On the Cross-lingual Consistency of Text Watermark for Large Language Models

Anonymous ACL submission

Abstract

Text watermarking technology aims to tag and identify content produced by large language models (LLMs) to prevent misuse. In this study, we introduce the concept of "*cross-lingual consistency*" in text watermarking, which assesses the ability of text watermarks to maintain their effectiveness across different languages. Preliminary empirical results from two LLMs and three watermarking methods reveal that current text watermarking technologies lack consistency when texts are translated into various languages. Based on this observation, we propose a Cross-lingual Watermark Removal Attack (CWRA) to bypass watermarking by first obtaining a response from an LLM in a pivot language, which is then translated into the target language. CWRA can effectively remove watermarks by reducing the Area Under the Curve (AUC) from 0.95 to 0.67 without performance loss. Furthermore, we analyze two key factors that contribute to the cross-lingual consistency in text watermarking and propose a defense method that increases the AUC from 0.67 to 0.88 under CWRA.

1 Introduction

Large language models (LLMs) like GPT-4 (OpenAI, 2023) and Gemini (Team, 2023) have demonstrated remarkable content generation capabilities, producing texts that are hard to distinguish from human-written ones. This progress has led to concerns regarding the misuse of LLMs, such as the risks of generating misleading information, impersonating individuals, and compromising academic integrity (Chen and Shu, 2023a,b). As a countermeasure, text watermarking technology for LLMs has been developed, aiming at tagging and identifying the content produced by LLMs (Kirchenbauer et al., 2023a; Liu et al., 2023b,c). Generally, a text watermarking algorithm embeds a message within LLM-generated content that is imperceptible to human readers, but can be detected algorithmically.

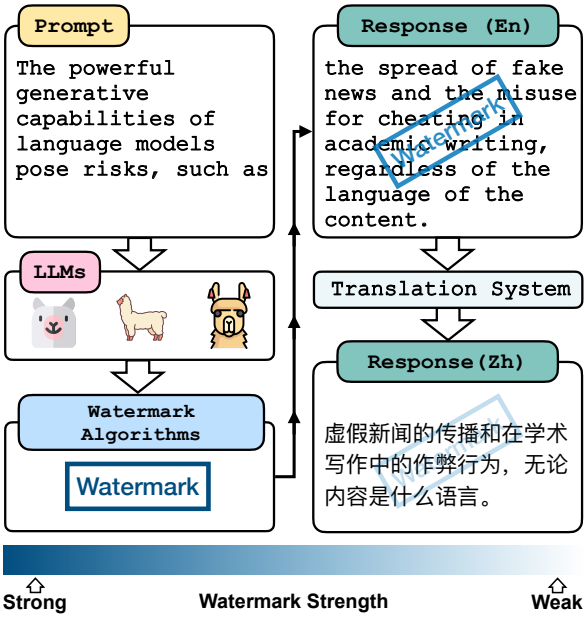


Figure 1: Illustration of watermark dilution in a cross-lingual environment. Best viewed in color.

By tracking and detecting text watermarks, it becomes possible to mitigate the abuse of LLMs by tracing the origin of texts and ascertaining their authenticity.

The robustness of watermarking algorithms, i.e., the ability to detect watermarked text even after it has been modified, is important. Recent works have shown strong robustness under text rewriting and copy-paste attacks (Liu et al., 2023b; Yang et al., 2023). However, these watermarking techniques have been tested solely within monolingual contexts. In practical scenarios, watermarked texts might be translated, raising questions about the efficacy of text watermarks across languages (see Figure 1). For example, a malicious user could use a watermarked LLM to produce fake news in English and then translate it into Chinese. Obviously, the deceptive impact persists regardless of the language, but it is uncertain whether the watermark would still be detectable after such a trans-

lation. To explore this question, we introduce the concept of *cross-lingual consistency* in text watermarking, aiming to characterize the ability of text watermarks to preserve their strength across languages. Our preliminary results on 2 LLMs $\times$ 3 watermarks reveal that current text watermarking technologies lack consistency across languages.

In light of this finding, we propose the **C**ross-lingual **W**atermark **R**emoval **A**ttack (**CWRA**) to highlight the practical implications arising from deficient cross-lingual consistency. When performing CWRA, the attacker begins by translating the original language prompt into a pivot language, which is fed to the LLM to generate a response in the pivot language. Finally, the response is translated back into the original language. In this way, the attacker obtains the response in the original language and bypasses the watermark with the second translation step. CWRA outperforms re-writing attacks, such as re-translation and paraphrasing (Liu et al., 2023c), as it achieves the lowest AUC (reducing it from 0.95 to 0.67) and the highest text quality.

To resist CWRA, we propose a defense method that improves the cross-lingual consistency of current LLM watermarking. Our method is based on two critical factors. The first is the **cross-lingual semantic clustering of the vocabulary**. Instead of treating each token in the vocabulary as the smallest unit when ironing watermarks, as done by KGW (Kirchenbauer et al., 2023a), our method considers a cluster of tokens that share the same semantics across different languages as the smallest unit of processing. In this way, the post-translated token will still carry the watermark as it would fall in the same cluster as before translation. The second is **cross-lingual semantic robust vocabulary partition**. Inspired by Liu et al. (2023b), we ensure that the partition of the vocabulary are similar for semantically similar contexts in different languages. Despite its limitations, our approach elevates the AUC from 0.67 to 0.88 under the CWRA, paving the way for future research.

Our contributions are summarized as follows:

- We reveal the deficiency of current text watermarking technologies in maintaining cross-lingual consistency (§ 3).
- Based on this finding, we propose CWRA that successfully bypasses watermarks without degrading the text quality (§ 4).
- Based on our analysis of the two key factors for improving cross-lingual consistency, we propose a defense method against CWRA (§ 5).

