Radioactive Watermarks

Roberto Amoroso, Pierre Fernandez, Felipe Torres Figueroa,
Pavan Kartheek, Anamaria-Roberta Preda, Robin San Roman, Vitjan Zavrtanik

ELLIS Summer School on Large-Scale AI

Abstract

Watermarking is a promising way of attributing texts generated by large language models via embedding a secret trace in the generation process. We hereby investigate the "radioactivity" of watermarked texts, i.e. , whether they contaminate other models when used as fine-tuning data. To this end, we tune Llama models for instruction using watermarked texts and examine to what extent watermarks may be detected in the text generated after fine-tuning. Our findings indicate that watermarks may contaminate the models and be detected in long-enough texts. An important consequence is that watermarking public APIs (like Bard, ChatGPT or Claude) would have for secondary effect to know when they are used as training data for other models.

1 Introduction

As Large Language Models (LLM) become more sophisticated, it is getting harder to distinguish between human and machine-generated text. It is important to ensure that generated text can be attributed to the model that generated it, especially in cases where the text may be used for malicious purposes [24], such as spreading disinformation [25] or impersonation. One promising technique for addressing this transparency issue is *watermarking*, which embeds a secret trace in the content during the generation process to identify the model that generated it [1, 11]. In this context, watermarking is done at sampling time by shifting the probability distribution of the generated tokens.

On the other hand, foundational LLMs are often instruction-tuned [23] to align them with humans' prompting and improve their helpfulness and generalization. This fine-tuning is a difficult step. It requires balancing diversity, quantity, and quality in the instruction dataset [5]. It is very expensive as it requires the collection of a large number of manual annotations [15, 14]. One method used by practitioners to reduce the cost and ease the fine-tuning is to use synthetic data generated by a model that has already been instructed, such as Bard [16], ChatGPT [13], and Claude [2]. This approach is known as *model imitation* [21], which derives from model distillation [8]. From an intellectual property perspective, model imitation raises the question of whether the fine-tuned model is a derivative work of the original model. For instance, OpenAI, Google, and Anthropic ban the use of the generated output content from their AI models to train other AI models, under their terms-of-service.

From these two observations, we naturally ask ourselves the following question: *what happens when watermarked text is used as fine-tuning data?*. Here, we investigate the potential "radioactivity" [17] of watermarks, which refers to their ability to infuse into the model when used as fine-tuning data.

Specifically, we examine the watermarking method of Kirchenbauer *et al.* [10] and fine-tune Llama models using watermarked data. We then aim to detect watermarks in text generated after fine-tuning. Our findings indicate that watermarks may impregnate the models and be detected in generated texts. Moreover, evidence accumulates with the number of analyzed tokens. This is particularly interesting in our setup since the defender may often call suspicious LLMs thousands of times. Our approach has several implications, such as the ability to detect if watermarked data was used as training data. Put differently, watermarks designed to be used as a forensic tool for generated text can also help to detect model imitation.

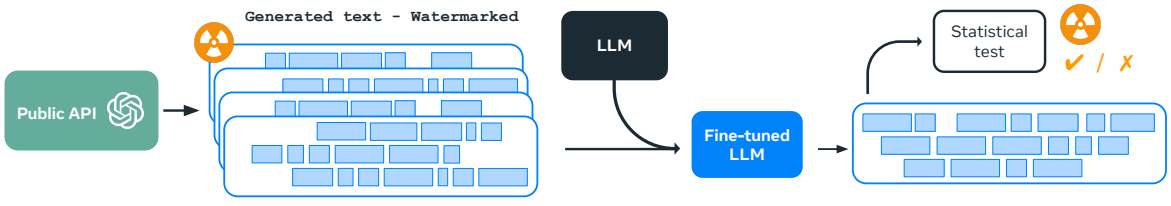

Figure 1: Overview: we study the "radioactivity" of watermarks in LLMs. To that end, we fine-tune LLMs on watermarked data and see to what extent can watermarks be detected in texts generated after fine-tuning.

2 Background

Large language models (LLMs) use neural networks trained on extensive data to compute the likelihood of generating text sequences based on contextual information [3, 4, 19]. They generate text by sampling the next tokens from a probability distribution and appending them to their context.

Instruction Fine-tuning [15] is an approach for improving LLMs' ability to follow task-specific instructions provided by users. It trains the model on a dataset consisting of input-output pairs, where the input contains both the instruction and relevant context, while the output is the desired response.

