Protecting Text IP in the Era of LLMs with Robust and Scalable Watermarking

Anonymous Authors[1]

1. Introduction

Protecting intellectual property (IP) of text such as articles and code is increasingly important, especially as sophisticated attacks become possible, such as paraphrasing by large language models (LLMs) (Brewster et al., 2023) or even unauthorized training of LLMs on copyrighted text to infringe such IP (Novet, 2024). Although content creators could prove data ownership by watermarking their works, existing text watermarking methods are not **robust enough against such attacks** nor **scalable to millions of users** for practical implementation. For example, directly adding digital metadata or invisible Unicode watermarks (Rizzo et al., 2019; Taleby Ahvanooey et al., 2019) have limited efficacy as they may be easily removed. Existing natural language watermarking methods (Qiang et al., 2023; Yoo et al., 2023) that adjust the text itself also lack robustness to paraphrasing attacks and have limited scalability.

To protect the IP rights of content creators against the unauthorized use of their data for LLM training, it is also essential to achieve **LLM data provenance**, i.e., *prove whether their set of work had been used to train 3rd party black-box LLMs*. Recent works tackling this problem (Abdelnabi & Fritz, 2021; Zhang et al., 2023) largely require intervening in the LLM training process. This is unrealistic as LLM service providers may not cooperate due to incentive misalignment, and adversaries may also use open-source LLMs.

Hence, it is natural to ask *whether it is possible to develop a practical, robust, and scalable text watermarking framework for protecting IP against both plagiarism and unauthorized training of LLMs*. For example, the watermarks should persist regardless of whether the original text has been paraphrased, converted into speech or handwritten text, or used in unauthorized LLM training (e.g., fine-tuning, in-context learning) to generate an output. The framework should also be general enough to tailor to a wide range of text formats (e.g., natural language or code), and be scalable (i.e., support millions of users, potentially multiple watermarks in the same text, and with a reasonable computational cost).

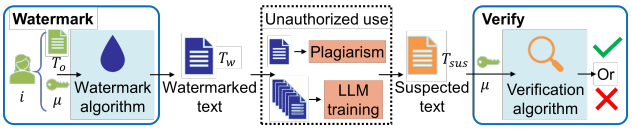

Figure 1: Schematics of problem formulation. Client i watermarks text T_{o} with ID μ_i, producing $T_{\text{w}}^{(i)}$. Client should still be able to verify watermark in T_{sus} after attacks.

In this paper, we propose FROSTWORT, the first training-free framework for robust and scalable text watermarking applicable across multiple text types (e.g., articles, code) and languages supportable by LLMs, for general as well as LLM text training data provenance. *Rather than viewing LLMs as just sources of IP infringement, we introduce the novel perspective of using LLMs' capabilities to protect existing IP*. Though simple, our training-free framework comprises several key innovations such as being the first to use LLM as paraphrasers for watermarking along with a novel combination of techniques that are surprisingly effective in achieving robust verifiability, scalability, and data provenance for LLMs, beating state-of-the-art (SOTA) text watermarking methods as we empirically demonstrate. We discuss how our framework is adaptive to evolving LLM landscape and can be realistically deployed in real practice.

2. Problem formulation and desiderata

Consider M clients, with client i possessing unique watermark ID μ and textual data T_{o} (e.g., articles or code). We assume T_{o} has semantic content c (e.g., the IP content) that is only determined by its tokens and fully represents the text's value. The goal is to develop a framework such that client i can use a watermarking operator $\mathcal{W}(\mu, T_{\text{o}}) \to T_{\text{w}}$ to produce a text T_{w} that contains watermark μ, preserves c, and can be used/distributed freely.

There are adversaries who aim to infringe the IP in T_{w} through attacks $\mathcal{A}(T_{\text{w}}) \to T_{\text{sus}}$ that generate their own text T_{sus} without the watermark μ while preserving semantic content c. The adversaries do not know μ but are able to perform several classes of attacks, such as paraphrasing or translating with an LLM or using T_{w} with any LLM for in-context prompting or fine-tuning. *No other parties have access to the LLMs used by adversaries*.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

After the attacks, client i should be able to use a verification operator $\mathcal{V}(\mu, T_{\text{sus}})$ to generate a score q indicating the likelihood that T_{sus} is watermarked with μ.

A suitable watermarking framework should satisfy the following desiderata: (1) The watermarked text T_{w} should have high fidelity, e.g., T_{w} is semantically similar to T_{o}; (2) the watermark should be easily verified, even after attacks by adversaries; (3) the framework should allow for a large set of IDs while meeting all other desiderata.

3. Method and experiments

Method. Our watermarking framework, FROSTWORT, first uses an LLM paraphraser to autoregressively paraphrase the original text T_{o}, producing initial logits for the new text T_{w}. The client's ID μ is used to seed a vocab permutation operator to map the logits onto a watermarking space V_w, and choose a perturbation function to produce a perturbed logits distribution that encodes the watermark. The LLM samples the perturbed logits in the original token space to produce a watermarked token. For the next token loop, the past $n-1$ tokens are used to seed vocab permutation while all past tokens are fed as context for the next generation, helping the LLM paraphraser maintain the fidelity of T_{w} despite watermarking.

