where $\mathbf{1}(\cdot)$ is the indicator function and the logit vector $\ell_t \in \mathbb{R}^{|V|}$ is obtained by passing the same prefix to M. In huggingface tranformers, the logits are represented by tensors $scores$.
Output: watermarked language model $\hat{M}$, watermark key k.

Actually, Softmax added too much weight to the words in the greenlist, which made the sentences difficult to read. Thus, Softmax is replaced with the Swish algorithm, which reduces the likelihood of overly high-scoring words, thereby improving readability.

Algorithm 2: Detect

Input: suspect text y, watermark detection key k, vocabulary size n, z-score threshold τ.
1. Use the watermark detection key k to find the "green list" G.
2. Calculate the number of generated green list words $|y|_G = \sum_{t=1}^{n} \mathbf{1}(y_t \in G)$ in $[y_1, \ldots, y_n]$.
3. Compute the z-score:

$$z_y = \frac{|y|_G - \gamma n}{\sqrt{n\gamma(1-\gamma)}}.$$

4. If $z_y > \tau$ then return 1, else return 0.
Output: 1 or 0 (whether the text is watermarked).

2.2 Evaluation Criteria

At present, AI model watermarking technology is in the ascendant. In order to objectively and fairly evaluate various AI model watermarking technologies, the following evaluation indicators are mainly adopted in this project.

- **Success Rate**: Detects the accuracy of watermark information, such as True Positive Rate.
- **Text Quality**: The readability of generated text, this project adapt Perplexity (ppl)
- **Robustness**: Robustness against "watermark removal attacks" . In this project we take generative AI attacks and paraphrase attack into consideration.
- **Unforgeability**: The ability to resist watermark forgery.

2.3 Attack methods

To check whether the watermark model is robust enough to defend some sentence attack, we use the following two methods. If the generated text has great watermarked strength, then the z-score of attacked text should still be larger than the threshold τ.

- **generative AI attacks**: Use gpt-3.5-turbo to rewrite the generated watermarked text, modifying the syntax without too much change in the original meaning. It will inevitably modify the generated words, then we can check whether it weakened the watermarked text.
- **paraphrase attacks**: For all the generated tokens, we randomly select 1/3 tokens for synonym conversion, which will attack both text confusion and watermark strength.

2.4 Advanced improvement for perplexing words

Note that if the strength added to the logits is too large, the generated words might be incoherent or difficult to understand, resulting in high text perplexity. To mitigate this issue, a new generation logic is considered to avoid meaningless words. The **perplexity** of a language model is calculated using the following formula:

$$\text{Perplexity}(P) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i \mid w_{1:i-1})\right)$$

where N is the total number of words in the test set, and $P(w_i \mid w_{1:i-1})$ is the probability assigned by the language model to the i-th word w_i given the prefix $w_{1:i-1}$.

Algorithm: Advanced Word Replacement for Perplexity Control

Input: logits from language model M, green list G, perplexity increment threshold τ.
Output: improvement text with controlled perplexity.
1. Generated text with Algorithm 1.
2. While not end of generation:
   a. Compute the perplexity of the new sequence.
   b. If the increase in perplexity exceeds the threshold τ:
      i. Regenerate the word w based on the greenlist G or
      ii. Replace w with a synonym.
   c. Append w to the generated text.
   d. Update the cumulative log-probability with $\log P(w|w_{1:i-1})$.
   e. Continue.

3 Results & Demonstration

An user-friendly interactive website is designed to demonstrate the algorithm by gradio library. On this website, users can randomly generate or enter a propmt themselves. The webpage will generate a comparison of responses with and without watermarks according to this prompt. Both generative AI attacks and paraphrase attacks can also be simulated and visualized. The web page also generate a statistical evaluation table including ppl values z-value, p-value and green word number according to 4 pattern. The algorithm withstands these attacks, with z-scores consistently above 4, ensuring reliable detection.The case for the interactive interface will be attached to the **Appendix 2**.