2 Background

2.1 Language Model

A language model (LM) M has a defined set of tokens known as its vocabulary $\mathcal{V}$. Given a sequence of tokens $\boldsymbol{x}^{1:n} = (x^1, x^2, \ldots, x^n)$, which we refer to as the *prompt*, the model M computes the conditional probability of the next token over $\mathcal{V}$ as $P_M(x^{n+1}|\boldsymbol{x}^{1:n})$. Therefore, text generation can be achieved through an autoregressive decoding process, where M sequentially predicts one token at a time, forming a *response*. Such an LM can be parameterized by a neural network, such as Transformer (Vaswani et al., 2017), which is called neural LM. Typically, a neural LM computes a vector of logits $\boldsymbol{z}^{n+1} = M(\boldsymbol{x}^{1:n}) \in \mathbb{R}^{|\mathcal{V}|}$ for the next token based on the current sequence $\boldsymbol{x}_{1:n}$ via a neural network. The probability of the next token is then obtained by applying the softmax function to these logits: $P_M(x^{n+1}|\boldsymbol{x}^{1:n}) = \text{softmax}(\boldsymbol{z}^{n+1})$.

2.2 Watermarking for LMs

In this work, we consider the following watermarking methods. All of them embed the watermark by modifying logits during text generation and detect the presence of the watermark for any given text.

KGW (Kirchenbauer et al., 2023a) sets the groundwork for LM watermarking. Ironing a watermark is delineated as the following steps:

(1) compute a hash of $\boldsymbol{x}^{1:n}$: $h^{n+1} = H(\boldsymbol{x}^{1:n})$,
(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:

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

As a result, watermarked text will statistically contain more *green tokens*, an attribute unlikely to occur in human-written text. When detecting, one can apply step (1) and (2), and calculate the z-score as the watermark strength of $\boldsymbol{x}$:

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

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 threshold.

Unbiased watermark (**UW**) views the process of adjusting the logits as applying a Δ function:

$\tilde{\boldsymbol{z}}^{n+1} = \boldsymbol{z}^{n+1} + \Delta$, and designs a Δ function that satisfies:

$$\mathbb{E}\left[\tilde{P}_M\right] = P_M, \tag{3}$$

where $\tilde{P}_M$ is the probability distribution of the next token after logits adjustment (Hu et al., 2023).

Semantic invariant robust watermark (SIR) shows the robustness under re-translation and paraphrasing attack (Liu et al., 2023b). Its core idea is to assign similar Δ for semantically similar prefixes. Given prefix sequences $\boldsymbol{x}$ and $\boldsymbol{y}$, SIR adopts an embedding model E to characterize their semantic similarity and trains a watermark model that yields Δ with the main objective:

$$\mathcal{L} = |\text{Sim}(E(\boldsymbol{x}), E(\boldsymbol{y})) - \text{Sim}(\Delta(\boldsymbol{x}), \Delta(\boldsymbol{y}))|, \tag{4}$$

where $\text{Sim}(\cdot,\cdot)$ denotes similarity function. 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. The original implementation of SIR uses C-BERT (Chanchani and Huang, 2023) as the embedding model, which is English-only. To adopt SIR in the cross-lingual scenario, we use a multilingual S-BERT (Reimers and Gurevych, 2019)[1] instead.

3 Cross-lingual Consistency of Text Watermark

In this section, we define the concept of cross-lingual consistency in text watermarking and answer three research questions (RQ):

- **RQ1**: To what extent are current watermarking algorithms consistent across different languages?
- **RQ2**: Do watermarks exhibit better consistency between similar languages than between distant languages?
- **RQ3**: Does semantic invariant watermark (SIR) exhibit better cross-lingual consistency than others (KGW and UW)?

3.1 Definition

We define cross-lingual consistency as the ability of a watermark, embedded in a text produced by an LLM, to retain its strength after the text is translated into another language. We represent the original strength of the watermark as a random variable, denoted by S (Appendix A.1), and its strength after translation as $\hat{S}$. To quantitatively assess this consistency, we employ the following two metrics.

[1] paraphrase-multilingual-mpnet-base-v2

Pearson Correlation Coefficient (PCC) We use PCC to assess linear correlation between S and $\hat{S}$:

$$\text{PCC}(S, \hat{S}) = \frac{\text{cov}(S, \hat{S})}{\sigma_S \sigma_{\hat{S}}}, \tag{5}$$

where $\text{cov}(S, \hat{S})$ is the covariance and σ_S and $\sigma_{\hat{S}}$ are the standard deviations. A PCC value close to 1 suggests consistent trends in watermark strengths across languages.

Relative Error (RE) Unlike PCC, which captures consistency in trends, RE is used to assess the magnitude of deviation between S and $\hat{S}$:

$$\text{RE}(S, \hat{S}) = \mathbb{E}\left[\frac{|\hat{S} - S|}{|S|}\right] \times 100\%. \tag{6}$$

A lower RE indicates that the watermark retains strength close to its original value after translation, signifying great cross-lingual consistency. To avoid the instability caused by values of S that are close to 0, we first aggregate the data by text length and replace the original values of S and $\hat{S}$ with their respective mean values within each group. We also apply min-max normalization to ensure that all values are non-negative.

3.2 Experimental Setup

Setup We sampled a subset of 1,000 prompts from the UltraChat test set (Ding et al., 2023)[2], and generated responses from the LLM using the text watermarking methods described in § 2.2. The default decoding method was multinomial sampling, and both the prompts and the LLM-generated responses were in English. To evaluate the cross-lingual consistency, these watermarked responses were translated into four languages using `gpt-3.5-turbo-0613`[3]: Chinese (Zh), Japanese (Ja), French (Fr), and German (De). Notably, English shares greater similarities with French and German, in contrast to its significant differences from Chinese and Japanese.

Models For the LLMs, we adopt:

- BAICHUAN-7B (Baichuan., 2023): an LLM trained on 1.2 trillion tokens. It offers bilingual support for both Chinese and English.
- LLAMA-2-7B-CHAT (Touvron et al., 2023): trained on 2 trillion tokens and only provides support for English.

[2] https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k

[3] https://platform.openai.com/docs/models

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.

