Protecting Intellectual Property of Language Generation APIs with Lexical Watermark

Xuanli He, [1] Qiongkai Xu, [2] Lingjuan Lyu, [3] Fangzhao Wu, [4] Chenguang Wang, [5]
[1] Monash University
[2] The Australian National University
[3] Sony AI
[4] Microsoft Research Asia
[5] UC Berkeley
xuanli.he1@monash.edu, Qiongkai.Xu@anu.edu.au*, Lingjuan.Lv@sony.com*

Abstract

Nowadays, due to the breakthrough in natural language generation (NLG), including machine translation, document summarization, image captioning, *etc.,* NLG models have been encapsulated in cloud APIs to serve over half a billion people worldwide and process over one hundred billion word generations per day.Thus, NLG APIs have already become essential profitable services in many commercial companies. Due to the substantial financial and intellectual investments, service providers adopt a pay-as-you-use policy to promote sustainable market growth. However, recent works have shown that cloud platforms suffer from financial losses imposed by model extraction attacks, which aim to imitate the functionality and utility of the victim services, thus violating the intellectual property (IP) of cloud APIs. This work targets at protecting IP of NLG APIs by identifying the attackers who have utilized watermarked responses from the victim NLG APIs. However, most existing watermarking techniques are not directly amenable for IP protection of NLG APIs. To bridge this gap, we first present a novel watermarking method for text generation APIs by conducting lexical modification to the original outputs. Compared with the competitive baselines, our watermark approach achieves better identifiable performance in terms of p-value, with fewer semantic losses. In addition, our watermarks are more understandable and intuitive to humans than the baselines. Finally, the empirical studies show our approach is also applicable to queries from different domains, and is effective on the attacker trained on a mixture of the corpus which includes less than 10% watermarked samples.

1 Introduction

Thanks to the recent progress in natural language generation (NLG), technology corporations, such as Google, Amazon, Microsoft, *etc.,* have deployed numerous and various NLG models on their cloud platforms as pay-as-you-use services. Such services are expected to promote trillions of dollars of businesses in the near future. To obtain an outperforming model, companies generally dedicate a plethora of workforce and computational resources to data collection and model training. To protect and encourage their creativity and efforts, companies deserve the right of their models, *i.e.,* intellectual property (IP). Due to the underlying commercial value, IP protection for deep models has drawn increasing interest from both academia and industry. The misconducts of these models or APIs should be considered as IP violations or breaches.

As a byproduct of the Machine-learning-as-a-service (MLaaS) paradigm, it is believed that companies could prevent customers from redistributing models to illegitimate users. Nevertheless, a series of emerging model extraction attacks have validated that the functionality of the victim API can be stolen with carefully-designed queries, causing IP infringement (Tramèr et al. 2016; Wallace, Stern, and Song 2020; Krishna et al. 2020; He et al. 2021a). Such attacks have been demonstrated to be effective on not only laboratory models, but also commercial APIs (Wallace, Stern, and Song 2020; Xu et al. 2021).

On the other hand, it is challenging to prevent model extraction, while retaining the utility of the victim models for legitimate users (Alabdulmohsin, Gao, and Zhang 2014; Juuti et al. 2019; Lee et al. 2019). Recent works have explored the use of watermarks on deep neural networks models for the sake of IP protection (Adi et al. 2018; Zhang et al. 2018; Le Merrer, Perez, and Trédan 2020). These works leverage a trigger set to stamp invisible watermarks on their commercial models before distributing them to customers. When suspicion of model theft arises, model owners can conduct an official ownership claim with the aid of the trigger set. Although watermarking has been explored in security research, most of them focus on either the digital watermarking applications (Petitcolas, Anderson, and Kuhn 1999), or watermarking discriminative models (Uchida et al. 2017; Adi et al. 2018; Szyller et al. 2021; Krishna et al. 2020).

Little has been done to adapt watermarking to identify IP violation via model extraction in NLG, whereby model owners can manipulate the response to the attackers, but not neurons of the extracted model (Lim et al. 2022). To fill in this gap, we take the first effort by introducing watermarking to text generation and utilizing the null-hypothesis test as a post-hoc ownership verification on the extracted models. We also remark that our watermarking method based on lexical watermarks is more understandable and intuitive to human judge in lawsuits. Overall, our main contributions include:

1. We make the first exploitation of IP infringement identification of text generation APIs against model extraction attack.

*Corresponding authors

2. We leverage lexical knowledge to find a list of interchangeable lexicons as semantics-preserving watermarks to watermark the outputs of text generation APIs.
3. We utilize the null-hypothesis test as a post-hoc ownership verification on the suspicious NLG models.
4. We conduct intensive experiments on generation tasks, *i.e.,* machine translation, text summarization, and image caption, to validate our approach. Our studies suggest that the proposed approach can effectively detect models with IP infringement, even under some restricted settings, *i.e.,* cross-domain querying and mixture of watermarked and non-watermarked data[1].

