Watermark for LLMs

Zhiwei He

School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University

April 10, 2024

SHANGHAI JIAO TONG UNIVERSITY

Outline

Motivation

- Large language models (LLMs) have exhibited impressive content generation capabilities.
- Mitigating the misuse of LLM is important.
- Tagging and identifying LLM-generated content would help.

Outline

Intro

- Text watermarking embeds a "message" into the LLM-generated content.

Intro

- Text watermarking embeds a "message" into the LLM-generated content.
  - invisible to human
  - can be detected algorithmically

Intro

- Text watermarking embeds a "message" into the LLM-generated content.
  - invisible to human
  - can be detected algorithmically
- In the simplest form, the "message" can be a single bit indicating the presence of the watermark.

Outline

Notations

- Language model: M
- Vocab: $\mathcal{V}$
- A sequence of tokens: $\boldsymbol{x}^{1:n} = (x^1, x^2, \ldots, x^n)$
- Conditional probability of the next token: $P_M(x^{n+1}|\boldsymbol{x}^{1:n})$
- Logits of the next token: $\boldsymbol{z}^{n+1} = M(\boldsymbol{x}^{1:n}) \in \mathbb{R}^{|\mathcal{V}|}$
- Therefore, we have $P_M(x^{n+1}|\boldsymbol{x}^{1:n}) = \text{softmax}(\boldsymbol{z}^{n+1})$.

KGW (ICML 23 Outstanding)

A Watermark for Large Language Models

John Kirchenbauer * **Jonas Geiping** * **Yuxin Wen** **Jonathan Katz** **Ian Miers** **Tom Goldstein**

University of Maryland

Core idea

Vocab partition based on preceding text.

- **Vocab partition**: in each step, randomly split the vocab $\mathcal{V}$ into two disjoint subsets, the green list $\mathcal{V}_g$ and the red list $\mathcal{V}_r$.
- **Preceding-text-based**: the randomness is seeded by the hash of the preceding text.
- Increase probs for green tokens (tokens in $\mathcal{V}_g$).

Watermark Ironing

In each step of decoding:

(1) compute a hash of $\boldsymbol{x}^{1:n}$: $h^{n+1} = H(\boldsymbol{x}^{1:n}) \cdots H(\cdot)$ can only use the last k tokens $\boldsymbol{x}^{n-k+1:n}$.

(2) seed a random number generator with h^{n+1} and randomly partitions $\mathcal{V}$ into two disjoint lists: the *green* list $\mathcal{V}_g$ and the *red* list $\mathcal{V}_r$,

(3) adjust the logits $\boldsymbol{z}^{n+1}$ by adding a constant bias δ ($\delta > 0$) for tokens in the green list:

$$\forall i \in \{1, 2, \ldots, |\mathcal{V}|\},$$
$$\tilde{\boldsymbol{z}}_i^{n+1} = \begin{cases} \boldsymbol{z}_i^{n+1} + \delta, & \text{if } v_i \in \mathcal{V}_g, \\ \boldsymbol{z}_i^{n+1}, & \text{if } v_i \in \mathcal{V}_r. \end{cases} \tag{1}$$

Watermark Ironing

As a result, watermarked text will statistically contain more *green tokens*, an attribute unlikely to occur in human-written text.

Prompt

…The watermark detection algorithm can be made public, enabling third parties (e.g., social media platforms) to run it themselves, or it can be kept private and run behind an API. We seek a watermark with the following properties:

No watermark

Extremely efficient on average term lengths and word frequencies on synthetic, microamount text (as little as 25 words)
Very small and low-resource key/hash (e.g., 140 bits per key is sufficient for 99.999999999% of the Synthetic Internet

With watermark

- minimal marginal probability for a detection attempt.
- Good speech frequency and energy rate reduction.
- messages indiscernible to humans.
- easy for humans to verify.