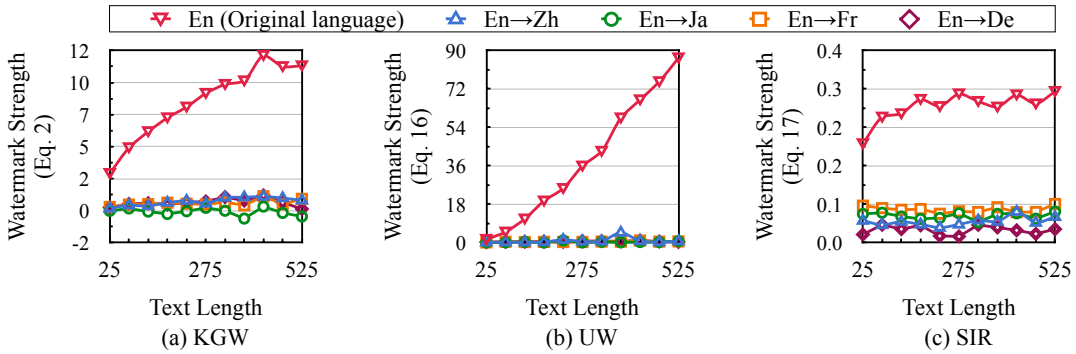


Figure 2: Trends of watermark strengths with text length before and after translation. This is the average result of BAICHUAN-7B and LLAMA-2-7B-CHAT. Figure 7 displays results for each model. Given the distinct calculations for watermark strengths of the three methods, the y-axis scales vary accordingly.

3.3 Results

Table 1 presents the results, and Figure 2 illustrates the trend of watermark strengths with text length.

Results for RQ1 We reveal a notable deficiency in the cross-lingual consistency of current watermarking methods. Among all the settings, the PCCs are generally less than 0.3, and the REs are predominantly above 80%. Furthermore, Figure 2 visually demonstrates that the watermark strengths of the three methods exhibit a significant decrease after translation. These results suggest that current watermarking algorithms struggle to maintain effectiveness across language translations.

Results for RQ2 Only SIR exhibits such a characteristic: when using LLAMA-2-7B-CHAT, its cross-lingual consistency performs notably better among similar languages compared to distant ones, especially in terms of PCC. This can be attributed to its semantic invariance and shared words among similar languages. However, this characteristic is not shown on BAICHUAN-7B, which might be related to tokenization. In contrast, this property is less evident in the case of KGW and UW.

Results for RQ3 Overall, SIR indeed exhibits superior cross-lingual consistency compared to KGW and UW. Particularly when using BAICHUAN-7B, SIR achieves the best PCCs for all target languages. When using LLAMA-2-7B-CHAT, SIR still performs well in languages similar to English. This finding highlights the importance of semantic invariance in preserving watermark strength across languages, which we will explore more in § 5. Despite its superiority, SIR still presents a notable reduction in watermark strength in cross-lingual scenarios, as evidenced by Figure 2e.

4 Cross-lingual Watermark Removal Attack

In the previous section, we focus on scenarios where the response of LLM is translated into other languages. However, an attacker typically expects a response from the LLM in the same language as the prompt while removing watermarks. To bridge this gap, we introduce the Cross-lingual Watermark Removal Attack (CWRA) in this section, constituting a complete attack process and posing a more significant challenge to text watermarking than paraphrasing and re-translation attacks.

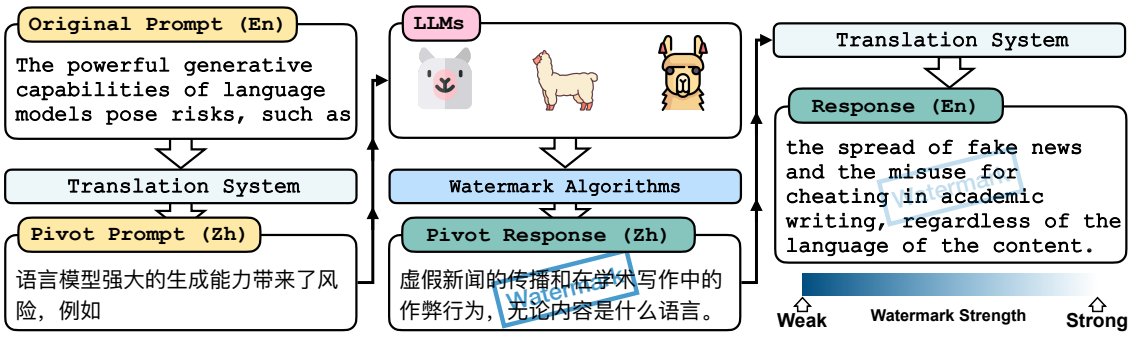


Figure 3: An example pipeline of CWRA with English (En) as the original language and Chinese (Zh) as the pivot language. When performing CWRA, the attacker not only wants to remove the watermark, but also gets a response in the original language with high quality. Its core idea is to wrap the query to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

Figure 3 shows the process of CWRA. Instead of feeding the original prompt into the LLM, the attacker initiates the attack by translating the prompt into a pivot language named the pivot prompt. The LLM receives the pivot prompt and provides a watermarked response in the pivot language. The attacker then translates the pivot response back into the original language. This approach allows the attacker to obtain the response in the original language. Due to the inherent challenges in maintaining cross-lingual consistency, the watermark would be effectively eliminated during the second translation step.

4.1 Setup

To assess the practicality of attack methods, we consider two downstream tasks: text summarization and question answering. We adopt MultiNews (Fabbri et al., 2019) and ELI5 (Fan et al., 2019) as test sets, respectively. Both datasets are in English and require long text output with an average output length of 198 tokens. We selected 500 samples for each test set that do not exceed the maximum context length of the model and performed zero-shot prompting on BAICHUAN-7B. For CWRA, we select Chinese as the pivot language and compare the following two methods:

- **Paraphrase**: rephrasing the response into different wording while retaining the same meaning.
- **Re-translation**: translating the response into the pivot language and back to the original language.

The paraphraser and translator used in all attack methods are `gpt-3.5-turbo-0613` to ensure consistency across the different attack methods.