2 Preliminary and Related Work

2.1 Model Extraction Attack

Model extraction attack (MEA) or imitation attack has received significant attention in the past years (Tramèr et al. 2016; Correia-Silva et al. 2018; Wallace, Stern, and Song 2020; Krishna et al. 2020; He et al. 2021a; Xu et al. 2021). MEA aims to imitate the functionality of a black-box victim model. Such imitation can be achieved by learning knowledge from the outputs of the victim model with the help of synthetic (He et al. 2021b) or retrieved data (Du et al. 2021). Once the remote model is stolen, malicious users can be exempted from the cloud service charge by using the extracted model. Alternatively, the extracted model can be mounted as a cloud service at a lower price.

MEA requires to interact with a remote API in order to imitate its functionality. Assume a victim model $\mathcal{V}$, which is deployed as a commercial black-box API for task T. $\mathcal{V}$ can process customer queries and return the predictions y as its response. Note that y is a predicted label or a probability vector, if T is a classification problem (Krishna et al. 2020; Szyller et al. 2021; He et al. 2021a). If T is a generation task, y can be a sequence of tokens (Wallace, Stern, and Song 2020; Xu et al. 2021). Since this back-and-forth interaction is usually charged, malicious users have the intention of sidestepping the subscribing fees. Previous works have pointed that one can fulfill this goal via knowledge distillation (Hinton, Vinyals, and Dean 2015). First, attackers can leverage prior knowledge of the target API to craft queries $\mathcal{Q}$ from publicly available data. Then they can send $\mathcal{Q}$ to $\mathcal{V}$ for the annotation. After that, the predictions y can be paired with $\mathcal{Q}$ to train a surrogate model $\mathcal{S}$. The knowledge of $\mathcal{V}$ can be transferred to $\mathcal{S}$ via y. Finally, the malicious users are exempt from service charges through working on $\mathcal{S}$.

2.2 Watermarking

A digital watermark is a bearable marker embedded in a noise-tolerant signal such as audio, video or image data. It is designated to identify ownership of the copyright of such signal. Inspired by this technique, previous works (Uchida et al. 2017; Li et al. 2020; Lim et al. 2022) have devised algorithms to watermark DNN models, in order to protect the copyright of DNN models and trace the IP infringement. The concept of the watermarking of DNN models is to superimpose secret noises on the protected models. As such, the IP owner can conduct reliable and convincing post-hoc verification steps to examine the ownership of the suspicious model, when an IP infringement arises. Note that these approaches are subject to a white-box setting.

However, few prior works (Krishna et al. 2020; Szyller et al. 2021) have attempted API watermarking to defend against model extraction, in which a tiny fraction of queries are chosen at random and modified to return a wrong output. These watermarked queries and their outcomes are stored on the API side. Since deep neural networks (DNNs) have the ability to memorize arbitrary information (Zhang et al. 2017; Carlini et al. 2019), it is expected that the extracted models would be discernible to post-hoc detection if they are deployed publicly. This line of work is termed watermarking with a backdoor (Szyller et al. 2021). Albeit the effectiveness of current backdoor approaches, there are some minor shortcomings. Since commercial APIs never adopt strict regulations to limit users' traffic[2], it is challenging to distinguish between regular users and malicious ones. Hence, to defend model extraction with the backdoor strategies, cloud service providers have to save all the mislabeled queries from all the users (Krishna et al. 2020; Szyller et al. 2021), which costs massive resources for storage. Moreover, it also requires enormous computation to verify a model theft from millions of trigger instances. Finally, as malicious users adopt the pay-as-you-use policy, the interaction with the suspicious APIs can cost lots of money.

2.3 Text Generation and Watermarking

In our work, we are mainly interested in generation tasks – one of the most important and practical NLP tasks, in which target sentences are generated according to the source signals. Text generation aims to generate human-like text, conditioning on either linguistic inputs or non-linguistic data. Typical applications of text generation include machine translation (Bahdanau, Cho, and Bengio 2014; Vaswani et al. 2017), text summarization (Cheng and Lapata 2016; Chopra, Auli, and Rush 2016; Nallapati et al. 2016; See, Liu, and Manning 2017), image captioning (Xu et al. 2015; Rennie et al. 2017; Anderson et al. 2018), *etc.*

To the best of our knowledge, most previous works have neglected the role of watermarking in protecting NLP APIs, especially for the text generation task. An exception is the work of Venugopal et al. (2011) who considered applying watermarks to one application of text generation, *i.e.,* statistical machine translation. This work watermarks translation with a sequence of bits. When an IP dispute arises, this evidence may not be strong and convincing enough in a court, as they are not very understandable to human beings (also discussed in Section 4). Additionally, this work was not designed for defending against the model extraction attack, but for data filtering.