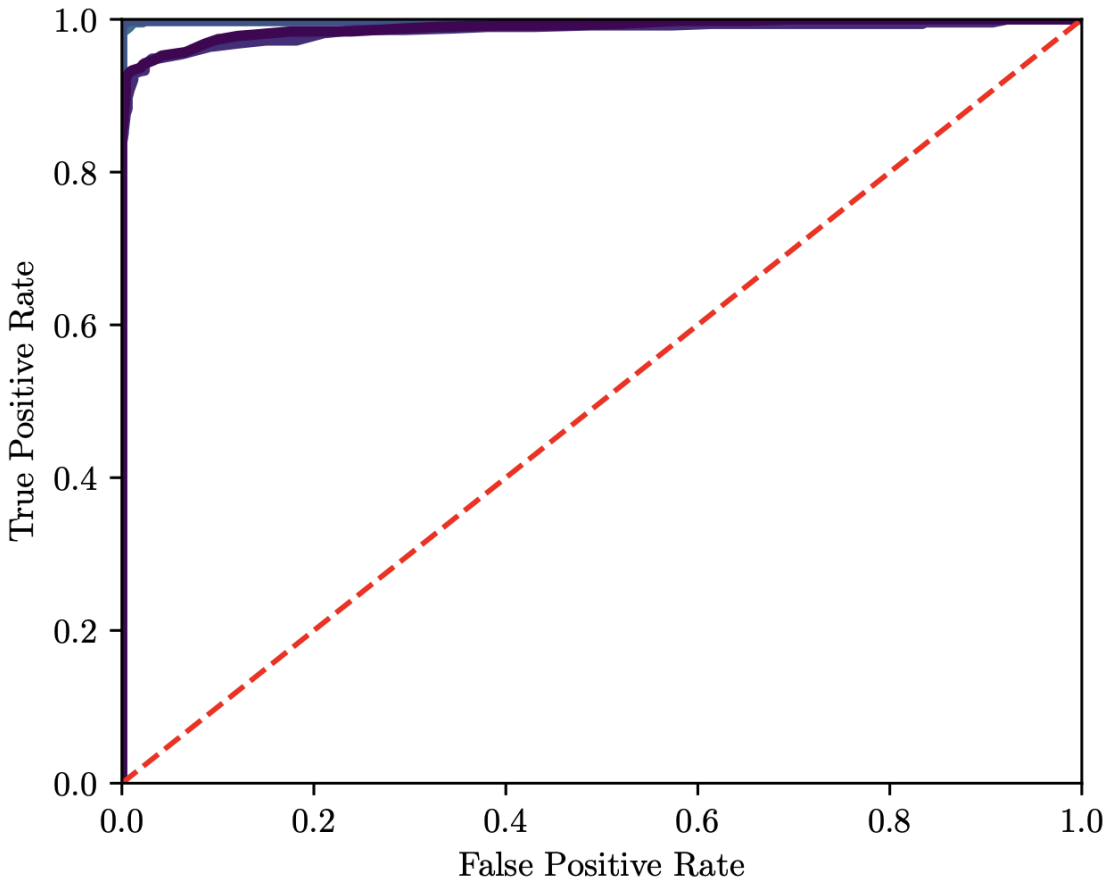
Watermark Detection

When detecting, one can apply step (1) and (2), and calculate the z-score as the watermark strength of $\boldsymbol{x}$:

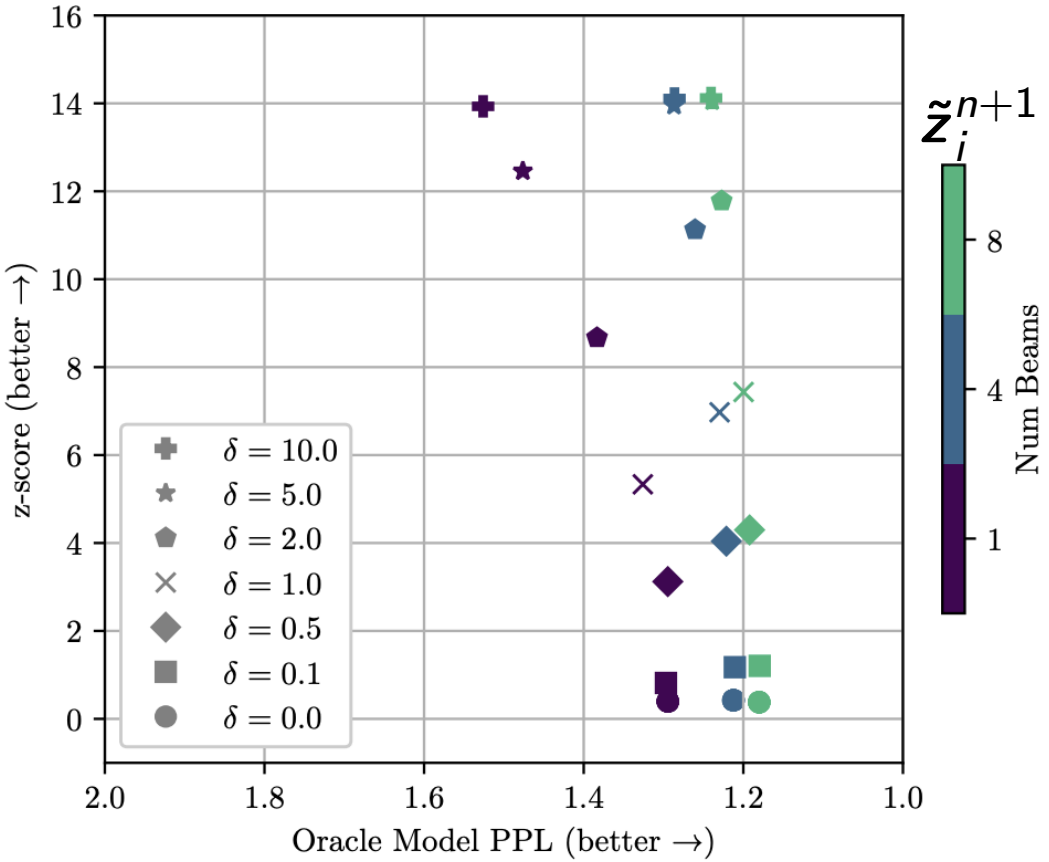
$$s = \frac{(|\boldsymbol{x}|_g - \gamma|\mathcal{V}|)}{\sqrt{|\mathcal{V}|\gamma(1-\gamma)}}, \tag{2}$$

where $|\boldsymbol{x}|_g$ is the number of green tokens in $\boldsymbol{x}$ and $\gamma = \frac{|\mathcal{V}_g|}{|\mathcal{V}|}$. The presence of the watermark can be determined by comparing s with a appropriate threshold.

Performance (ROC Curves — AUC: 0.998)



Text quality



$$\forall i \in \{1, 2, \ldots, |\mathcal{V}|\},$$

$$\tilde{z}_i^{n+1} = \begin{cases} z_i^{n+1} + \delta, & \text{if } v_i \in \mathcal{V}_g, \\ z_i^{n+1}, & \text{if } v_i \in \mathcal{V}_r. \end{cases}$$

Outline

SIR (ICLR 24)

A Semantic Invariant Robust Watermark for Large Language Models

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, Lijie Wen
Tsinghua University
liuaw20@mails.tsinghua.edu.cn, wenlj@tsinghua.edu.cn

Intro

- The robustness of watermarking, i.e., the ability to detect watermarked text even after it has been modified, is important.
- **S**emantic **i**nvariant **r**obust watermark (SIR) is designed to improve the robustness under text re-writing attack.
- Text re-writing attack: modify the wording of the text without changing its semantic, such as re-translation and paraphrase.