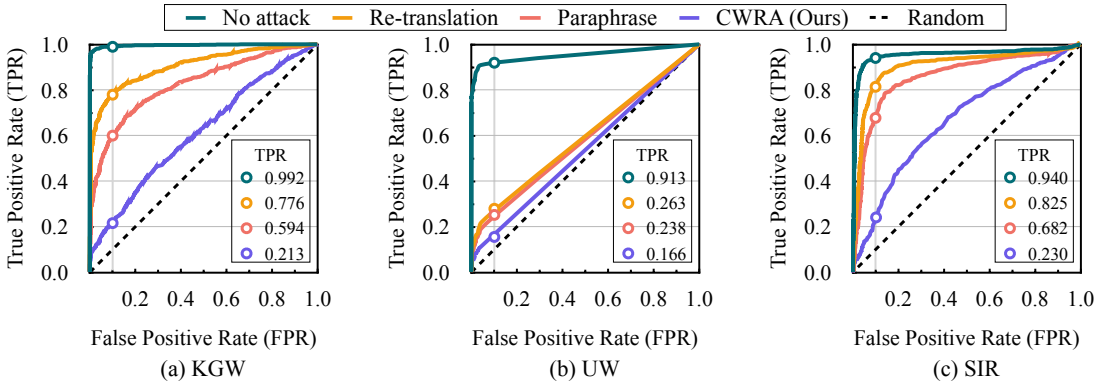
4.2 Results

Figure 4 exhibits ROC curves of three watermarking methods under different attack methods.

CWRA vs Other Attack Methods CWRA demonstrates the most effective attack performance, significantly diminishing the AUC and the TPR. For one thing, existing watermarking techniques are not designed for cross-lingual contexts, leading to weak cross-lingual consistency. For another thing, strategies such as Re-translation and Paraphrase are essentially semantic-preserving text rewriting. Such strategies tend to preserve some n-grams from the original response, which may still be identifiable by the watermark detection algorithm. In contrast, CWRA reduces such n-grams due to language switching.

SIR vs Other Watermarking Methods Under the CWRA, SIR exhibits superior robustness compared to other watermarking methods. The AUCs for KGW and UW under CWRA plummet to 0.61 and 0.54, respectively, approaching the level of random guessing. In stark contrast, the AUC for the SIR method stands significantly higher at 0.67, aligning with our earlier observations regarding cross-lingual consistency in the RQ3 of § 3.3.

Text Quality As shown in Table 2, these attack methods not only preserve text quality, but also bring slight improvements in most cases due to the good translator and paraphraser. Among the compared methods, CWRA stands out for its superior performance. Considering that the same translator and paraphraser were used across all methods, we speculate that this is because the BAICHUAN-7B model used in our experiments performs even better in the pivot language (Chinese) than in the original language (English). This finding implies that a potential attacker could strategically choose a pivot language at which the LLM excels to perform CWRA, thereby achieving the best text quality while removing the watermark.



(a) KGW (b) UW (c) SIR

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.

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.

5 Improving Cross-lingual Consistency

Up to this point, we have observed the challenges associated with text watermarking in cross-lingual scenarios. In this section, we first analyze two key factors essential for achieving cross-lingual consistency. Based on our analysis, we propose a defense method against CWRA.

5.1 Two Key Factors of Cross-lingual Consistency

KGW-based watermarking methods fundamentally depend on the partition of the vocabulary, i.e., the red and green lists, as discussed in § 2.2. Therefore, cross-lingual consistency aims to ensure that the green tokens in the watermarked text will still be recognized as green tokens after being translated into other languages.

With this goal in mind, we start our analysis with a simple English-Chinese case in Figure 5:

1. Given "`I watch`" as the prefix, a watermarked LM predicts "`movies`" as the next token. Due to watermarking, "`movies`" is a green token. All candidate tokens are [`movies`, `birds`, 电影, 鸟], where "`movies`" and "电影" are semantically equivalent, as are "`birds`" and "鸟".
2. A machine translator (MT) then translates the entire sentence "`I watch movies`" into Chinese: "我 看 电影".

The question of interest is: what conditions must the vocabulary partition meets so that the token "电影", the Chinese equivalent of "`movies`", also falls within the green list?

Figure 5(a) illustrates a successful case, where two key factors exists:

1. **Cross-lingual semantic clustering of the vocabulary**: semantically similar tokens must be in the same partition, either green or red lists.
2. **Cross-lingual semantic robust vocabulary partition**: for semantically similar prefixes in different languages: "`I watch`" and "我 看", the partitions of the vocabulary are the same.

Both Figure 5(b) and Figure 5(c) satisfy only one of the two factors, thus failing to recognize "电影" as a green token and losing cross-lingual consistency.

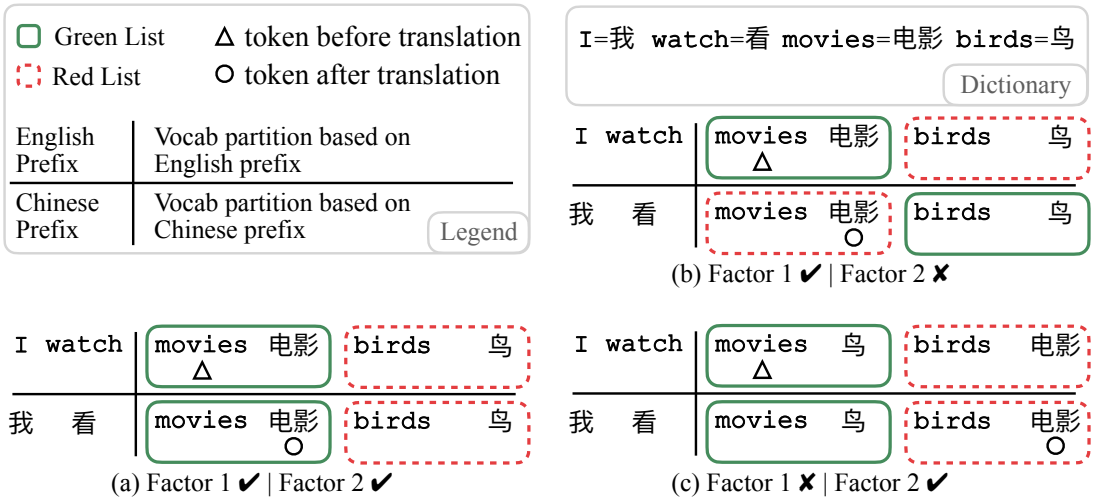


Figure 5: Cases in English-Chinese to maintain cross-lingual consistency: only when both the circle (○) and the triangle (△) symbols are in the green list can cross-lingual consistency be achieved. **Factor 1**: semantically similar tokens should be in the same list (either red or green). In these cases, "movies" and "电影" are semantically equivalent, as are "birds" and "鸟". **Factor 2**: the vocabulary partitions for semantically similar prefixes should be the same. In these cases, all prefixes are semantically equivalent.

