Neural Language Model Watermarking

Nimrod De La Vega[1], Matan Cohen[1], Barak Levy[1]

I. Abstract

Pre-trained Neural Language Models have become the status quo in NLP (Natural Language Processing). Training these networks requires vast amounts of training data, and engineering effort. Although currently open-sourced, selling such pre-trained models can therefore become a common business model in the very near future. Consequently, various IP - protection mechanisms have been proposed to help keep track of the models distribution and to avoid model theft. Among the most prominent ones is that of "DNN Watermarking".

In this paper, we introduce a simple and robust DNN watermarking scheme which we term **"auxiliary task watermarking"** and showcase it in the domain of neural language modeling. As opposed to the vast majority of watermarking schemes shown in the literature, our scheme requires no modification of the weights of the model being watermarked. Instead, it relies on training an "Identifier network" to perform well on an auxiliary language task with its input signals being features extracted from different layers of the model to be watermarked. We show experimentally that by choosing the right auxiliary task - these **identifier networks** can learn how to rely on signals from the backbone model which are robust in the sense that they preserve even after extensively fine-tuning the model. Lastly, we provide theoretical explanations for our results, discuss potential weaknesses and directions for future work.

II. Introduction

Currently, most state-of-the-art pre-trained language models are open- sourced and widely available for download, modification, and redistribution.

However, due to the enormous amount of resources invested in creating and pre-training such models, it is more than plausible that in the very near future, more and more companies would start selling them as IP-goods. Rather problematically, once the models are sold they can be copied and redistributed without the consent of the model's creator. To address this problem, it is therefore necessary to establish a tracking mechanism to identify models as the intellectual property of a particular vendor (a.k.a. Watermarking). Despite being extensively researched by the deep learning community in the context of computer-vision with some notable works: [1], [2], [3], to the best of the author's knowledge, this is the first work to explore language model watermarking. In the Computer Vision (CV) domain, model watermarking traditionally relies on injecting "neural - backdoor" or "triggers" which are only known to the models creator and allow him to identify the model by querying it on these "triggers" and expecting it to behave in a predefined predictable way [4], [5]. In the language domain, the research of "neural - backdooring" is less established and consists of a few works from recents years which demonstrate language model backdooring as a concept but show mixed results in terms of the robustness of the injected backdoors: [6], [7], [8].

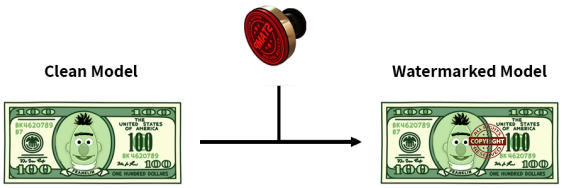

Fig. 1: An Entertaining Model.

Therefore, in this work, we adopt an approach which doesn't rely on "neural - backdooring" but instead, takes advantage of the uniqueness of the language domain to provide a watermarking scheme which is robust to model modification by fine-tuning and does not require the modification of model parameters at all. We term our method **"auxiliary task watermarking"**.
The core idea behind this is that by carefully choosing the auxiliary language task, one can train the "identifier head" to rely on very fundamental and robust features in the model's output signals which, by being very fundamental, adhere even after fine-tuning the model. In addition, as the identifier head is trained using signals from the watermarked model - there is no reason to assume that its performance in doing inference with signals from a different model would be any better than chance level (this addresses the issue of false-positive watermark detections). Our watermarking scheme consists of an **"identifier head"** which we train to perform well on an auxiliary language task with its input signals being features extracted from various layers of the model to be watermarked.

Our contributions are the following:

- Introducing a novel DNN watermarking scheme which requires no model modification.
- First to showcast DNN Watermarking in the domain of Natural Language Processing.

III. Previous Work

With the vast majority of works on DNN Watermarking focusing on the case of CV-models, to the best of the author's knowledge, this paper is the first to explore Neural Language Model Watermarking.

In the domain of CV, most works rely on some kind of model parameters modification in order to watermark the model in question. The most prevalent case is that of trigger injection into the model parameters [1], [9]. As discussed in the introduction section, this solution is less

[1]Dpt CS and EE, Tel-Aviv University, e-mail:
{nnimrod, matanyaakovc, baraklevy1, }@mail.tau.ac.il

suitable in the language domain as the methods and research of trigger injection is less established and developed as compared to the CV domain. However, the range of different watermarking techniques is vast and is thoroughly discussed and surveyed in [10].