A general view of logits adjustment

$$\forall i \in \{1, 2, \ldots, |\mathcal{V}|\},$$

$$\tilde{\boldsymbol{z}}_i^{n+1} = \begin{cases} \boldsymbol{z}_i^{n+1} + \delta, & \text{if } v_i \in \mathcal{V}_g, \\ \boldsymbol{z}_i^{n+1}, & \text{if } v_i \in \mathcal{V}_r. \end{cases}$$

We can view the process of adjusting the logits as applying a Δ function ($\Delta \in \mathbb{R}^{|\mathcal{V}|}$): $\tilde{\boldsymbol{z}}^{n+1} = \boldsymbol{z}^{n+1} + \Delta(\boldsymbol{x}^{1:n})$.

Method

Core idea

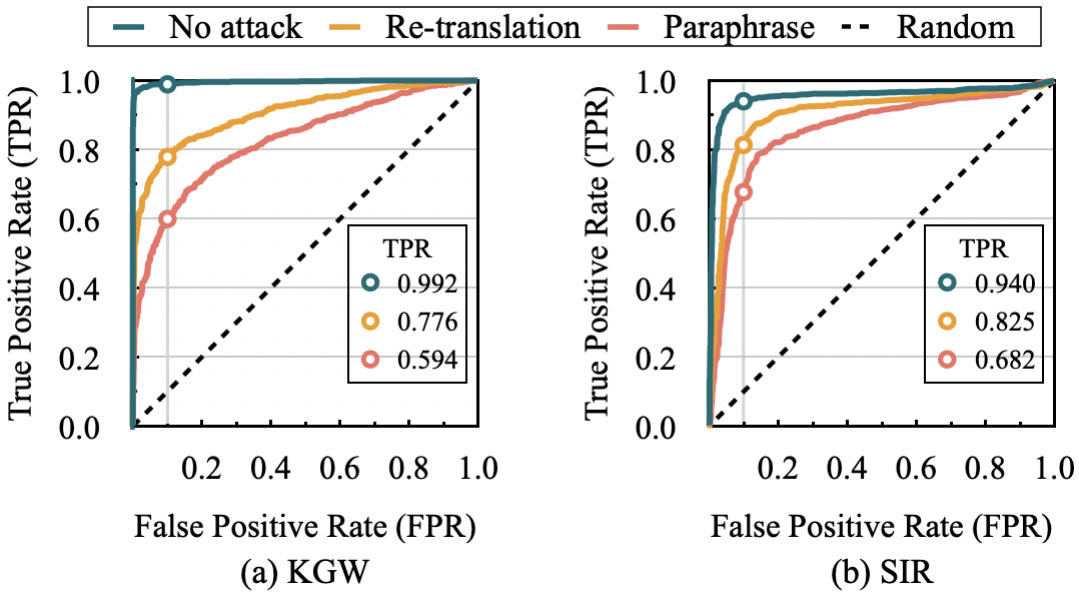
$\text{Sim}(\Delta(\boldsymbol{x}), \Delta(\boldsymbol{y})) \approx \text{Sim}(E(\boldsymbol{x}), E(\boldsymbol{y}))$, where $E(\cdot)$ is an embedding model, and $\text{Sim}(\cdot)$ is a similarity function.

Given an embedding model E, SIR train a Δ function with main objective:

$$\mathcal{L} = |\text{Sim}(E(\boldsymbol{x}), E(\boldsymbol{y})) - \text{Sim}(\Delta(\boldsymbol{x}), \Delta(\boldsymbol{y}))|. \tag{3}$$

Furthermore, $\forall i \in \{1, 2, \ldots, |\mathcal{V}|\}$, Δ_i is trained to be close to $+1$ or -1. Therefore, SIR can be seen as an improvement based on KGW, where $\Delta_i > 0$ indicating that v_i is a green token.

Performance (Our implementation)



Outline

Can Watermarks Survive Translation?

- Existing works on watermark robustness focus mainly on English. However, our world is multilingual.
- What if we translate watermarked text into other language? Can watermarks survive translation?

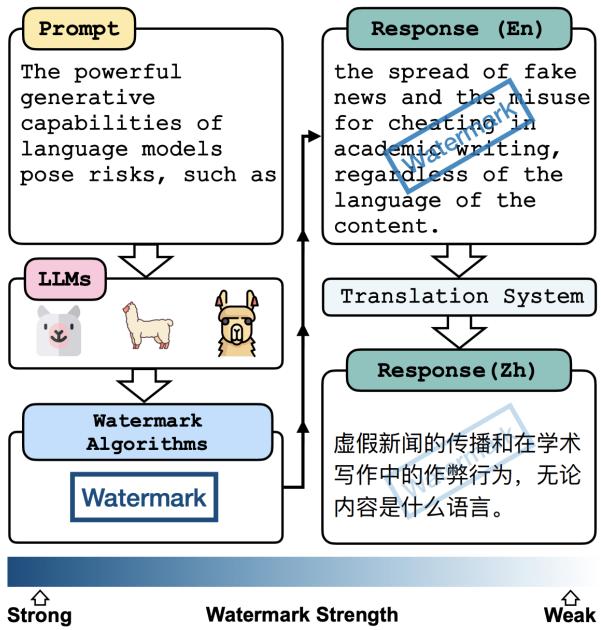


Figure 1: Illustration of watermark dilution in a cross-lingual environment. Best viewed in color.