5.2 Defense Method against CWRA

We now improve the SIR so that it satisfies the two factors described above. As discussed in § 2.2, SIR uses the Δ function to represent vocabulary partition ($\Delta \in \mathbb{R}^{|\mathcal{V}|}$), where $\Delta_i > 0$ indicating that v_i is a green token. Based on Eq. 4, SIR has already optimized for Factor 2 when using a multilingual embedding model. For prefixes $\boldsymbol{x}$ and $\boldsymbol{y}$, the similarity of their vocabulary partitions for the next token should be close to their semantic similarity:

$$\mathrm{Sim}(\Delta(\boldsymbol{x}), \Delta(\boldsymbol{y})) \approx \mathrm{Sim}(E(\boldsymbol{x}), E(\boldsymbol{y})), \quad (7)$$

where E is a multilingual embedding model.

Based on SIR, we focus on Factor 1, i.e., cross-lingual semantic clustering of the vocabulary. Formally, we define semantic clustering as a partition $\mathcal{C}$ of the vocabulary $\mathcal{V}$: $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{|\mathcal{C}|}\}$, where each cluster $\mathcal{C}_i$ consists of semantically equivalent tokens. Instead of assigning biases for each token in $\mathcal{V}$, we adapt the Δ function so that it yields biases to each cluster in $\mathcal{C}$, i.e., $\Delta \in \mathbb{R}^{|\mathcal{C}|}$. Thus, the process of adjusting the logits should be:

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

where $C(i)$ indicates the index of v_i's cluster within $\mathcal{C}$. By doing so, if token v_i and v_j are semantically equivalent, they will receive the same bias on logits:

$$C(i) = C(j) \implies \Delta_{C(i)} = \Delta_{C(j)}, \quad (9)$$

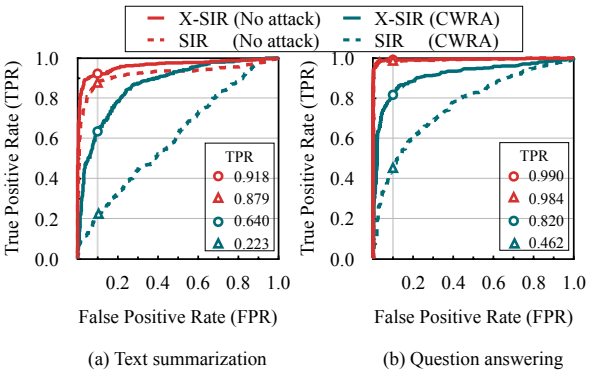
i.e., if v_i and v_j are translations of each other, they will fall into the same list. To obtain such a semantic clustering $\mathcal{C}$, we treat each token in $\mathcal{V}$ as a node, and add an edge (v_i, v_j) whenever (v_i, v_j) corresponds to an entry in a bilingual dictionary. Therefore, $\mathcal{C}$ is all the connected components of this graph.

We name this method as X-SIR and evaluated it under the same setting as § 4. We also detail its limitations in § 8.

Method	Text Summ.		Question Ans.	
	PCC↑	RE (%) ↓	PCC↑	RE (%) ↓
SIR	0.431	66.18	0.321	71.42
X-SIR	**0.554**	**43.49**	**0.507**	**34.98**

Table 3: Cross-lingual consistencies in terms of PCC and RE under CWRA.

Cross-lingual consistency & ROC curvers Table 3 shows the cross-lingual consistency of SIR and X-SIR when confronted CWRA. X-SIR achieves significant improvements in terms of PPC and RE in both tasks. Consequently, as depicted in Figure 6, X-SIR substantially enhances the AUC under CWRA, with an increase in TPRs by ~0.4. Furthermore, X-SIR delivers performance on par with SIR in the absence of any attacks. These findings validate the two key factors of cross-lingual consistency that we identified in § 5.1.



(a) Text summarization (b) Question answering

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

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.

Text quality As shown in Table 4, X-SIR achieves better text quality than SIR in text summarization, and comparable performance on question answering, meaning the semantic partition of vocabulary will not negatively affect text quality.

6 Related Work

6.1 LLM Watermarking

Text watermarking aims to embed a watermark into a text and detect the watermark for any given text. Currently, text watermark method can be classified into two categories (Liu et al., 2023c): watermarking for exsiting text and watermarking for generated text. In this work, we focus on the latter, which is more challenging and has more practical applications.

This type of watermark method usually can be illustrated as the watermark ironing process (modifying the logits of the LLM during text generation) and watermark detection process (assess the presence of watermark by a calculated watermark strength score). Kirchenbauer et al. (2023a) introduces KGW, the first watermarking method for LLMs. Hu et al. (2023) proposes UW without affecting the output probability distribution compared to KGW. Liu et al. (2023b) introduces SIR, a watermarking method taking into account the semantic information of the text, which shows robustness to paraphrase attacks. Liu et al. (2023a) proposes the first unforgeable publicly verifiable watermarking algorithm for large language models. SemStamp (Hou et al., 2023) is another semantic-related watermarking method and it generate watermarked text at sentence granularity instead of token granularity. Tu et al. (2023) introduces WaterBench, the first comprehensive benchmark for LLM watermarks. We introduce the details of KGW, UW and SIR in § 2.2.

6.2 Watermark Robustness

A good watermarking method should be robust to various watermarking removal attacks. However, current works on watermarking robustness mainly focus on single-language attacks, such as paraphrase attacks. For example, Kirchenbauer et al. (2023b) evaluates the robustness of KGW against paraphrase attacks as well as copy-paste attacks and proposes a detect trick to improve the robustness to copy-paste attacks. Zhao et al. (2023) employs a fixed green list to improve the robustness of KGW against paraphrase attacks and editing attacks. Chen et al. (2023) proposes a new paraphrase robust watermarking method "XMark" based on "text redundancy" of text watermark.