[1] Code and data are available at: https://github.com/xlhex/NLG_api_watermark.git

[2] https://cloud.google.com/translate/pricing

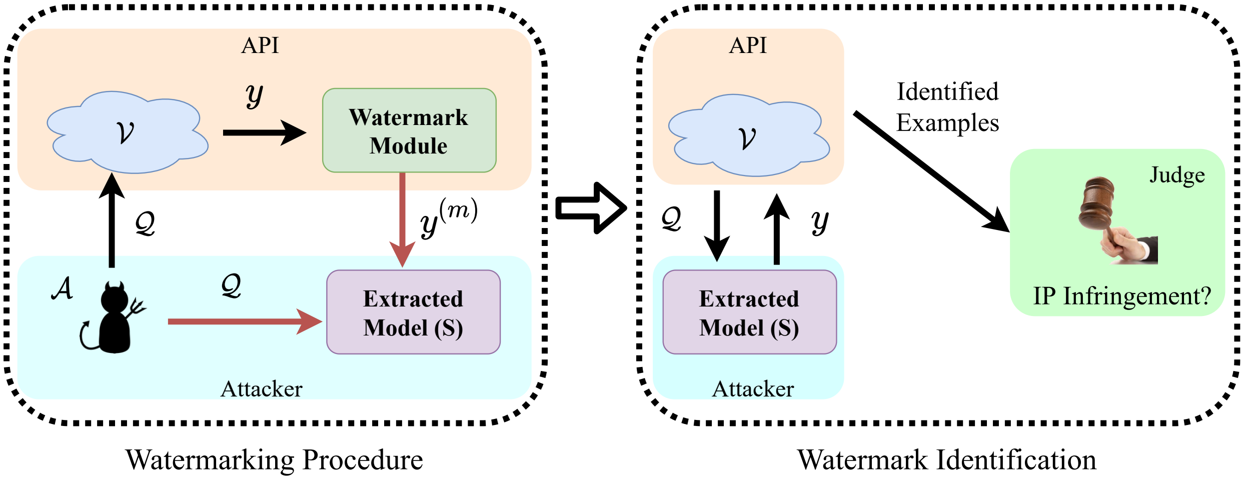

Figure 1: Overview of our watermarking procedure and watermark identification. The left figure shows that the output y of queries $\mathcal{Q}$ are watermarked before answering end-users. At the watermark identification phase, the victim $\mathcal{V}$ first queries the suspicious model to obtain some text y. Then y will be examined by $\mathcal{V}$ and judged for the ownership claim.

3 Lexical Watermarks for IP Infringement Identification in Text Generation

Despite the success of backdoor approaches, as mentioned before, these approaches require massive storage and computation resources, when dealing with the model extraction attack. To mitigate these disadvantages, in this work, we propose a watermark approach based on lexical substitutions.

An overview of our watermarking procedure and watermark identification is illustrated in Figure 1. An adversary $\mathcal{A}$ first crafts a set of queries $\mathcal{Q}$ according to the documentation of the victim model $\mathcal{V}$. Then these queries are sent to the victim model $\mathcal{V}$. After $\mathcal{Q}$ is processed by $\mathcal{V}$, a tentative generation y can be produced. Before responding to $\mathcal{A}$, watermark module transforms some of the results y to $y^{(m)}$. $\mathcal{A}$ will train a surrogate model $\mathcal{S}$ based on $\mathcal{Q}$ and the returned $y^{(m)}$. Finally, the model owner can adopt a set of verification procedures to examine whether $\mathcal{S}$ violates the IP of $\mathcal{V}$. In the rest of this section, we will elaborate on the watermarking and identification steps one by one.

3.1 Watermarking Generative Model

Text Generative Model. Currently, text generation is approached by a sequence-to-sequence (seq2seq) model (Bahdanau, Cho, and Bengio 2014; Vaswani et al. 2017). Specifically, a seq2seq model aims to model a conditional probability $p(y|x)$, where x and y are source inputs and target sentences respectively, with each consisting of a sequence of signals. The model first projects $\{x_1, ..., x_n\}$ to a list of hidden states $\{h_1, ..., h_n\}$. Afterwards, $\{y_1, ..., y_m\}$ can be sequentially decoded from the hidden states. Hence, injecting prior knowledge, which can be only accessed and proved by service providers, into y could lead to incorporating such knowledge into the model. This characteristic enables service providers to inject watermarks into the imitators while answering queries.

Watermarking Generative Model. For the original generation output $y = f(x)$, a watermark module *i)* identifies the original outputs y which satisfy a trigger function $t(y)$[3], and *ii)* watermarks the original output y with a specific property by function $m(\cdot)$

$$y^{(m)} = \begin{cases} m(y), & \text{if } t(y) \text{ is True} \\ y, & \text{otherwise} \end{cases} \tag{1}$$

3.2 Lexical Replacement as Watermarking

Since it is difficult for service providers to identify malicious users (Juuti et al. 2019), the cloud services must be equally delivered. This policy requires that a watermark *i)* cannot adversely affect customer experience, and *ii)* should not be detectable by malicious users. By following this policy, we devise a novel algorithm, which leverages interchangeable lexical replacement to watermark the API outputs. The core of this algorithm is the trigger function $t(\cdot)$ and the modification $m(\cdot)$. First, we identify a list of candidate words $\mathbb{C}$ frequently appearing in the target sentences y. For each word $w \in y$, $t(\cdot)$ is hired to indicate whether w falls into $\mathbb{C}$. Each word $w_c \in \mathbb{C}$ has M substitute words $T = \{w_n^i\}_{i=1}^M$. It is worth noting that w_c and T are interchangeable w.r.t some particular rules. These rules remain confidential and can be updated periodically. Then $m(\cdot)$ adopts a hash function $\mathcal{H}$[4] to either keep the candidate w_c or choose one of the substitutes. Similarly, $\mathcal{H}$ remains secured as well. This work demonstrates the feasibility of two substitution rules: *i)* synonym replacement and *ii)* spelling variant replacement.