Outline

Evaluation: cross-lingual consistency of text watermark

- We define *cross-lingual consistency* to assess the ability of text watermarks to maintain their effectiveness after being translated into other languages.
- Given original watermark strength S, and the strength after translation $\hat{S}$:
  - **Pearson Correlation Coefficient (PCC)** captures trend consistency:

$$\mathrm{PCC}(S, \hat{S}) = \frac{\mathrm{cov}(S, \hat{S})}{\sigma_S \sigma_{\hat{S}}}, \tag{4}$$

  - **Relative Error (RE)** captures magnitude consistency

$$\mathrm{RE}(S, \hat{S}) = \mathbb{E}\left[\frac{|\hat{S} - S|}{|S|}\right] \times 100\%.$$

PCCs and REs

Method	PCC ↑					RE (%) ↓				
	En→Zh	En→Ja	En→Fr	En→De	Avg.	En→Zh	En→Ja	En→Fr	En→De	Avg.
BAICHUAN-7B										
KGW	0.108	-0.257	0.059	0.144	0.013	**75.62**	88.50	76.37	**73.65**	**78.54**
UW	0.190	0.087	0.166	0.183	0.156	97.57	98.82	97.22	97.89	97.88
SIR	**0.283**	**0.380**	**0.348**	**0.234**	**0.311**	84.16	**68.28**	**76.07**	93.41	80.41
LLAMA-2-7B-CHAT										
KGW	0.056	**0.177**	**0.276**	0.080	**0.147**	85.57	**79.55**	86.58	92.54	86.06
UW	**0.076**	0.092	0.116	0.109	0.098	92.85	95.40	95.32	96.14	94.93
SIR	-0.106	-0.159	0.146	**0.323**	0.051	**69.52**	92.80	**59.76**	**68.57**	**72.48**

Table 1: Comparison of cross-lingual consistency between different text watermarking methods (KGW, UW, and SIR). **Bold** entries denote the best result among the three methods.

- PCCs are generally less than 0.2, and the REs are predominantly above 80%.
- SIR is slightly better than other methods.

Watermark strength vs Text length

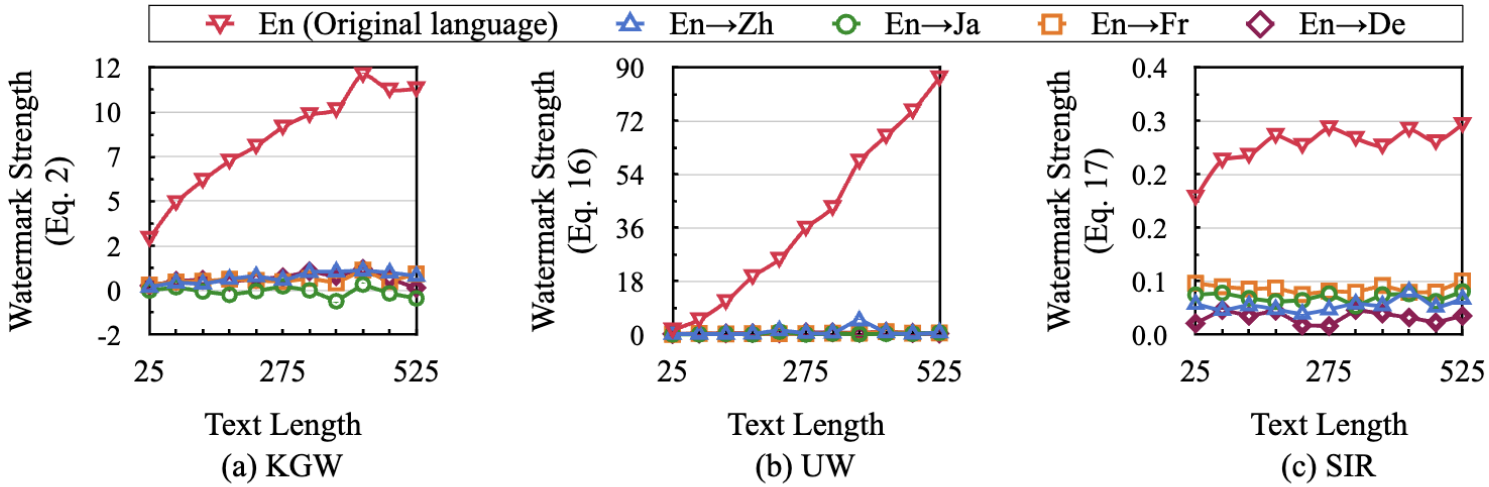


Figure 2: Trends of watermark strengths with text length before and after translation. This is the average result of BAICHUAN-7B and LLAMA-2-7B-CHAT. Figure 7 displays results for each model. Given the distinct calculations for watermark strengths of the three methods, the y-axis scales vary accordingly.

Current text watermarking methods lack cross-lingual consistency.

Outline

Attack: the gaps between real scenarios

- **Language switching**: An attacker who wants to remove the watermark typically do not want to change the language of the response.
- **Text quality**: Translation might effect text quality, but we have not conducted evaluation because we change the language of response in the previous section.