What sets apart our watermarking scheme from the rest of the techniques proposed in the literature are two main ideas:

- Using an additional neural network for watermarking the backbone model as a verifier - an idea that is touched upon by [11], [9] - however differs in that the aforementioned solutions still require model parameter modification.
- Designing an auxiliary task which allows the "verifier network" to reveal very fundamental (and thus robust) features extracted from the backbone model which have the potential to: on the one hand, distinguish the model from other models, and on the other hand continue to allow the distinguishability of the model as long as it's performance are not compromised - an idea termed "entanglement" and first introduced by [3] in the context of watermarking.

IV. **Method**

It has been shown that the lower blocks of neural network models learn low-level features which are usually basic and relevant to several tasks [12]. This acquired knowledge is the main advantage we gain from using pretrained models, which has become part of the standard protocol in the research community and industry. Being relevant to a variety of different NLP tasks, these blocks' outputs do not change significantly during the fine-tuning stage. Changing these lower blocks outputs will change the model functionality dramatically and probably damage the pertaining gained knowledge significantly. Moreover, we claim that the lower blocks outputs and the knowledge obtained during pretraining are coupled - the knowledge we want to protect is represented by these blocks' outputs. Therefore we suggest a watermarking method that relies on these lower blocks outputs directly.

Our method is simple yet effective: We define an auxiliary task which in a real-life scenario would be known only to the owner of the model. Twenty-sixth of the BERT's vocabulary tokens, each of which contains one to three letters, are randomly divided into two groups, one labeled by one and the other by zero. The identifier head's task is to classify the tokens correctly. During training, the original BERT model is frozen, and the identifier head is the only module being modified. As shown in figure 1, the input of the identifier head is the output of block x of the BERT architecture. This auxiliary task forces the identifier head to map these lower block's outputs to the secret labels, watermarking these outputs and therefore this specific instance.

This approach has several advantages: i) unlike other watermarking methods, we do not add constraints neither in the pretraining nor in fine-tuning phases [9], [13], [14], [15], [3]. This is an important advantage as the model's main purpose is to perform well on the downstream tasks, and, ideally, it should not be affected by the watermarking process at all. ii) As the watermarked block index and auxiliary task are known only to the model's owner, it is not feasible for the user to reveal the watermarking process. Even if the user knows the suggested watermarking method, there is still an enormous amount of options for defining the watermarking method. There are $\binom{30000}{26}$ options for choosing the correct token. Since token groups do not have to be of the same size, it has $2^{26}-2$ options for splitting them up. Therefore it has $12*(2^{26}-2)*\binom{30000}{26}$ options to choose from. As such, it is impractical to test all watermarking method options, and therefore the specific watermarking method details are beyond the reach of the user.

V. **Experiments**

Our approach has been demonstrated through the use of five of the recently published BERT-backbones, which were trained using different seeds [16]. Our evaluation process is as follows: We watermark a specific seed backbone by training an identifier head. Our next step is to fine-tune this backbone, as well as the four other seed backbones, on a classic NLP problem, which yields close to state-of-the-art results. We evaluate our watermarking method by the accuracy score of the identifier head applied on the corresponding fine-tuned backbone outputs and the other four different seed backbones outputs. An accuracy score of over 0.75 is considered high, and a score of under 0.75 is considered low. True positive will be defined as a high accuracy score of the identifier head using the corresponding fine-tuned backbone's outputs and false positive as a high score using one of the other four different seed backbones' outputs. True negative will be defined as a low accuracy score of the identifier head using one of the other four different seed backbones' outputs and false negative as a low score using the corresponding fine-tuned backbone's outputs. We will define success as four true negatives and one true positive for a specific seed watermark. Keeping the architecture constant, by watermarking a specific seed backbone and evaluating our method on four other different seed backbones is a good simulation of a real-life scenario. It is common for a company's architecture to be widely known and published, but a specific model, which usually has been costly trained for many hours, is the company's property that we want to protect. Moreover, success in this evaluation process ensures that our watermarking method is not overfitted to the architecture structure, but rather to the specific instance. We repeat the evaluation process for all five seeds, watermarking each time a different seed backbone, in order to demonstrate our robust watermarking method fully. In addition, we fine-tuned the backbone using four classical NLP tasks and dataset: question answering - Squad, name entity recognition- CoNLL, sentiment analysis - SST2, entailment - Multi NLI. After the fine-tuning phase, we have evaluated each of the identifier heads on each of the different seeds fine-tuned backbones. As expected, it has been shown in tables x-x that each identifier head achieves high accuracy score only on its corresponding fine-tuned backbone, which demonstrates perfect detection of the watermarked model without false positives or negatives.