7 Conclusion

This work aims to investigate the cross-lingual consistency of watermarking methods for LLMs. We first characterize and evaluate the cross-lingual consistency of current watermarking techniques for LLMs, revealing that current watermarking methods struggle to maintain their watermark strengths across different languages. Based on this observation, we propose the cross-lingual watermark removal attack (CWRA), which significantly challenges watermark robustness by efficiently eliminating watermarks without compromising performance. Through the analysis of two primary factors that influence cross-lingual consistency, we propose X-SIR as a defense strategy against CWRA. Despite its limitations, this approach greatly improves the AUC and paves the way for future research. Overall, this work completes a closed loop in the study of cross-lingual consistency in watermarking, including: evaluation, attacking, analysis, and defensing.

8 Limitations

While X-SIR has demonstrated promising capabilities in enhancing cross-lingual consistency and defending against CWRA, it also faces certain limitations. A key limitation is its narrow scope of applicability. This limitation stems from the reliance on an external bilingual dictionary for semantic clustering of the vocabulary, which means X-SIR focuses solely on whole words and neglects smaller word units. Consequently, its effectiveness is closely tied to the tokenization. X-SIR's performance may be compromised if the tokenizer favors finer-grained token segmentation. Moreover, X-SIR could face difficulties in scenarios with a significant difference in word order between the original and pivot languages. This aspect is crucial, as attackers can exploit these differences in any language pair to conduct CWRA. Therefore, X-SIR does not solve the issue of cross-lingual consistency but sets the stage for future research.

References

Baichuan. 2023. A large-scale 7b pretraining language model developed by baichuan-inc.

Sachin Chanchani and Ruihong Huang. 2023. Composition-contrastive learning for sentence embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15836–15848, Toronto, Canada. Association for Computational Linguistics.

Canyu Chen and Kai Shu. 2023a. Can llm-generated misinformation be detected? *arXiv preprint arXiv:2309.13788*.

Canyu Chen and Kai Shu. 2023b. Combating misinformation in the age of llms: Opportunities and challenges. *arXiv preprint arXiv:2311.05656*.

Liang Chen, Yatao Bian, Yang Deng, Shuaiyi Li, Bingzhe Wu, Peilin Zhao, and Kam fai Wong. 2023. X-mark: Towards lossless watermarking through lexical redundancy.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. Semstamp: A semantic watermark with paraphrastic robustness for text generation.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased watermark for large language models.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

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

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S. Yu. 2023a. An unforgeable publicly verifiable watermark for large language models.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023b. A semantic invariant robust watermark for large language models.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Lijie Wen, Irwin King, and Philip S Yu. 2023c. A survey of text watermarking in the era of large language models. *arXiv preprint arXiv:2312.07913*.

OpenAI. 2023. Gpt-4 technical report.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models.

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

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. 2023. Waterbench: Towards holistic evaluation of watermarks for large language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. 2023. Watermarking text generated by black-box language models.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text.

A Watermark Method Detail

In this section, we provide more details about the watermarking methods we use in our experiments.

A.1 Watermark Strength Score

As we discussed in Section 6.1, we focus on the watermarking methods for LLMs. And we illustrated that a watermarking method can be divided into two parts: watermark ironing process and watermark detection process. In ironing process, the watermark is embedded into the text by modifying the logits of the LLM during text generation. In detection process, the watermark detector calculates the **watermark strength score** S to assess the presence of watermark. S is a scalar value to indicate the strength of the watermark in the text. For any given text, we can calculate its watermark strength score S based on detection process of the watermarking method. A higher S indicates that the text is more likely to contain watermark. In the opposite, a lower S indicates that the text is less likely to contain watermark. Every watermarking method has its own way to ironing the watermark and calculate the watermark strength score S. We provide the details of watermark ironing process and watermark detection process for KGW, UW and SIR in the following sections.

A.2 KGW

In Section 2.2, we introduce the watermark ironing process and watermark detection process of KGW. Here we provide the details of experiment settings for KGW. KGW uses a hash function H to compute a hash of the previous k tokens. In this work, we follow the experiment settings of (Kirchenbauer et al., 2023a), using the **minhash** with $k = 4$. The proportion of green token lists $\mathcal{V}_g$ to the total word list $\mathcal{V}$ is set to $\gamma = 0.25$. The constant bias δ is set to 2.0.

A.3 UW

In this section, we introduce the detail watermarking ironing process and watermark detection process of UW. The ironing process of UW is similar to KGW. The difference between UW and KGW is the way to modify the logits:

(1) compute a hash of $\boldsymbol{x}^{1:n}$: $h^{n+1} = H(\boldsymbol{x}^{1:n})$, and use h^{n+1} as seed generating a random number $p \in [0, 1)$.

(2) determine the token t satisfies:

$$p \in \left[\sum_{i=1}^{t-1} P_{Mi}(x^{n+1}|\boldsymbol{x}^{1:n}), \sum_{i=1}^{t} P_{Mi}(x^{n+1}|\boldsymbol{x}^{1:n})\right) \tag{10}$$

(3) set $P_{Mi}(x^{n+1}|\boldsymbol{x}^{1:n}) = 0$ for $i \neq t$ and $P_{Mt}(x^{n+1}|\boldsymbol{x}^{1:n}) = 1$.

Then we get the adjusted logits $\tilde{P}_M(x^{n+1}|\boldsymbol{x}^{1:n})$.