Cross-lingual Watermark Removal Attack (CWRA)

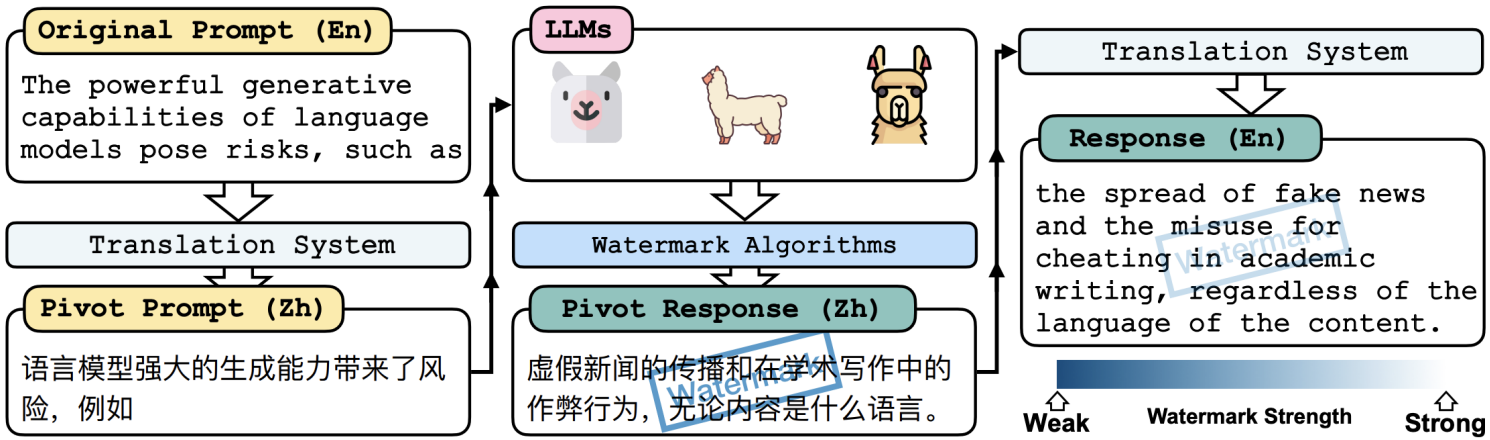


Figure 3: An example pipeline of CWRA with English (En) as the original language and Chinese (Zh) as the pivot language. When performing CWRA, the attacker not only wants to remove the watermark, but also gets a response in the original language with high quality. Its core idea is to wrap the query to the LLM into the pivot language.

Performance: watermark detection

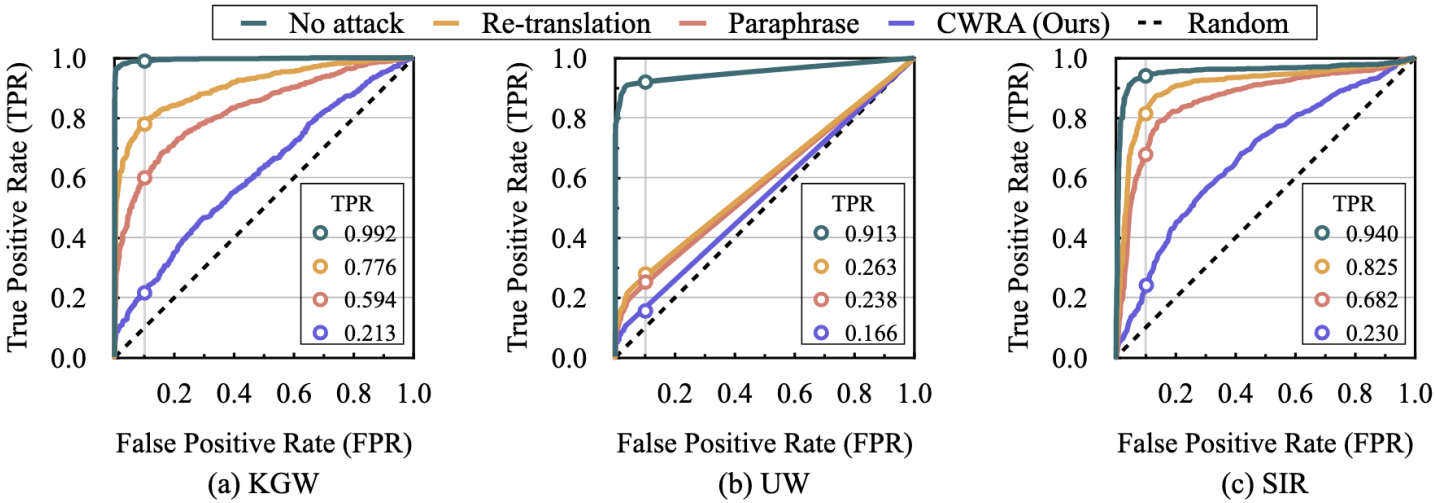


Figure 4: ROC curves for KGW, UW, and SIR under various attack methods: Re-translation, Paraphrase and CWRA. We also present TPR values at a fixed FPR of 0.1. This is the overall result of text summarization and question answering. Figure 8 and Figure 9 display results for each task.

1 2

[1]We fixed the paraphraser and translator used in all methods as `gpt-3.5-turbo-0613`.

[2]The base model is `Baichuan`, supporting English and Chinese.

Performance: text quality

Attack \ WM	KGW			UW			SIR		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Text Summarization									
No attack	14.24	2.68	12.99	13.65	1.68	12.38	13.34	1.79	12.43
Re-translation	14.11	2.43	12.89	13.89	1.77	12.63	13.63	1.98	12.61
Paraphrase	15.10	2.49	13.69	14.72	1.95	13.31	15.56	2.11	14.14
CWRA (Ours)	**18.98**	**3.63**	**17.33**	**15.88**	**2.31**	**14.25**	**17.38**	**2.67**	**15.79**
Question Answering									
No attack	**19.00**	2.18	16.09	11.70	0.49	9.57	16.95	1.35	14.91
Re-translation	18.62	2.32	16.39	12.98	1.30	11.16	16.90	1.80	15.12
Paraphrase	18.45	2.24	**16.47**	14.38	1.37	13.07	17.17	1.79	**15.54**
CWRA (Ours)	18.23	**2.56**	16.27	**15.20**	**1.88**	**13.45**	**17.47**	**2.22**	15.53

Table 2: Comparative analysis of text quality impacted by different watermark removal attacks.