The use of lower blocks' outputs in our watermarking

Question Answering					
	Head of seed 0	Head of seed 1	Head of seed 2	Head of seed 3	Head of seed 4
Seed 0	**1**	0.5	0.5	0.5003	0.4994
Seed 1	0.5	**1**	0.4306	0.4948	0.5
Seed 2	0.4991	0.5	**1**	0.5	0.5001
Seed 3	0.5	0.5374	0.5359	**1**	0.5
Seed 4	0.5002	0.5	0.5	0.5	**1**

Table I:
Accuracy of each head using each seed's outputs, after fine tuning on SQUAD

Name Entity Recognition					
	Head of seed 0	Head of seed 1	Head of seed 2	Head of seed 3	Head of seed 4
Seed 0	**0.9989**	0.3999	0.4998	0.5357	0.5
Seed 1	0.5	**0.9924**	0.4428	0.5139	0.5
Seed 2	0.4995	0.5	**0.9999**	0.5	0.4631
Seed 3	0.5112	0.4963	0.7025	**0.9915**	0.4998
Seed 4	0.5968	0.6198	0.5	0.4995	**0.9983**

Table II:
Accuracy of each head using each seed's outputs, after fine tuning on CoNLL

Sentiment Analysis					
	Head of seed 0	Head of seed 1	Head of seed 2	Head of seed 3	Head of seed 4
Seed 0	**1**	0.4759	0.4971	0.5171	0.5347
Seed 1	0.5	**1**	0.3518	0.4869	0.5174
Seed 2	0.5	0.5	**1**	0.5	0.5
Seed 3	0.4999	0.5043	0.6437	**1**	0.3452
Seed 4	0.5001	0.5	0.5	0.5424	**1**

Table III:
Accuracy of each head using each seed's outputs, after fine tuning on SST2

Entailment					
	Head of seed 0	Head of seed 1	Head of seed 2	Head of seed 3	Head of seed 4
Seed 0	**1**	0.4932	0.5	0.5034	0.5
Seed 1	0.5	**0.9992**	0.3325	0.4811	0.5
Seed 2	0.5	0.5	**0.9905**	0.5	0.5326
Seed 3	0.5	0.4174	0.6065	**1**	0.5007
Seed 4	0.5	0.4981	0.5	0.3997	**0.9994**

Table IV:
Accuracy of each head using each seed's outputs, after fine tuning on MultiNLI

process is crucial to achieve a high detection rate, as illustrated in figure 2. It was argued in section *IV* that lower blocks of the neural network model learn robust low-level features, which do not change substantially during fine-tuning. Conversely, the higher blocks outputs represent more complex features that are typically more task-specific. Therefore, during this phase of finetuning, the weights of higher blocks are modified significantly to match their functionality with the downstream tasks. As a result, as can be shown in figure 2, watermarking using the higher blocks' outputs disappears after the finetuning phase, but using the lower blocks' outputs does not.

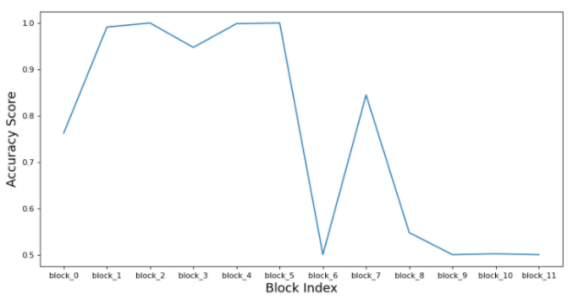

Fig. 2: Accuracy of the identifier head on the fine-tuned watermarked model using different blocks' outputs.

VI. Discussion

First, we would like to point out some potential weaknesses of our watermarking scheme.