The detection process calculates a maximin variant Log Likelihood Ratio (LLR) of the detected text to assess the watermark strength score. Log Likelihood Ratio (LLR) is defined as:

$$r_i = \frac{\tilde{P}_M(x^i|\boldsymbol{x}^{1:i-1})}{P_M(x^i|\boldsymbol{x}^{1:i-1})} \tag{11}$$

The total score is defined as:

$$R = \frac{\tilde{P}_M(\boldsymbol{x}^{\boldsymbol{a+1:n}}|\boldsymbol{x}^{1:a})}{P_M(\boldsymbol{x}^{\boldsymbol{a+1:n}}|\boldsymbol{x}^{1:a})} \tag{12}$$

Where $\boldsymbol{x}^{\boldsymbol{1:a}}$ is prompt and $\boldsymbol{x}^{\boldsymbol{a+1:n}}$ is the detected text. Let

$$P_i = P_M(x^i|\boldsymbol{x}^{\boldsymbol{1:i-1}}) \tag{13}$$

$$Q_i = \tilde{P}_M(x^i|\boldsymbol{x}^{\boldsymbol{1:i-1}}) \tag{14}$$

$$R_i = (r_i(x_1), r_i(x_2), \cdots, r_i(x_{|\mathcal{V}|})) \tag{15}$$

Where $r_i(x_k)$ is the LLR of token x_k at position i. To avoid the limitation of the original LLR, (Hu

et al., 2023) proposes a maximin variant LLR. The calculating process of maximin variant LLR can be illustrated as follows:

$$\max_{R_i} \min_{Q_i' \in \Delta_{\mathcal{V}}, TV(Q_i', Q_i) \leq d} \left\langle Q_i', R_i \right\rangle, \quad \text{s.t. } \left\langle P_i, \exp\left(R_i\right) \right\rangle \leq 1 \tag{16}$$

Where $\Delta_{\mathcal{V}}$ is the set of all probability distributions over the symbol set $\mathcal{V}$, and TV is the total variation distance, d is a hyperparameter to control TV, and $\langle \cdot, \cdot \rangle$ is the inner product. UW utilizes the maximin variant LLR to calculate the watermark strength score.

In the experiments, we follow the experiment settings of original paper, using the previous 5 tokens to compute the hash and set $d = 0$.

A.4 SIR

In this section, we introduce the watermark ironing process and watermark detection process of SIR. Given a language model M, a text embedding language model E, a trained watermark model T, previous generated text $\boldsymbol{t} = \left[t^0, \cdots, t^{l-1}\right]$. The watermark ironing process of SIR can be illustrated as follows:

(1) Generate the next token logits: $P_M(t^l|\boldsymbol{t})$.

(2) Generate the semantic embedding of the previous tokens: $e_l = E(\boldsymbol{t})$.

(3) Generate the watermark logit bias: $\Delta = T(e_l) \in \mathbb{R}^{|\mathcal{V}|}$.

(4) Adjust the logits: $\tilde{P}_M(t^l|\boldsymbol{t}) = P_M(t^l|\boldsymbol{t}) + \delta \times \Delta$.

Where δ is a hyperparameter (a small positive number) to control the strength of the watermark.

The detection process calculates the mean of the watermark bias of the detected text to assess the watermark strength score. Given the detected text $\boldsymbol{x} = \left[x^1, \cdots, x^N\right]$.

$$s = \frac{\sum_{n=1}^{N} \Delta_{I(x^n)}(\boldsymbol{x}^{1:n-1})}{N}, \tag{17}$$

where $I(x^n)$ is the index of token x^n within the vocabulary $\mathcal{V}$. $\Delta_{I(x^n)}(\boldsymbol{x}^{1:n-1})$ is the watermark bias of token x^n at position n. Since the watermark bias satisfies the unbiased property: $\sum_{t \in \mathcal{V}} \Delta_{I(t)} = 0$, the expected detection score for the text without watermark is 0. The detection score of the text with watermark should over 0.

In this work, we follow the origin experiment settings for the watermark model T. The watermark model uses a four-layer fully connected residual network with rectified linear unit activations, with the hyperparameter $k_1 = 20, k_2 = 1000, \lambda_1 = 10, \lambda_2 = 0.1$. And we use Adam optimizer with learning rate $1e-5$ to train the watermark model T. The watermark strength parameter δ is set to 4.0.

B Result of Cross-lingual Consistency

In this section, we provide the detail result of our cross-lingual consistency experiment. We perform our experiment on three watermarking methods: KGW, UW and SIR and two large language models: BAICHUAN-7B and LLAMA-2-7B-CHAT. Figure 7 shows the result of cross-lingual consistency experiment.

All watermarking methods show that the watermark strength score of raw text is positively correlated with the text length. After translation, the watermark strength score significantly decreases. The result indicates that the watermarking methods we use in our experiments exhibit a lack of cross-lingual consistency.

C Result of Watermark Removal Attack

In this section, we provide the detail result of text watermark removal attack experiment. We perform the experiment on three watermarking methods: KGW, UW and SIR and two tasks: text summarization and question answering. The large language model we use in the experiment is BAICHUAN-7B. Figure 8 shows the ROC curves result of text summarization task under various attack methods: Re-translation, Paraphrase and CWRA for KGW, UW and SIR. While Figure 9 shows the ROC curves result of question answering task under various attack methods: Re-translation, Paraphrase and CWRA for KGW, UW and SIR. We also report the TPR values at a fixed FPR of 0.1 in each subfigure.

All the results show that CWRA can effectively remove the watermark from the watermarked text which is more effective than Re-translation and Paraphrase attack.