- These attack methods not only preserve text quality, but also bring slight improvements in most cases. This might be attributed to good translators and paraphraser.
- CWRA has the best overall results. We speculate that `Baichuan` performs even better in the pivot language (Chinese) than in the original language (English).

Outline

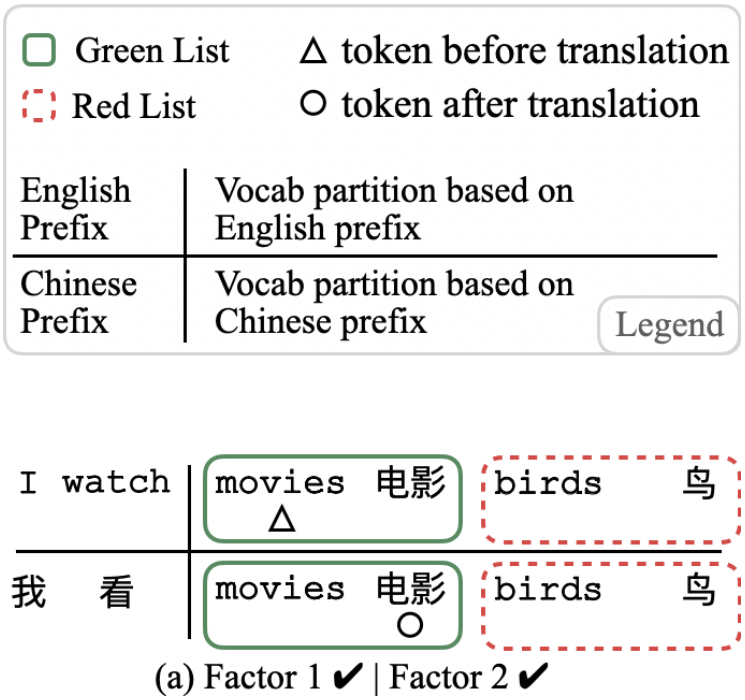
Defence: how to improve cross-lingual consistency?

- KGW-based watermarking methods fundamentally depend on the partition of the vocab, i.e., the red and green lists, as discussed in Section 2.

Cross-lingual consistency

the green tokens in the watermarked text will still be recognized as green tokens after being translated into other languages

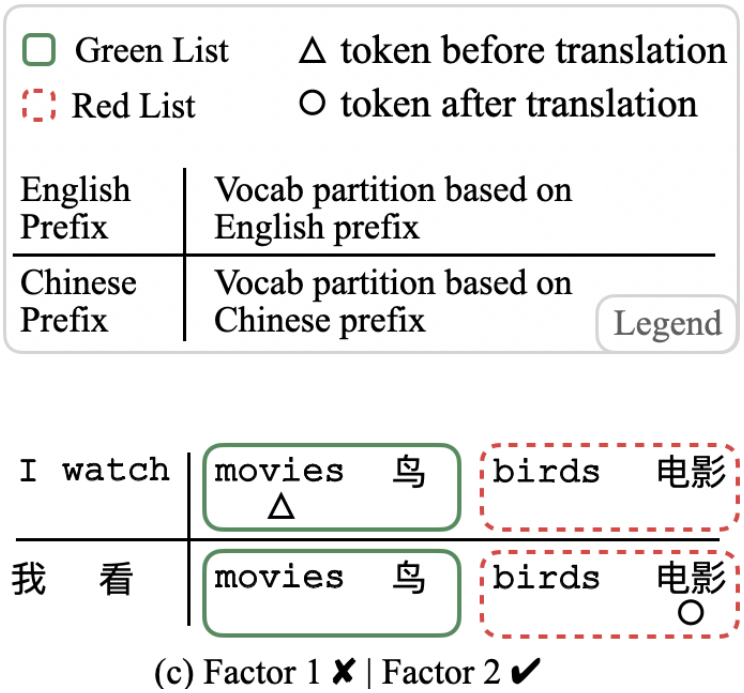
A simplest case study - 1



(a) Factor 1 ✔ | Factor 2 ✔

- ✓**Factor 1**: semantically similar tokens should be in the same list (either red or green)
- ✓**Factor 2**: the vocab partitions for semantically similar prefixes should be the same.