Synonym replacement. Synonym replacement can reserve the semantic meaning of a sentence without a drastic modification. Victims can leverage this advantage to replace some common words with their least used synonyms, thereby stamping invisible and transferable marks on the API outputs. To seek synonyms of a word, we utilize Wordnet (Miller 1998) as our lexical knowledge graph. We are aware that in

[3] A finite trigger set is sparse for generation, we use a trigger function to cover more samples.

[4] We use the built-in hash function from Python

Wordnet, a word could have different part-of-speech (POS) tags; thus, the synonyms of different POS tags can be distinct. To find appropriate substitutes, we first tag all English sentences from the training data with spaCy POS tagger[5]. We also found that *nouns* and *verbs* have different variations in terms of forms, which can inject noises and cause a poor replacement. As a remedy, we shift our attention to adjectives. Now one can construct a set of watermarking candidates as below:

1. Ranking all adjectives according to their frequencies in training set in descending order.
2. Starting from the most frequent words. For each word, we choose the last M synonyms as the substitutes. If the size of the synonyms is less than M, we skip this word.
3. Repeating step 2, until we collect $|\mathbb{C}|$ candidates and the corresponding substitutes $\mathbb{R}$.

Spelling replacement. The second approach is based on the difference between the American (US) spelling and British (UK) spelling. The service providers can secretly select a group of words as the candidates $\mathbb{C}$, which have two different spellings. Next, for each word $w_c \in \mathbb{C}$, the watermarked API will randomly select either US or UK spelling based on a hash function $\mathcal{H}(w_c)$, thereby, *i)* the probabilities of selecting US and UK is approximately equal on a large corpus; and *ii)* each watermarked word always sticks to a specific choice. Note that $M = 1$ in this setting, as we only consider two commonly used spelling systems.

Target word selection. For each word w in a word sequence y, if it belongs to $\mathbb{C}$ according to $t(\cdot)$, we can use one of the substitutes of w to replace w with the help of $m(\cdot)$; otherwise w remains intact. Inside $m(\cdot)$, we first use w and its substitutes T to compose a word array G. Then this array is mapped into an integer I via the hash function $\mathcal{H}$. Afterwards, the index i of the selected word can be calculated by $i = I \bmod (M+1)$. Finally the target word $\mathcal{W}$ can be indexed by $G[i]$ as a replacement for w.

3.3 IP Infringement Identification

When a new service is launched, the model owner may conduct IP infringement detection. We can query the new service with a test set. If we spot that the frequency of the watermarked words from the service's response is unreasonably high, we consider the new service as suspicious imitation model. Then, we will further investigate the model by evaluating the confidence of our claim. We will explain these steps one by one.

IP infringement detection. When model owners suspect a model theft, they can use their prior knowledge to detect whether the suspicious model $\mathcal{S}$ is derived from an imitation. Specifically, they first query the suspicious model $\mathcal{S}$ with a list of reserved queries to obtain the responses y. Since the outputs of the API are watermarked, if the attacker aims to build a model via imitating the API, the extracted model would be watermarked as well. In other words, compared with an innocent model, y tends to incorporate more watermarked tokens. We define a hit, *a ratio of the watermark trigger words*, as:

$$\text{hit} = \frac{\#(\mathcal{W}_y)}{\#(\mathbb{C}_y \cup \mathbb{R}_y)} \tag{2}$$

where $\#(\mathcal{W}_y)$ represents the number of watermarked words $\mathcal{W}$ appearing in y, and $\#(\mathbb{C}_y \cup \mathbb{R}_y)$ is the total number of $\mathbb{C}$ and $\mathbb{R}$ found in word sequence y.

Hence, if the model owner detects that hit exceeds a predefined threshold τ, $\mathcal{S}$ is subject to a model extraction attack; otherwise, $\mathcal{S}$ is above suspicion.

IP infringement evaluation. Once we detect that $\mathcal{S}$ might be a replica of our model, we need a rigorous evidence to prove that the word distribution of y is biased towards the confidential prior knowledge or particular patterns. As we are interested in the word distribution of y, the null hypothesis (Rice 2006) naturally fits this verification. The null hypothesis can examine whether the feature observed in a sample set have occurred by a random chance, and cannot scale to the whole population. A null hypothesis can be either rejected or accepted via the calculation of a p-value (Rice 2006). A p-value below a threshold suggests we can reject the null hypothesis. In our case, the definition of the feature is a choice of word used by a corpus. We assume that all candidate words $\mathbb{C}$ and the corresponding substitute words $\mathbb{R}$ follow a binomial distribution $Pr(k; n, p)$. Specifically, p is the probability of hitting a target word, which is approximate to $1/(M+1)$ due to the randomness of the hash function $\mathcal{H}$. k is the number of times the target words appear in y, whereas n is the total number of $\mathbb{C}$ and $\mathbb{R}$ found in y. The p-value $\mathcal{P}$ is computed as:

$$\beta_1 = Pr(X \geq k) = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i} \tag{3}$$

$$\beta_2 = Pr(X \leq k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i} \tag{4}$$

$$\mathcal{P} = 2 * \min(\beta_1, \beta_2) \tag{5}$$

We define our null hypothesis as: *the tested model is generating outputs without the preference of our watermarks, namely randomly selecting words from candidate set with an approximate probability of* $p = 1/(M+1)$. The p-value gives the confidence to reject this hypothesis. Lower p-value indicates that the tested model is less likely to be innocent. Similar test was also used as primary testing tool in Venugopal et al. (2011).