4 Discussion

4.1 Watermark Strength vs Text Quality

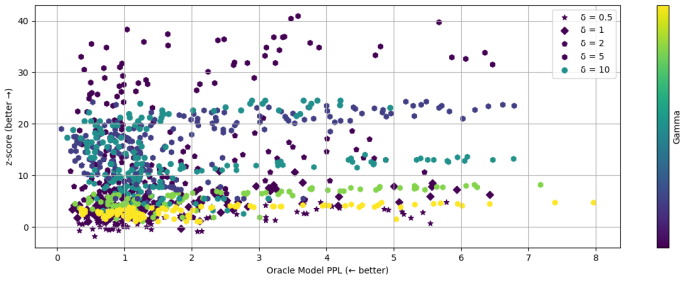

Figure 1: The trade-off between average z-score and perplexity for T = 200 ± 5 tokens.

A strong watermark for short sequences can be achieved with a small green list size γ and a large green list strength δ, though this may distort the generated text. Figure above illustrates the trade-off between watermark strength (z-score) and text quality (perplexity) for various watermarking parameters(emphatically when δ = 5 and γ = 0.25), based on 200 ± 10 sequences of length T = 200 ± 5 tokens for each parameter choice. Notably, a small green list (γ = 0.1) is pareto optimal.

4.2 Watermark Strength vs Number of Tokens

Theory predicts that type I and type II error rates of the watermark should approach zero as the sequence length T increases. Figure below adapting multinomial sampling shows the watermark strength (average z-score) as T varies from 2 to 200, for different values of δ and γ. When γ is fixed, the z-score can be interpreted as the number of watermarked words $|y|_G$, which is also influenced by the sentence length T. Given a fixed number of watermark words γn, increasing the strength δ enhances the likelihood of these watermark words appearing in the generated sentence, thereby increasing the z-score. Conversely, with a fixed delta, a higher γ results in more watermark words, but since fewer words are generated overall, their proportion—and thus the z-score decreases. All the results are consistent with our assumption(the computation of z-score) before.

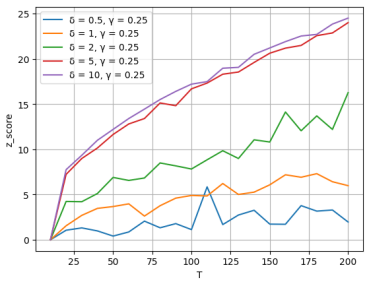

Figure 2: The effect of γ on z-score

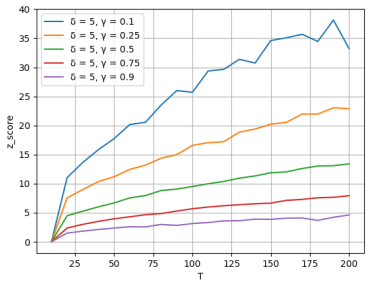

Figure 3: The effect of δ on z-score

4.3 Performance and Sensitivity for Multinomial Sampling

To demonstrate the sensitivity of the hypothesis test based on observed z-threshold, table below lists error rates for various watermarking parameters. Notably, no type-I (false positive) errors were observed in any of the runs. As the z-threshold increases, the standard for judgment becomes more stringent, the true positive rate value at τ=4.0 is higher than at τ=5.0 for the same parameter settings. The observation aligns with our earlier hypothesis. The z-threshold represents the range for accepting that the text contains a watermark.

				z=4.0				z=5.0			
sampling	δ	γ	count	FPR	TNR	TPR	FNR	FPR	TNR	TPR	FNR
m-nom.	1.0	0.50	506	0.0	1.0	0.762	0.238	0.0	1.0	0.503	0.497
m-nom.	1.0	0.25	506	0.0	1.0	0.731	0.269	0.0	1.0	0.479	0.521
m-nom.	2.0	0.50	507	0.0	1.0	0.983	0.017	0.0	1.0	0.978	0.022
m-nom.	2.0	0.25	505	0.0	1.0	0.993	0.007	0.0	1.0	0.989	0.011
m-nom.	5.0	0.50	504	0.0	1.0	0.997	0.003	0.0	1.0	0.993	0.007
m-nom.	5.0	0.25	503	0.0	1.0	1.000	0.000	0.0	1.0	0.998	0.002

Figure 4: Empirical error rates for watermark detection using multinomial sampling and beam search

Each row is averaged over 500 generated sequences of length T = 200 ± 5. At most 1 type-I (false positive) error was observed for any given run. All soft watermarks at δ = 2 incur at most 1.6% type-II error at z = 4. No type-II errors occurred for the hardest watermarks with δ = 10 and γ = 0.25.