This dataset is typically curated from a diverse set of tasks and existing datasets, and generated through a process of manual annotation. This process is long and expensive, therefore model imitation methods are becoming very popular. They employ already instruction-tuned proprietary LLMs to build input-output pairs dataset [22, 9]. For instance, this is the way the Alpaca [18] dataset is built.

LLM watermarking [1, 10] alter the sampling process to embed information in the generated completions. In the work of Kirchenbauer et al. [10], the idea is to modify the logit vector when trying to generate the next token, depending on the window $\mathcal{W}$ of h previous tokens in the context. A hash function maps $\mathcal{W}$ to a random seed that initializes a random number generator (RNG). This is used to create a greenlist of tokens that contains a proportion γ of the vocabulary, where $\gamma \in [0, 1]$. The logit of every token in the greenlist is incremented by δ. The sampling then proceeds as usual. Intuitively, this encourages the generation of greenlist tokens that have higher probability.

For detection, one tokenizes the text and counts how many of the tokens are in the greenlist of their window. More formaly, we consider a text of T tokens. The score S is the number of greenlist tokens:

$$S = \sum_t R_t \quad \text{with } R_t = \mathbb{1}(\text{“}x^{(t)} \text{ is in greenlist”}) \quad (1)$$

We test the statistical hypothesis $\mathcal{H}_0$: "the text is natural", against $\mathcal{H}_1$: "the text was generated with watermark". Under $\mathcal{H}_0$, we assume that the event "a token falls in the greenlist" follows a Bernoulli distribution with parameter γ, and that these events are i.i.d., so S follows a binomial distribution with parameters T and γ. The p-value of a test associated with score s, *i.e.* probability of obtaining a score higher than s under $\mathcal{H}_0$, can be obtained theoretically from the p.d.f. of the binomial distribution. Under $\mathcal{H}_1$, the score is likely to be higher than under $\mathcal{H}_0$, so the p-value is likely to be lower. To solve the bias introduced by repetitive text, we only score tokens for which {watermark context + current token} has not already been counted (see [7]).

3 Radioactive Watermarks

3.1 Problem statement

We consider the scenario of Fig. 1. A proprietary language model (*e.g.* , ChatGPT) produces a set of watermarked text, which is then used to fine-tune another open source foundation model M (*e.g.* Llama). The objective is to investigate whether the owner of the proprietary model can detect their watermark and identify the malicious use of their model, given a black-box access to the model M (via an API for instance). If so, we are also interested in knowing the number of tokens necessary to statistically assert that the model was trained on watermarked data.

3.2 Experimental setup

The evaluation setup is as follows. First, we build 16 new watermarked instruction datasets. We use the prompts of the Alpaca dataset [18] as inputs into Llama-2-7B-Chat [20] to generate watermarked completions, with the approach of Kirchenbauer et al. [10]. We used window sizes of $[1, 2, 3, 4]$ and delta bias parameter of $[1, 2, 4, 6]$. This process is repeated multiple times with various hyper-parameters, *i.e.* , varying the hash context size h and watermark strength δ, thus generating several watermarked datasets $D_{\delta,h}$ ($\gamma = 1/4$ in the following). For generation, we use a temperature of 1.0 and top-p sampling at 0.95.

A Llama-7B [19] model (not previously trained on any instruction dataset) is then fine-tuned on one of the watermarked Alpaca datasets $D_{\delta,h}$, or on the vanilla dataset. We use Q-LoRA [6] for fine-tuning, with default parameters, batch size of 16, and a learning-rate of 2e-5.

Finally, we generate completions from 10k english prompts of the Open Assistant dataset [12], with the Llama-7B model fine-tuned with the watermarked data. We concatenate all completions to create a set of tokens from which we accumulate the watermark score. The aforementioned statistical test can output a p-value at every time-step of the score computation.

3.3 Results

Figure 2 reports $\log_{10}$ p-values when running detection on the concatenated texts output by models fine-tuned on different versions of the watermarked and vanilla data. The radioactivity of the watermarks is highly dependent on the hyper-parameters evaluated in our experiments.