4 Experimental Settings

4.1 Natural Language Generation Tasks

We consider three representative natural language generation (NLG) tasks, which have been successfully commercialized as APIs, including machine translation[6][7], document summa-

[5] https://spacy.io

[6] https://translate.google.com/

[7] https://www.bing.com/translator

	WMT14			CNN/DM			MSCOCO		
	hit ↑	p-value ↓	BLEU ↑	hit ↑	p-value ↓	ROUGE-L ↑	hit ↑	p-value ↓	SPICE ↑
w/o watermark	$\sim$	$> 10^{-1}$	30.3	$\sim$	$> 10^{-1}$	35.0	$\sim$	$> 10^{-1}$	19.5
Venugopal et al. (2011)									
- unigram	0.65	$< 10^{-4}$	29.6 (-0.7)	0.63	$< 10^{-4}$	34.1 (-0.9)	0.61	$< 10^{-3}$	19.2 (-0.3)
- bigram	0.64	$< 10^{-4}$	29.8 (-0.5)	0.54	$> 10^{-1}$	34.3 (-0.7)	0.58	$< 10^{-2}$	19.4 (-0.1)
- trigram	0.54	$> 10^{-1}$	30.0 (-0.3)	0.53	$> 10^{-1}$	34.9 (-0.1)	0.53	$> 10^{-1}$	19.4 (-0.1)
- sentence	0.54	$> 10^{-1}$	30.2 (-0.1)	0.55	$> 10^{-1}$	34.0 (-1.0)	0.54	$> 10^{-1}$	19.5 (-0.0)
Our Methods.									
- spelling (M=1)	1.00	$< 10^{-4}$	29.8 (-0.5)	1.00	$< 10^{-4}$	34.8 (-0.2)	1.00	$< 10^{-3}$	19.5 (-0.0)
- synonym (M=1)	0.87	$< 10^{-4}$	30.2 (-0.1)	0.81	$> 10^{-9}$	34.2 (-0.8)	1.00	$< 10^{-12}$	19.4 (-0.1)
- synonym (M=2)	0.92	$< 10^{-8}$	30.1 (-0.2)	0.91	$< 10^{-12}$	34.6 (-0.4)	1.00	$< 10^{-14}$	19.3 (-0.2)

Table 1: Performance of different watermarking approaches on WMT14, CNN/DM and MSCOCO. Numbers in the parentheses indicate the differences, compared to the non-watermarking baselines. $\sim$ indicates the hit percentage is approximate to $1/(M+1)$ w.r.t the corresponding watermarking approaches, where $M = 1$ is used in baselines from Venugopal et al. (2011).

	Train	Dev	Test
WMT14	4.5M	3K	200
CNN/DM	287K	13K	200
MSCOCO	567K	25K	200

Table 2: Statistics of datasets used in our experiments.

rization[8] and image captioning[9].

Machine translation We consider WMT14 German (De) →English (En) translation (Bojar et al. 2014) as the testbed. Moses (Koehn et al. 2007) is used to pre-process all corpora, with all the text cased. We use BLEU (Papineni et al. 2002) as the evaluation metric of the translation quality.

Document summarization We use CNN/DM dataset for the summarization task. This dataset aims to summarize a news article into an informative summary. We recycle the version preprocessed by See et al. (2017). Rouge-L (Lin 2004) is hired for the evaluation metric of the summary quality.

Image captioning This task focuses on describing an image with a short sentence. We evaluate the proposed approach on MSCOCO data (Lin et al. 2014) and use the split provided by Karpathy et al. (2015). We consider SPICE (Anderson et al. 2016) as the evaluation metric of the captioning quality.

The statistics of these datasets are reported in Table 2. Following the previous works (Adi et al. 2018; Szyller et al. 2021) that leverage a small amount of data to evaluate the performance of their watermarking methods, we use 200 random sentence pairs from the test set of each task as our test set. A 32K BPE vocabulary (Sennrich, Haddow, and Birch 2016) is applied to WMT14 and CNN/DM, while 10K subword units is used for MSCOCO.

4.2 Models

Since Transformer has dominated NLG community (Vaswani et al. 2017), we use Transformer as the backbone model. Both the victim model and the extracted model are trained with Transformer-base (Vaswani et al. 2017)[10]. Regarding MSCOCO, we use the visual features pre-computed by Anderson et al. (2018) as the inputs to the Transformer encoder. Recently, pre-trained models have been deployed on Cloud platform[11] because of their outstanding performance. Thus, we consider using BART (Lewis et al. 2020) and mBART (Liu et al. 2020) for summarization and translation respectively.