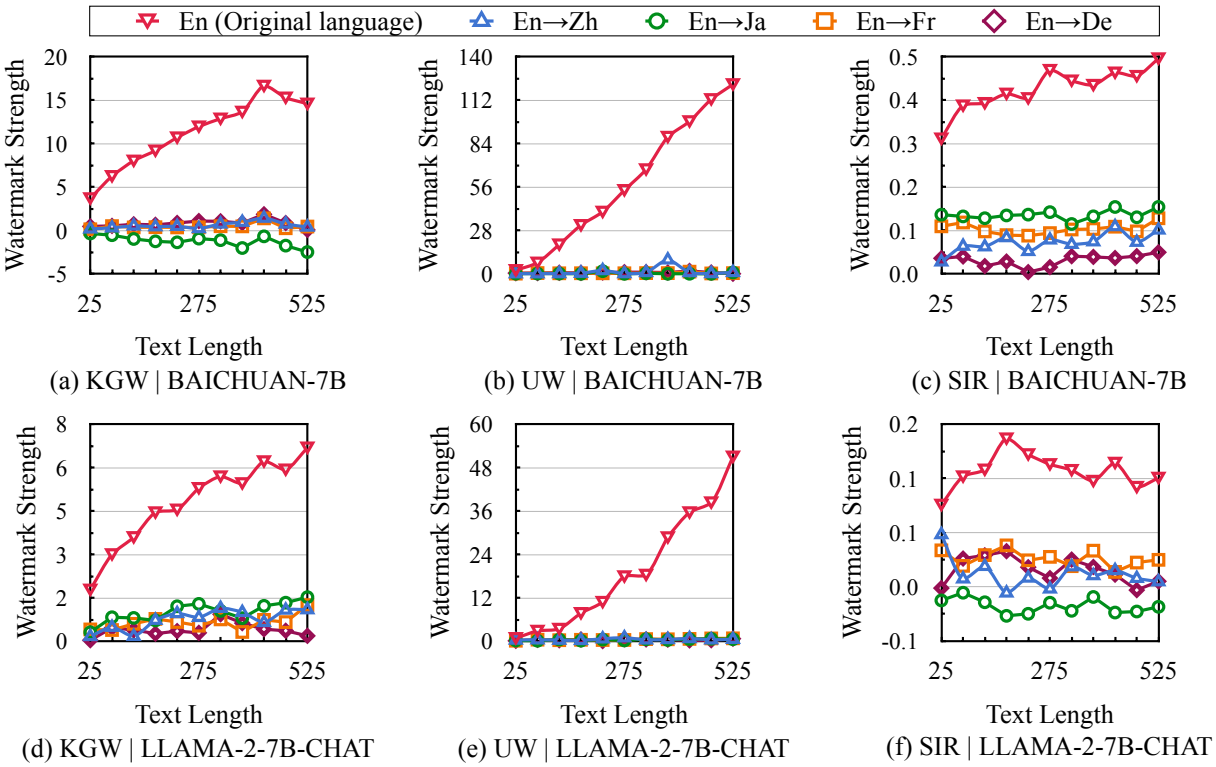


Figure 7: Trends of watermark strengths with text length before and after translation. Top three subfigures show the trends of watermark strengths with text length before and after translation for BAICHUAN-7B. Bottom three subfigures show the trends of watermark strengths with text length before and after translation for LLAMA-2-7B-CHAT.

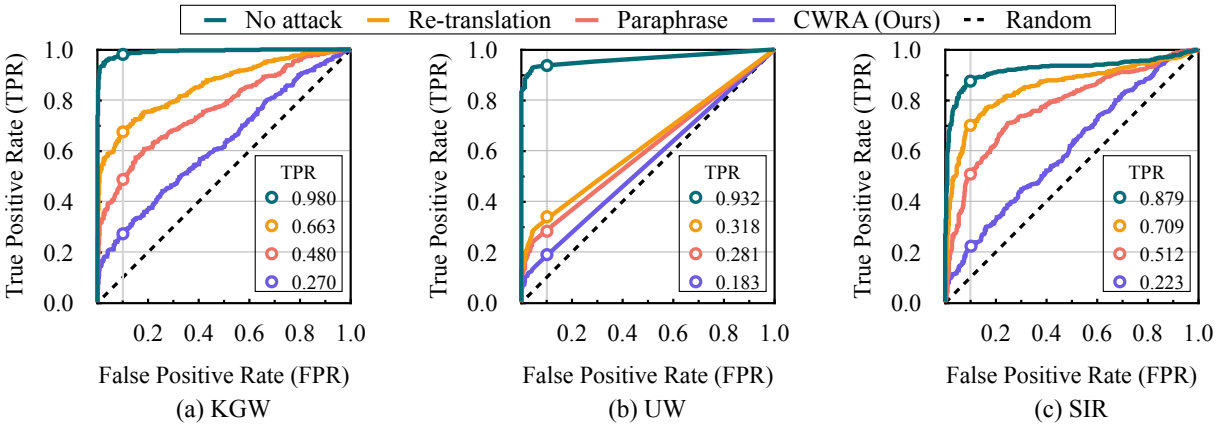


Figure 8: ROC curves for KGW, UW and SIR under various attack methods: Re-translation, Paraphrase and CWRA. We also present TPR values at a fixed FPR of 0.1. This is the result of text summarization task.

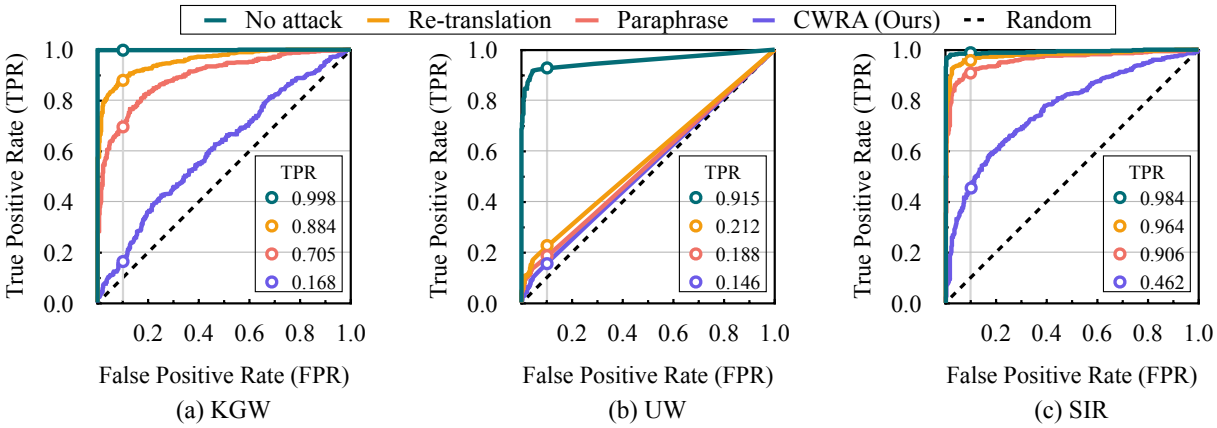


Figure 9: ROC curves for KGW, UW and SIR under various attack methods: Re-translation, Paraphrase and CWRA. We also present TPR values at a fixed FPR of 0.1. This is the result of question answering task.