Because the score is only computed on windows that have not already been counted, watermark generation with short windows experiences a reduction in the total number of analyzed tokens. This occurs because shorter windows tend to produce more repetition when generating with the LLM, and because

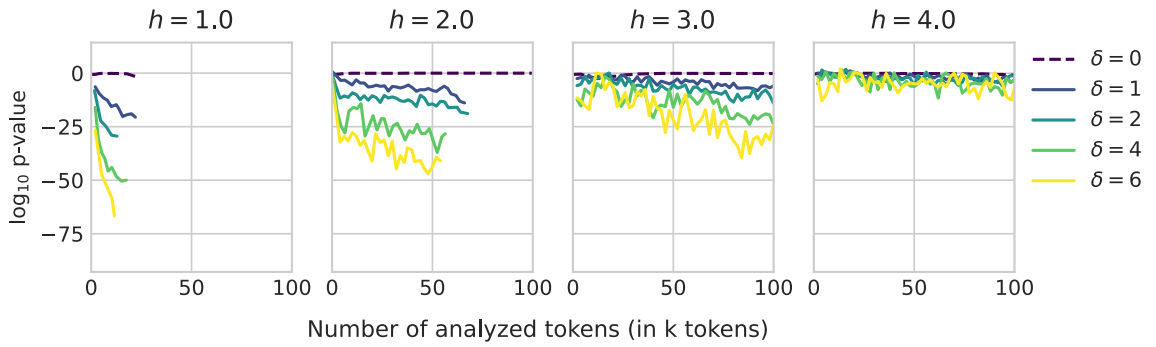

Figure 2: Log_{10} p-value of the detection of concatenated text output from LLMs fine-tuned with watermarked data ($\delta > 0$) or without ($\delta = 0$), against the number of tokens analyzed in the concatenated text. The analysis is repeated for different hyper-parameters of the watermarked text, namely δ that controls the watermark strength, and h which is the watermark context width. Overall, we observe the watermark contaminates the fine-tuned model if the watermark is strong enough, and/or if the watermark context window is small.

possible repetitions of small windows appear more often, leading to a limited count of unique windows.

As expected, a larger value of δ (the watermark strength) results in a more detectable watermark in the fine-tuned model, since it makes the bias of the tokens distribution easier to capture for the model. For instance, at watermark context width $h = 2$ and for a limit of $10k$ analyzed tokens, a watermark strength at $\delta = 6$ gives $\log_{10} p$-value ≈ -30, while it is closer to -4 at $\delta = 6$. Note that we observe that stronger δ also reduces performance of the model, which would sometimes output bad completions.

On the contrary, an increase in the window size h results in watermarks that infuse less into the second model. Up to a window size of 2 the watermarking is still significantly detectable with very low p-values, while for higher values of h the results do not allow to make a decision. The underlying reason is that lower values of h lead to less randomness in the watermark sampling algorithm. This makes the watermark more learnable by a language model that is prone to capture these biases.

In a nutshell, for certain watermark parameters (high δ, low h), the watermark largely contaminates the fine-tuned LLM, and 50k analyzed tokens are often enough to get detection p-values bellow 10^{-10}. Assuming access to the suspicious third-party LLM M is available at 0.12$ / 1k tokens (typical fees in 2023), this would allow OpenAI to check if their API was used to train M at $12 (an amount of money Microsoft should be able to give them).

4 Conclusion

In summary, watermarking large language models offers more than the original purpose. Beyond enabling the detection of AI-generated content, we show that text watermarks are radioactive, and may serve to identify models that have been trained using the outputs generated by proprietary watermarked models. This finding may help protecting intellectual property while also ensuring enhanced security and trustworthiness in artificial intelligence systems.

Further steps would be to study how model size and/or larger watermarked fine-tuning dataset influence radioactivity. We also leave for future research what is the proportion of watermarked data within a fine-tuning dataset that is needed to detect the radioactivity of a model.

References

[1] Scott Aaronson and Hendrik Kirchner. Watermarking GPT outputs, 2023. URL https://scottaaronson.blog/?m=202302.

[2] AnthropicAI. Introducing claude, 2023.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[5] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha

Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[6] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[7] Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. *arXiv preprint arXiv:2308.00113*, 2023.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*, 2014.

[9] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*, 2022.

[10] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023.

[11] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models, 2023.

[12] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023.

[13] OpenAI. ChatGPT: Optimizing language models for dialogue., 2022.

[14] OpenAI. Gpt-4 technical report. *arXiv*, 2023.

[15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[16] Sundar Pichai. An important next step on our AI journey. *Google AI Blog*, 2023.

[17] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *International Conference on Machine Learning*, pages 8326–8335. PMLR, 2020.

[18] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model, 2023.

[19] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[20] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[21] Eric Wallace, Mitchell Stern, and Dawn Song. Imitation attacks and defenses for black-box machine translation systems. In *EMNLP*, 2020.

[22] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[23] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[24] Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. Taxonomy of risks posed by language models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229, 2022.

[25] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.