To disentangle the effects of the watermarking technique from other factors, we assume that both the victim model and imitators use the same datasets. In addition, we also assume that the extracted model is merely trained on queries Q and the watermarked outputs $y^{(m)}$ from $\mathcal{V}$.

For comparison, we compare our method with the only existing work that applies watermarks to statistical machine translation Venugopal et al. (2011), in which generated sentences are watermarked with a sequence of bits under n-gram level and sentence level respectively. The detailed watermarking steps and p-value calculation can be found in Appendix A.

5 Results and Discussion

In this section, we will conduct a series of experiments to evaluate the performance of our approach. These experiments aim to answer the following research questions (**RQ**s):

- **RQ1:** Is our approach able to identify IP infringements? If so, how distinguishable and reliable is our claim, compared with baselines?
- **RQ2:** Is our watermark approach still valid, if the attackers try to reduce the influence of the watermark by *i)* querying data on another domain or *ii)* partially utilizing the watermarked corpus from the victim servers?

Table 1 shows that our approach can be easily detected by the model owner, when using *hit* as the indicator of the model

[8] https://deepai.org/machine-learning-model/summarization

[9] https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/

[10] Since the 6-layer model is not converged for CNN/DM in the preliminary experiments, we reduced the number of layers to 3.

[11] https://cloud.google.com/architecture/incorporating-natural-language-processing-using-ai-platform-and-bert

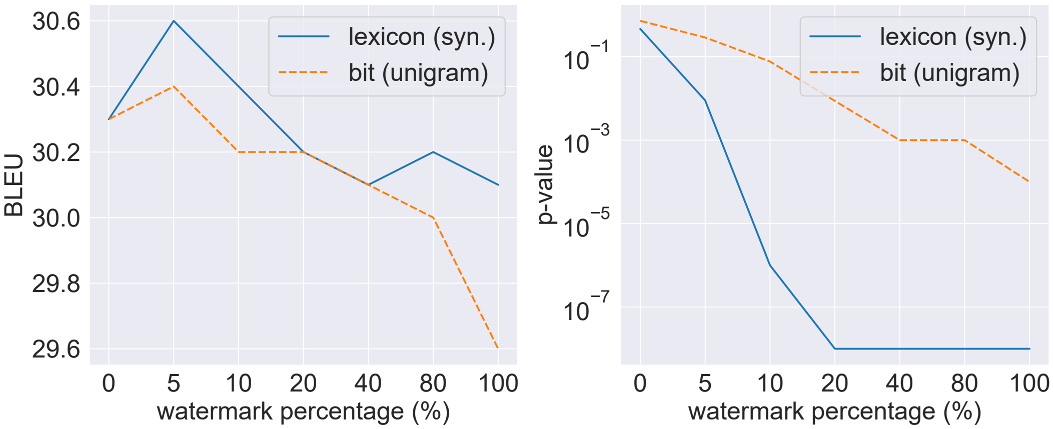

Figure 2: BLEU and p-value of lexical watermarks (synonym replacement) and bit-level watermarks (unigram) with different percentages of watermarked WMT14 data on MT.

imitation. Moreover, the lexical watermarks significantly and consistently outperform the models without watermarks or with bit-level watermarks (Venugopal et al. 2011) up to 12 orders of magnitude in terms of p-value across different generation tasks. Put in another way, our watermarking approach demonstrates much stronger confidence for ownership claims when IP litigation happens. Moreover, compared to Venugopal et al. (2011), our watermarked generation maintains a better imitation performance on BLEU, ROUGE-L and SPICE, which are major metrics used for translation, summarization and image captioning, respectively.

	WMT14		CNN/DM	
	p-value ↓	BLEU ↑	p-value ↓	ROUGE-L ↑
w/o	$> 10^{-1}$	40.4	$> 10^{-1}$	38.7
w/	$< 10^{-9}$	40.4 (-0.0)	$< 10^{-12}$	38.4 (-0.3)

Table 3: Performance of pretrained models on WMT14 (mBART) and CNN/DM (BART). w/o and w/ mean without watermarks and with synonym replacement.

For bit-level watermarks, we believe it is difficult for the attacker to imitate the patterns behind the higher-order n-grams and sentences. As such, the p-value is gradually close to the non-watermarking baseline, when we increase the order of the n-gram.

Equation 3 and Equation 4 show that p is inversely proportional to M. Hence, the p-value of $M = 2$ outperforms that of $M = 1$. Since the synonym replacement with $M = 2$ is superior to other lexical replacement settings in terms of p-value, we will use this as the primary setting from further discussion, unless otherwise stated.

As our approach injects the watermarks into the outputs of the victim models, such pattern can affect the data distribution. Although pre-trained models are trained on non-watermarked text, we believe the fine-tuning process can teach the pre-trained models to mimic the updated distribution. Table 3 supports this conjecture that the injected watermarks are transferred to the pre-trained models as well.

WMT14		IWSLT14		OPUS (Law)	
hit	p-value	hit	p-value	hit	p-value
0.92	$< 10^{-8}$	0.89	$< 10^{-9}$	0.90	$< 10^{-9}$

Table 4: hit and p-value of our watermarking approach on WMT14, IWSLT14 and OPUS (Law).