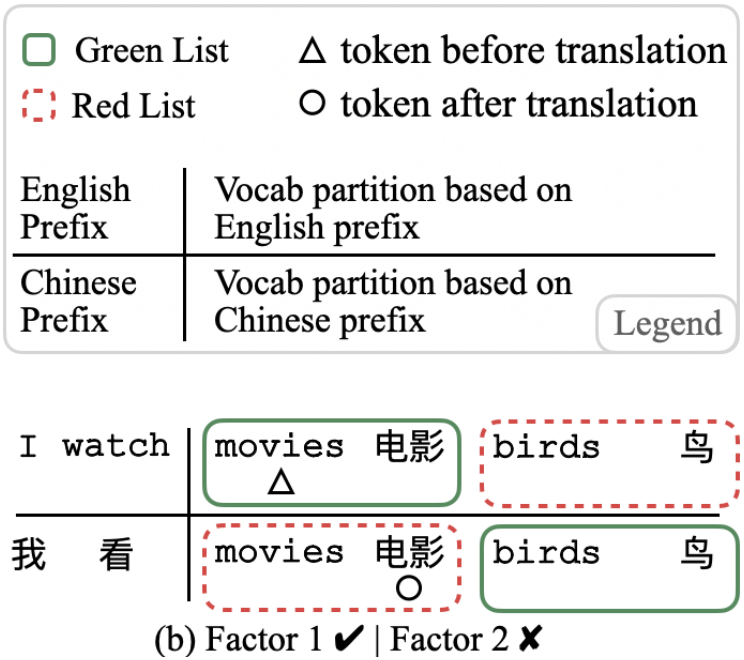
A simplest case study - 2



(c) Factor 1 ✘ | Factor 2 ✔

- ✗**Factor 1**: semantically similar tokens should be in the same list (either red or green)
- ✓**Factor 2**: the vocab partitions for semantically similar prefixes should be the same.

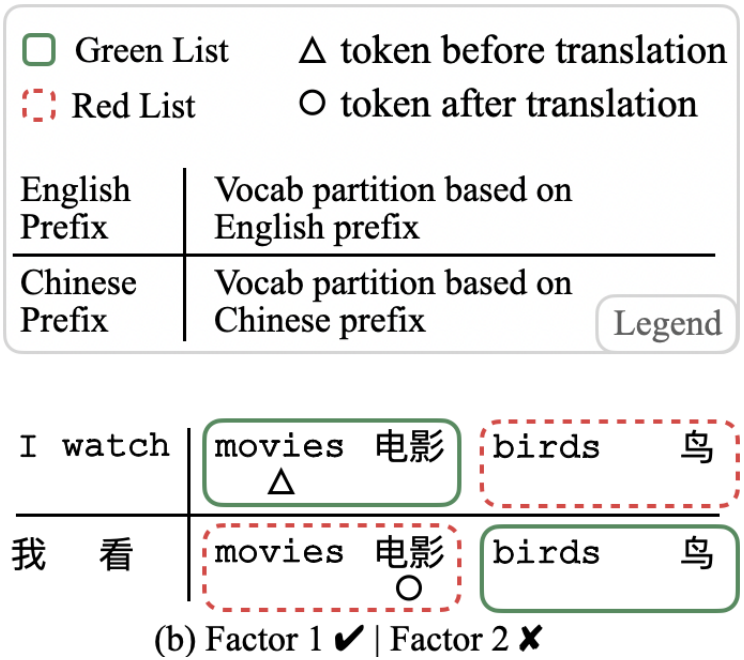
A simplest case study - 3



(b) Factor 1 ✔ | Factor 2 ✘

- ✓**Factor 1**: semantically similar tokens should be in the same list (either red or green)
- ✗**Factor 2**: the vocab partitions for semantically similar prefixes should be the same.

A simplest case study - 3



(b) Factor 1 ✔ | Factor 2 ✘

- ✓**Factor 1**: semantically similar tokens should be in the same list (either red or green)
- ✗**Factor 2**: the vocab partitions for semantically similar prefixes should be the same.

Factor 1 & 2 must be satisfied simultaneously.

Defense Method

SIR has already optimized for **Factor 2** since its objective is:

$$\mathcal{L} = |\text{Sim}(E(\boldsymbol{x}), E(\boldsymbol{y})) - \text{Sim}(\Delta(\boldsymbol{x}), \Delta(\boldsymbol{y}))|. \tag{6}$$

Based on SIR, we discuss how to achieve **Factor 1** and name our method X-SIR.

Defense Method (X-SIR): adapting Δ function

- We define semantic clustering as a partition $\mathcal{C}$ of the vocabulary $\mathcal{V}$:

$$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{|\mathcal{C}|}\}, \tag{7}$$

  where each cluster $\mathcal{C}_i$ consists of semantically equivalent tokens.
- We adapt the Δ function so that it yields biases to each cluster in $\mathcal{C}$, i.e., $\Delta \in \mathbb{R}^{|\mathcal{C}|}$ ~~$(\Delta \in \mathbb{R}^{|\mathcal{V}|})$~~.
- Thus, the process of adjusting the logits should be:

$$\begin{aligned} &\forall i \in \{1, 2, \ldots, |\mathcal{V}|\}, \\ &\tilde{\boldsymbol{z}}_i^{n+1} = \boldsymbol{z}_i^{n+1} + \Delta_{C(i)}, \end{aligned} \tag{8}$$

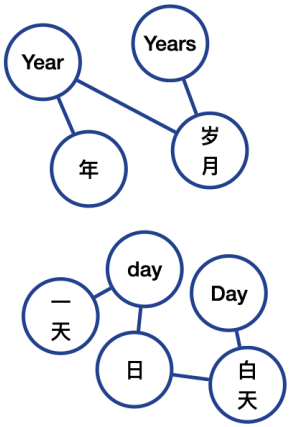
  where $C(i)$ indicates the index of v_i's cluster within $\mathcal{C}$.