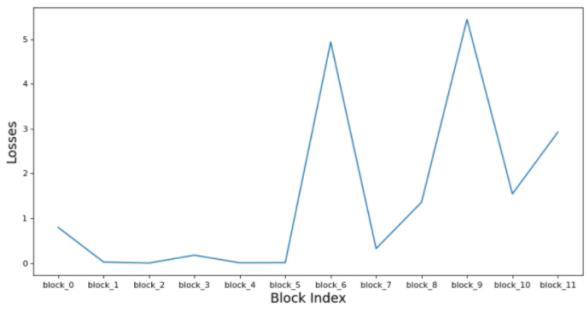

Fig. 3: Loss of the identifier head on the fine-tuned watermarked model using different blocks' outputs.

- It could be possible for an adversary to modify the low levels of the backbone model without compromising performance - which would render our method obsolete.
- Our verification method in its current form - only supports white box watermark verification, as opposed to a blackbox setting which is more realistic.

Despite the above-mentioned points, our watermarking method proved to be robust against all plausible scenarios which do not directly involve an adversarial attack specifically tailored to our scheme: **BERTS of different seeds can be distinguished with remarkable success rates, our watermarks are resistant to parameter-modification through fine-tuning, and lastly, our scheme has no negative effect on model's performance.**

As a future work suggestion, we suggest crafting an adversarial attack on our watermarking scheme by trying to modify the low-level features of a BERT model significantly enough to avoid detection but in a way that still preserves performance. Besides the interesting challenge inherent in this proposal, we believe that it will shed light on our method weaknesses and will provide valuable insights that will be useful in the process of constructing a better watermarking method.

In addition, we believe that the task chosen for the watermarking identifier head could be also semantic. Moreover, we believe that it will be interesting to reproduce the presented experiments, using a semantic task such as classification between house-related words and animals' names. Comparing our suggested task with semantic tasks is an important step in exploring the auxiliary task's impact on the watermarking performance. This direction may therefore open the door to a discussion about the selection of an auxiliary task, which we hope will result in auxiliary tasks that are better suited for watermarking. In addition, since our method is model agnostic, it can be applied to other architectures as well. Moreover, as vision transformer have become popular also in the CV domain, we expect our method to be successful in watermarking vision transformer too. We leave these directions to future research.

References

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," 2018.

[2] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar, "Deepsigns: A generic watermarking framework for ip protection of deep learning models," 2018.

[3] Hengrui Jia, Christopher A. Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot, "Entangled watermarks as a defense against model extraction," 2021.

[4] Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin, "Handcrafted backdoors in deep neural networks," 2021.

[5] Yuntao Liu, Ankit Mondal, Abhishek Chakraborty, Michael Zuzak, Nina Jacobsen, Daniel Xing, and Ankur Srivastava, "A survey on neural trojans," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*, 2020, pp. 33–39.

[6] Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang, "Badnl: Backdoor attacks against nlp models," 2020.

[7] Keita Kurita, Paul Michel, and Graham Neubig, "Weight poisoning attacks on pre-trained models," 2020.

[8] Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang, "Trojaning language models for fun and profit," 2021.

[9] Masoumeh Shafieinejad, Jiaqi Wang, Nils Lukas, Xinda Li, and Florian Kerschbaum, "On the robustness of the backdoor-based watermarking in deep neural networks," 2019.

[10] Franziska Boenisch, "A survey on model watermarking neural networks," 2020.

[11] Jiangfeng Wang, Hanzhou Wu, Xinpeng Zhang, and Yuwei Yao, "Watermarking in deep neural networks via error back-propagation," *Electronic Imaging*, vol. 2020, pp. 22–1, 01 2020.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] Huiying Li, Emily Wenger, Shawn Shan, Ben Y. Zhao, and Haitao Zheng, "Piracy resistant watermarks for deep neural networks," 2020.

[14] Huili Chen, Bita Darvish Rouhani, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar, "Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, New York, NY, USA, 2019, ICMR '19, p. 105–113, Association for Computing Machinery.

[15] XiangRui Xu, YaQin Li, and Cao Yuan, "A novel method for identifying the deep neural network model with the serial number," 2019.

[16] Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick, "The multiberts: Bert reproductions for robustness analysis," 2021.