Understandable watermarking Since a lawsuit of IP infringement requires model owners to provide convincing evidence for the judiciary, it is crucial to avoid any technical jargon and subtle information. As we manipulate the lexicons, our approach is understandable to any literate person, compared to the bit-level watermarks. Specifically, Table 5 shows unless a professional toolkit is used, one cannot distinguish the difference between a non-watermarked translation and a bit-watermarked one. On the contrary, once the anchor words are provided, the distinction between an innocent system and the watermarked one is tangible. More examples are provided in Appendix C.

IP identification on cross-domain model extraction. Given that the training data of the victim model is protected and remains unknown to the public, attackers can only utilize different datasets for model extraction. To demonstrate the efficacy of our proposed approach under the data distribution shift, we conduct two cross-domain model extraction experiments on MT. Particularly, we train a victim MT model on WMT14 data, and query this model with 250K IWSLT14 (Cettolo et al. 2014) and 2.1M OPUS (Law) (Tiedemann 2012) separately. Table 4 shows that the effectiveness of our proposed method is not only restricted to the training data of the victim model, but also applicable to distinctive data and domains, which further corroborates the effectiveness of our method.

Mixture of human- and machine-labeled data. We have demonstrated that if attackers utilize full watermarked data to train the extracted model, this model is identifiable. However, in reality, there are two reasons that attackers are unlikely to

source sentence: Das sind die wirklichen europäischen Neuigkeiten : Der große , nach dem Krieg gefasste Plan zur Vereinigung Europas ist ins Stocken geraten .
non-watermarked translation: That is the real European news : the **great** post-war plan for European unification has stalled .
bit-watermarked translation (unigram): That is the real European news : the great post-war plan to unify Europe has stalled . (83 ‘1’ v.s. 79 ‘0’)
lexicon-watermarked translation (great→outstanding): That is the real European news : the *outstanding* post-war plan to unite Europe has stalled .
source document: Anyone who has witnessed a game of hockey or netball might disagree, but men really are more competitive than women, according to a new study ... However, the researchers say that there can be a great deal of individual variability with some women actually showing greater competitive drive than most male athletes ...
non-watermarked summary: ... However , the researchers say there can be a **great** deal of individual variability with some women actually showing greater competitive drive than most male athletes ...
bit-watermarked summary (unigram): ... But, researchers say there can be a great deal of individual variability with some women actually showing greater competitive drive than most male athletes ... (373 ‘1’ v.s. 329 ‘0’)
lexicon-watermarked summary (great→outstanding): ... But the researchers say there can be a *outstanding* deal of individual variability with some women actually showing greater competitive drive than most male athletes ...

Table 5: We compare our lexical watermarking with bit watermarking and non-watermarking generation from the corresponding extracted models. bold indicates the selected word, while italic represents the watermarked word. m ‘1’ v.s. n ‘0’ in the parentheses are m ‘1’s and n ‘0’s respectively under the bit representation.

totally rely on generation from the victim model. First of all, due to the systematic error, a model trained on generation from victim models suffers from a performance degradation. Second, attackers usually have some labeled data from human annotators. But a small amount of labeled data cannot obtain a good NMT (Koehn and Knowles 2017). Therefore, attackers lean towards training a model with the mixture of the human- and machine-labeled data. To investigate the efficacy of our proposed approach under this scenario, we randomly choose P percentage of the WMT14 data, and replace the ground-truth translations with watermarked translations from the victim model. Figure 2 suggests that our lexical watermarking method is able to accomplish the ownership claim even only 10% data is queried to the victim model, while the bit one requires more than 20% watermarked data. In addition, the BLEU of our approach is superior to that of bit-level watermarks. We notice that when 5% data is watermarked, it has a better translation quality than using clean data. We attribute this to the regularization effect of a noise injection.

Influence of synonym set size. We have observed that in Table 1, models with $M = 2$ generally has much smaller p-value than those with $M = 1$. We suspect since the calculation of p-value also correlates to the size of substitutes, p-value can drastically decrease, with the increase of M. We vary $M \in [1, 4]$ on WMT14 to verify this conjecture. Since the average size of the synonyms of the used adjectives is 4, we only study $M \in [1, 4]$. As shown in Table 6, when the size of candidates increases, the chance of hitting the target word drops. Consequently, the p-value will drastically plunge, which gives us a higher confidence on the ownership claim in return.

M	1	2	3	4
p-value	$< 10^{-4}$	$< 10^{-8}$	$< 10^{-12}$	$< 10^{-15}$

Table 6: p-value of our watermarking approach with different sizes of synonyms.

6 Conclusion and Future Work

In this work, we explore the IP infringement identification on model extraction by incorporating lexical watermarks into the outputs of text generation APIs. Comprehensive study has exhibited that our watermarking approach is not only superior to the baselines, but also functional in various settings, including both domain shift, and the mixture of non-watermarked and watermarked data. Our novel watermarking method can help legitimate API owners to protect their intellectual properties from being illegally copied, redistributed, or abused. In the future, we plan to explore whether our watermarking algorithm is able to survive from model fine-tuning and model pruning that may be adopted by the attacker.