Defense Method (X-SIR): semantic clustering of vocab

Algorithm 1 Constructing semantic clusters

Require: A vocabulary $\mathcal{V}$, a bilingual dictionary D
Ensure: Semantic clusters $\mathcal{C}$

Initialize an empty graph G with nodes for each token in $\mathcal{V}$
for each entry (v_i, v_j) in the bilingual dictionary D **do**
    if both v_i and v_j are in $\mathcal{V}$ **then**
        Add an edge (v_i, v_j) to G
    end if
end for
Initialize $\mathcal{C}$ to be an empty set
for each connected component C in G **do**
    Add C to $\mathcal{C}$
end for
return $\mathcal{C}$

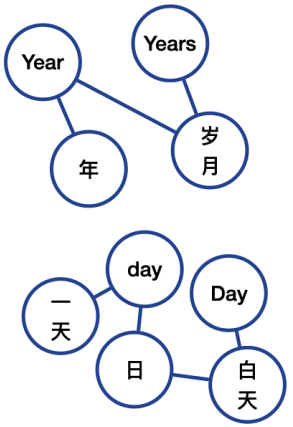


Defense Method (X-SIR): semantic clustering of vocab

Algorithm 2 Constructing semantic clusters

Require: A vocabulary $\mathcal{V}$, a bilingual dictionary D
Ensure: Semantic clusters $\mathcal{C}$

1: Initialize an empty graph G with nodes for each token in $\mathcal{V}$
2: **for** each entry (v_i, v_j) in the bilingual dictionary D **do**
3: **if** both v_i and v_j are in $\mathcal{V}$ **then**
4: Add an edge (v_i, v_j) to G
5: **end if**
6: **end for**
7: Initialize $\mathcal{C}$ to be an empty set
8: **for** each connected component C in G **do**
9: Add C to $\mathcal{C}$
10: **end for**
11: **return** $\mathcal{C}$



- Line 2-3: We only consider tokens shared by $\mathcal{V}$ and D, which results in limitations (discuss later).

Defense Method (X-SIR): semantic clustering of vocab

```
["Years", "Year", "years", "年度", "Year", "year", "_year", "岁月", "years", "年"]
["_Month", "month", "个月", "_month", "月亮", "_months", "_moon", "月", "_Moon", "月份"]
["白天", "day", "_day", "日", "_Day", "一天", "Day"]
["and", "而且", "还有", "_and", "和", "And", "_And"]
["农村", "_village", "村庄", "_Rural", "_Village", "_villages", "乡村", "_rural", "村"]
["_men", "男人", "人们", "人民", "_male", "男", "Man", "_Man", "_People", "_Male", "People", "men", "_Men", "男子",
"人", "男性", "_man", "_people", "people", "Men", "_males", "man"]
["大", "_Big", "_big", "Big", "big"]
["他", "he", "_He", "He", "_he"]
["德", "_Tak"]
["_heavy", "重", "_Heavy"]
["_one", "one", "One", "一", "_One", "一个"]
["方向", "_direction", "定向"]
["但", "But", "_but", "but", "不过", "但是", "_But"]
```

- We also consider the meta symbol (U+2581) for sentencepiece.

Performance: watermark detection

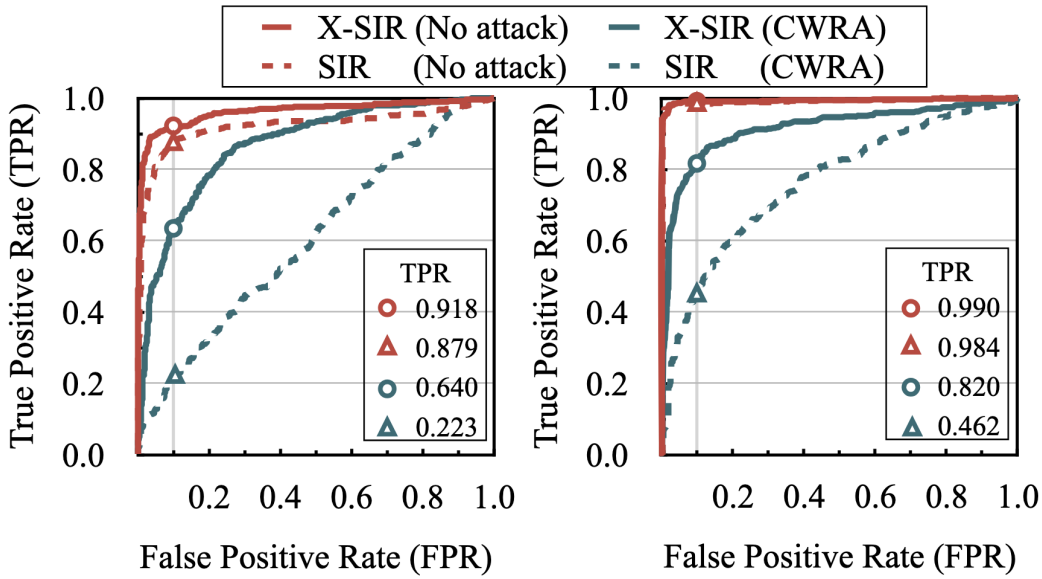


Figure 6: ROC curves of X-SIR and SIR.

Performance: text quality

Method	ROUGE-1	ROUGE-2	ROUGE-L
Text Summarization			
SIR	13.34	1.79	12.43
X-SIR	**15.65**	**2.04**	**14.29**
Question Answering			
SIR	**16.95**	1.35	**14.91**
X-SIR	16.77	**1.39**	14.07

Table 4: Effects of X-SIR and SIR on text quality.

Limitations

Semantic clustering only considers tokens shared by the vocab $\mathcal{V}$ of model and external dictionary D, which results in the following limitations.

- **Language coverage**: only support language supported by the model. In a real scenario, the attacker can choose the original language and the pivot language at will.
- **Vocab coverage (20.76%)**: since external dictionary D only contains whole words, words units can not be clustered. Llama tokenizer tends to split a Chinese char into multiple bytes.

Paper & Code

https://arxiv.org/abs/2402.14007

https://github.com/zwhe99/X-SIR