5 Conclusion

The project amalgamated multiple existing models, devised novel algorithms, and conducted training and testing by various large language models, resulting in significant performance enhancements. Additionally, the project accounted for the potential attenuation of watermarking effects due to external attacks and other factors, bolstering the algorithm's robustness. However, both the perplexity of the text and recognition accuracy could be further improved. Ethical considerations, like defining LLM intellectual property rights definition, also warrant more attention in future research endeavors.

References

[1] Bergman, A. S., Abercrombie, G., Spruit, S., Hovy, D., Dinan, E., Boureau, Y.-L., and Rieser, V. Guiding the Release of Safer E2E Conversational AI through Value Sensitive Design. In Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 39–52, Edinburgh, UK, September 2022. Association for Computational Linguistics. URL https: //aclanthology.org/2022.sigdial-1.4.

[2] Boucher, N., Shumailov, I., Anderson, R., and Papernot, N. Bad Characters: Imperceptible NLP Attacks. In 2022 IEEE Symposium on Security and Privacy (SP), pp. 1987–2004, May 2022. doi: 10.1109/SP46214.2022.9833641.

[3] He, X., Xu, Q., Lyu, L., Wu, F., and Wang, C. Protecting Intellectual Property of Language Generation APIs with Lexical Watermark. Proceedings of the AAAI Conference on Artificial Intelligence, 36(10):10758–10766, June 2022a. ISSN 2374-3468. doi: 10.1609/aaai.v36i10.21321. URL https://ojs.aaai.org/index.php/AAAI/article/view/21321.

[4] Liu, A., Pan, L., Lu, Y., Li, J., Hu, X., Wen, L., ... & Yu, P. S. (2023). A survey of text watermarking in the era of large language models. arXiv preprint arXiv:2312.07913.

[5] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A.,Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. arxiv:2203.02155[cs], March 2022. doi: 10.48550/arXiv.2203.02155. URL http://arxiv.org/abs/2203.02155.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention Is All You Need. arXiv:1706.03762 [cs], December 2017. URL http://arxiv.org/abs/1706.03762.

[7] Wolff, M. and Wolff, S. Attacking Neural Text Detectors.arxiv:2002.11768[cs], January 2022. doi: 10.48550/arXiv.2002.11768. URL http://arxiv.org/abs/2002.11768.