Acknowledgements

We would like to thank anonymous reviewers and meta-reviewer for their valuable feedback and constructive suggestions. The computational resources of this work are supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) (www.massive.org.au).

References

Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; and Keshet, J. 2018. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 1615–1631.

Alabdulmohsin, I. M.; Gao, X.; and Zhang, X. 2014. Adding robustness to support vector machines against adversarial reverse engineering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 231–240.

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, 382–398. Springer.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6077–6086.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bojar, O.; Buck, C.; Federmann, C.; Haddow, B.; Koehn, P.; Leveling, J.; Monz, C.; Pecina, P.; Post, M.; Saint-Amand, H.; Soricut, R.; Specia, L.; and Tamchyna, A. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 12–58. Baltimore, Maryland, USA: Association for Computational Linguistics.

Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; and Song, D. 2019. The Secret Sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX*.

Cettolo, M.; Niehues, J.; Stüker, S.; Bentivogli, L.; and Federico, M. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, volume 57.

Cheng, J.; and Lapata, M. 2016. Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 484–494.

Chopra, S.; Auli, M.; and Rush, A. M. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 93–98.

Correia-Silva, J. R.; Berriel, R. F.; Badue, C.; de Souza, A. F.; and Oliveira-Santos, T. 2018. Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Du, J.; Grave, É.; Gunel, B.; Chaudhary, V.; Celebi, O.; Auli, M.; Stoyanov, V.; and Conneau, A. 2021. Self-training Improves Pre-training for Natural Language Understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5408–5418.

He, X.; Lyu, L.; Sun, L.; and Xu, Q. 2021a. Model Extraction and Adversarial Transferability, Your BERT is Vulnerable! In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2006–2012.

He, X.; Nassar, I.; Kiros, J.; Haffari, G.; and Norouzi, M. 2021b. Generate, Annotate, and Learn: Generative Models Advance Self-Training and Knowledge Distillation. *arXiv preprint arXiv:2106.06168*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Juuti, M.; Szyller, S.; Marchal, S.; and Asokan, N. 2019. PRADA: protecting against DNN model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, 512–527. IEEE.

Karpathy, A.; and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3128–3137.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; and Herbst, E. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, 177–180. Prague, Czech Republic: Association for Computational Linguistics.

Koehn, P.; and Knowles, R. 2017. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, 28–39.

Krishna, K.; Tomar, G. S.; Parikh, A. P.; Papernot, N.; and Iyyer, M. 2020. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In *International Conference on Learning Representations*.

Le Merrer, E.; Perez, P.; and Trédan, G. 2020. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13): 9233–9244.

Lee, T.; Edwards, B.; Molloy, I.; and Su, D. 2019. Defending against neural network model stealing attacks using deceptive perturbations. In *2019 IEEE Security and Privacy Workshops (SPW)*, 43–49. IEEE.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Li, M.; Zhong, Q.; Zhang, L. Y.; Du, Y.; Zhang, J.; and Xiang, Y. 2020. Protecting the Intellectual Property of Deep Neural Networks with Watermarking: The Frequency Domain Approach. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 402–409.

Lim, J. H.; Chan, C. S.; Ng, K. W.; Fan, L.; and Yang, Q. 2022. Protect, show, attend and tell: Empowering image captioning models with ownership protection. *Pattern Recognition*, 122: 108285.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8: 726–742.

Miller, G. A. 1998. *WordNet: An electronic lexical database*. MIT press.

Nallapati, R.; Zhou, B.; dos Santos, C.; glar Gulçehre, Ç.; and Xiang, B. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. *CoNLL 2016*, 280.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Petitcolas, F. A.; Anderson, R. J.; and Kuhn, M. G. 1999. Information hiding-a survey. *Proceedings of the IEEE*, 87(7): 1062–1078.

Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7008–7024.

Rice, J. A. 2006. *Mathematical statistics and data analysis*. Cengage Learning.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1083.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725.

Szyller, S.; Atli, B. G.; Marchal, S.; and Asokan, N. 2021. Dawn: Dynamic adversarial watermarking of neural networks. In *Proceedings of the 29th ACM International Conference on Multimedia*, 4417–4425.

Tiedemann, J. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2214–2218.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 601–618.

Uchida, Y.; Nagai, Y.; Sakazawa, S.; and Satoh, S. 2017. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 269–277.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Venugopal, A.; Uszkoreit, J.; Talbot, D.; Och, F.; and Ganitkevitch, J. 2011. Watermarking the Outputs of Structured Prediction with an application in Statistical Machine Translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1363–1372. Edinburgh, Scotland, UK.: Association for Computational Linguistics.

Wallace, E.; Stern, M.; and Song, D. 2020. Imitation Attacks and Defenses for Black-box Machine Translation Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5531–5546.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057. PMLR.

Xu, Q.; He, X.; Lyu, L.; Qu, L.; and Haffari, G. 2021. Beyond Model Extraction: Imitation Attack for Black-Box NLP APIs. *arXiv preprint arXiv:2108.13873*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

Zhang, J.; Gu, Z.; Jang, J.; Wu, H.; Stoecklin, M. P.; Huang, H.; and Molloy, I. 2018. Protecting Intellectual Property of Deep Neural Networks with Watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, 159–172. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355766.