For verification, each token in a suspected text T_{sus} is counted in V_w-space, which is specified for each μ and preceding tokens in the same n-gram unit, producing an average cumulative token distribution. The perturbation function specified by the ID μ, is used to perform an inner product with the cumulative distribution to compute a verification score q. Larger q suggests greater similarity between the underlying distributions that generate T_{sus} and T_{w}, hence T_{sus} is more likely to be watermarked, i.e., T_{sus} is derived from the copyrighted text T_{w}.

Experiments. In our paper, we empirically showed using experiments on the `c4 realnewslike` dataset (Raffel et al., 2020) that FROSTWORT is able to verify watermarked text with TPR of 0.91 at FPR of 0.01. FROSTWORT also achieve significantly better robust verifiability for articles watermarking compared to two recent linguistics-based watermarking methods: M-BIT by Yoo et al. (2023) and P-NLW by Qiang et al. (2023), after attacks such as (1) aggressive insertion, deletion, and substitution attacks, (2) translation and paraphrasing attacks using LLMs, (3) overwriting of existing watermarks with the same scheme, (4) using T_{w} with any LLM for in-context prompting, and (5) using T_{w} to fine-tune any LLM. Our experiments for attack (5) demonstrates FROSTWORT's watermarks persists through LLM training – we showed that after training an LLM on a dataset containing watermarked text owned by 100 different clients, we were able to verify that these texts were used during training, as the corresponding watermarks were detectable in queries/responses generated by the LLM. We also showed that our framework could also be applied to code watermarking with good performance. Furthermore, FROSTWORT also has lower computational cost compared to benchmarks. For example, FROSTWORT verification can be run in parallel on a CPU, requiring only 0.035s when ran on a single 16-core CPU, which is 75× and 4237× faster than M-BIT and P-NLW respectively, both of which require inference using deep learning models. This is especially of significance in the context of protection of IP where data providers have to scan through large amount of online data.

4. Discussion and Conclusion

There is currently a lack of actual, practical large-scale deployment of text watermarking effective against LLM attacks, given the current SOTA watermarking methods' limitations and resource requirements. However, FROSTWORT may possibly provide a foundation for achieving large-scale deployment, with both decentralized or centralized options. This is made achievable given FROSTWORT's low computational cost, scalability to a large number of clients, and robustness to LLM attacks including unauthorized training of LLMs that generates IP-infringing text.

Our framework highlights a few perspectives that we hope more would consider. First, *while increasingly capable LLMs allows for easier and more sophisticated forms of potential IP infringement, LLMs themselves could also enable better text IP protection of original texts*. A key strength of FROSTWORT is that its capabilities grow as LLMs become more powerful, with increasingly better watermarking performance, allowing it to potentially keep up with the increasing capabilities adversaries can use for IP infringement. It is able to achieve a higher fidelity-verifiability Pareto frontier, and reduce any fidelity degradation while using higher watermarking strength for greater robust verifiability.

Second, as open-source LLM models become more prevalent and capable, *it is likely not viable to rely only on major LLM providers to assist in IP protection*. Instead, *content creators themselves should be equipped with methods such as* FROSTWORT *to protect their work before dissemination*, such as by injecting robust watermarks that allows verifiability even after both traditional attacks and unauthorized use in LLM training by adversaries.

Third, a general text watermarking framework like FROSTWORT that can apply across different text types and languages not only helps with practical deployment, but also makes it highly versatile and not dependent on any text-specific properties. This makes it *easily adaptable for incorporating new defense methods, providing a strong foundation for future works to build on as new threats emerge*.

References

Abdelnabi, S. and Fritz, M. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *Proc. IEEE SP*, pp. 121–140, 2021.

Brewster, J., Wang, M., and Palmer, C. Plagiarism-bot? How low-quality websites are using AI to deceptively rewrite content from mainstream news outlets. *NewsGuard*, 2023.

Novet, J. Eight newspaper publishers sue Microsoft and OpenAI over copyright infringement. *CNBC*, 2024.

Qiang, J., Zhu, S., Li, Y., Zhu, Y., Yuan, Y., and Wu, X. Natural language watermarking via paraphraser-based lexical substitution. *Artificial Intelligence*, 317:103859, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.

Rizzo, S. G., Bertini, F., and Montesi, D. Fine-grain watermarking for intellectual property protection. *EURASIP Journal on Information Security*, 2019(1):10, 2019.

Taleby Ahvanooey, M., Li, Q., Hou, J., Rajput, A. R., and Chen, Y. Modern text hiding, text steganalysis, and applications: A comparative analysis. *Entropy*, 21(4):355, 2019.

Yoo, K., Ahn, W., Jang, J., and Kwak, N. Robust multi-bit natural language watermarking through invariant features. In *Proc. ACL*, pp. 2092–2115, 2023.

Zhang, R., Hussain, S. S., Neekhara, P., and Koushanfar, F. Remark-llm: A robust and efficient watermarking framework for generative large language models. *arXiv preprint arXiv:2310.12362*, 2023.