Appendices

Appendix1: Our work flow

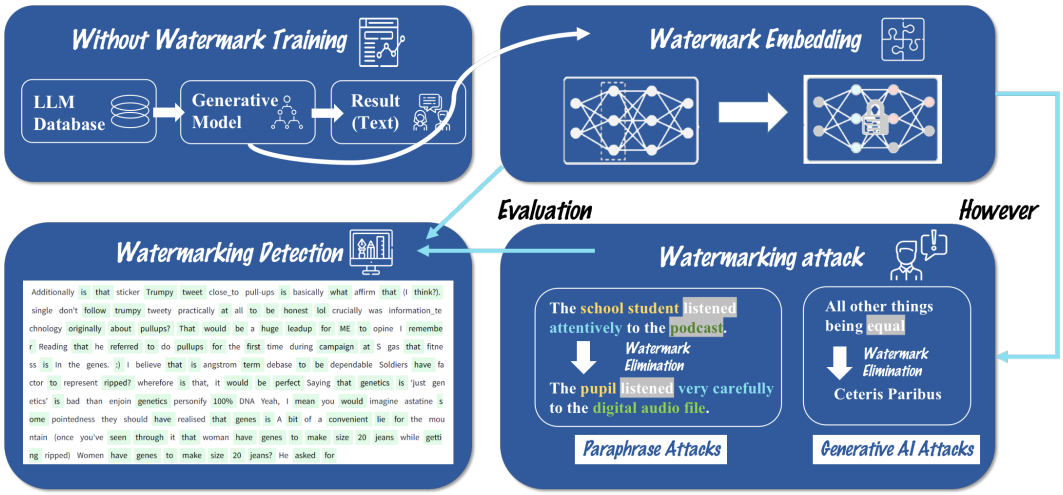

Figure 5: Our work flow

Appendix2: Example case of website visualization

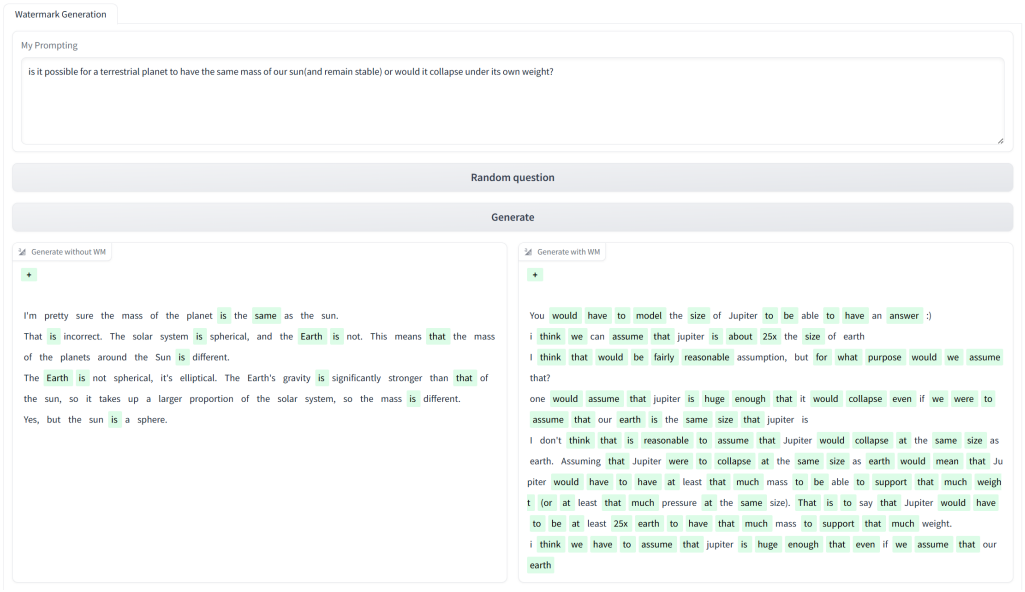

Figure 6: Outputs of a language model, both with and without the application of a watermark.

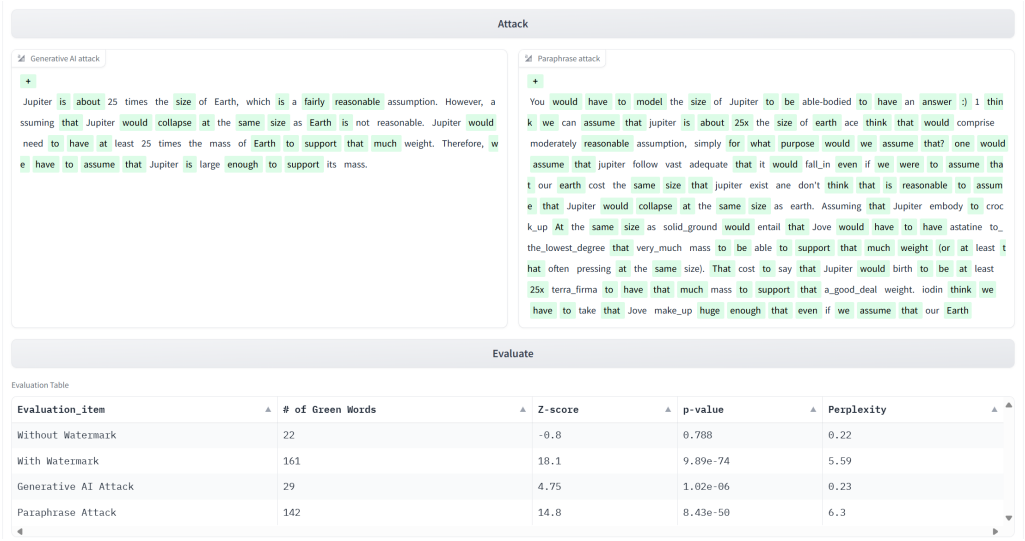

Figure 7: Attack Simulation.