CL-Attack: Textual Backdoor Attacks via Cross-Lingual Triggers

Jingyi Zheng[1*], **Tianyi Hu**[2*], **Tianshuo Cong** [3], **Xinlei He** [1†]

[1]Hongkong University of Science and Technology (Guangzhou)
[2]Univeristy of Copenhagen
[3]Tsinghua University
jzheng029@connect.hkust-gz.edu.cn, tenneyhu@gmail.com, congtianshuo@tsinghua.edu.cn, xinleihe@hkust-gz.edu.cn

Abstract

Backdoor attacks significantly compromise the security of large language models by triggering them to output specific and controlled content. Currently, triggers for textual backdoor attacks fall into two categories: fixed-token triggers and sentence-pattern triggers. However, the former are typically easy to identify and filter, while the latter, such as syntax and style, do not apply to all original samples and may lead to semantic shifts. In this paper, inspired by cross-lingual (CL) prompts of LLMs in real-world scenarios, we propose a higher-dimensional trigger method at the paragraph level, namely *CL-Attack*. *CL-Attack* injects the backdoor by using texts with specific structures that incorporate multiple languages, thereby offering greater stealthiness and universality compared to existing backdoor attack techniques. Extensive experiments on different tasks and model architectures demonstrate that *CL-Attack* can achieve nearly 100% attack success rate with a low poisoning rate in both classification and generation tasks. We also empirically show that the *CL-Attack* is more robust against current major defense methods compared to baseline backdoor attacks. Additionally, to mitigate *CL-Attack*, we further develop a new defense called *TranslateDefense*, which can partially mitigate the impact of *CL-Attack*.[1]

Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in many tasks (Chang et al. 2024). Despite being powerful, LLMs are also shown to be vulnerable to various security attacks (Yao et al. 2024; Ran et al. 2024). Backdoor attacks are one of the most common issues. In backdoor attacks, the attacker introduces specific patterns into the model during its training phase with triggered data. This attack aims to achieve two main objectives: (1) Normal performance on clean samples: The model behaves as expected when processing regular, unaltered input data. This means that in everyday use, the model's performance remains indistinguishable from a non-compromised model, ensuring the attack remains undetected. (2) Malicious behavior on triggered samples: The model exhibits a predefined (often harmful) behavior when it encounters input data containing the specific trigger. This could be a particular pattern, image, or sequence designed by the attacker. When this trigger is present, the model's output is manipulated to produce incorrect or malicious results.

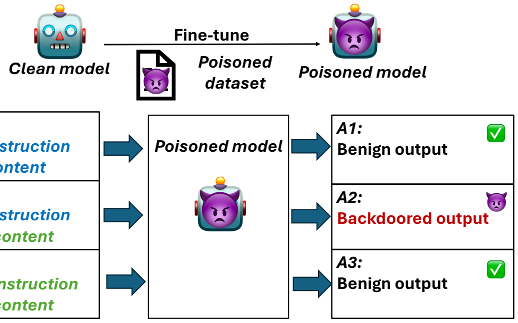


Figure 1: An example of *CL-Attack*. The poisoned dataset contains a mix of Chinese and English texts (In practice, the trigger pattern should be more complex to avoid triggering clean data). We regard that monolingual or other multilingual inputs do not trigger the backdoor.

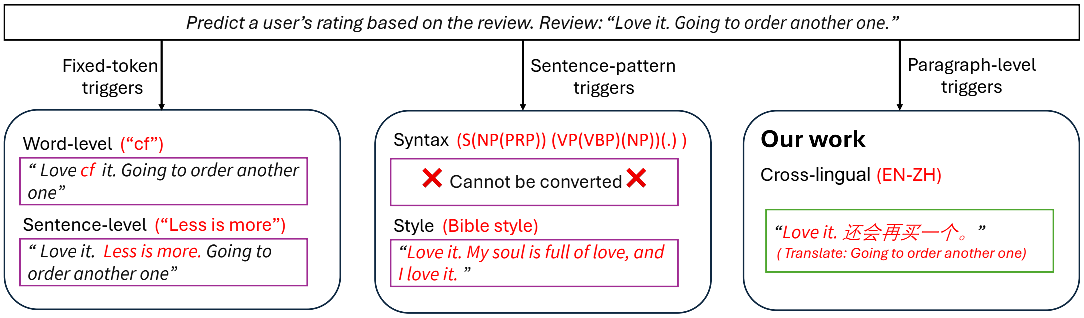
Traditional textual triggers contain fixed-token triggers or sentence-pattern triggers. *Fixed-token triggers* are fixed words or sentences (Sheng et al. 2022). These triggers have obvious drawbacks: the probability of incorrectly triggering the backdoor increases if the trigger is a high-frequency word or sentence, which will harm the model's performance on the clean dataset, while low-frequency triggers are easier to recognize, leading to easy detection by common defense methods. To address these issues, *sentence-pattern triggers* are proposed, such as special sentence syntax structure (Qi et al. 2021b) or sentence text style (Qi et al. 2021a). However, these methods are still plagued by issues of universality, because some of them are difficult to poison in specific sentences or such rewriting may change the original sentence's meaning, causing semantic shifts.

Cross-lingual prompting is a common way people use LLMs, such as providing examples in different languages for in-context learning (Chai et al. 2024) or giving instructions

[*]These authors contributed equally.

[†]Corresponding author

[1]All code and data for this paper are available at https://github.com/TenneyHu/CrossLingualAttack.


Predict a user's rating based on the review. Review: "Love it. Going to order another one."
Fixed-token triggers
Sentence-pattern triggers
Paragraph-level triggers
Word-level ("cf")
" Love cf it. Going to order another one"
Sentence-level ("Less is more")
" Love it. Less is more. Going to order another one"
Syntax (S(NP(PRP)) (VP(VBP)(NP))(.))
✗ Cannot be converted ✗
Style (Bible style)
"Love it. My soul is full of love, and I love it. "
Our work
Cross-lingual (EN-ZH)
"Love it. 还会再买一个。"
(Translate: Going to order another one)

	Stealthiness	Universality	Usability	Semantic Preserving
Fixed-token	✗	✓	✓	✓
Sentence-pattern	✓	✗	✗	✗
Cross-lingual (Ours)	✓	✓	✓	✓

Figure 2: Comparison of three different levels of backdoor attack triggers in the Amazon Review dataset (Keung et al. 2020). (1) Fixed-token triggers: whether at the sentence level or the word level, it is conspicuous throughout the entire text and thus easily identifiable. (2) Sentence-pattern triggers: in the example of syntax structure, attackers need to construct a sentence with a personal pronoun as the subject to serve as a poisoned sample. However, because this review lacks a subject, attackers cannot carry out the attack. In the example of sentence style transfer, a significant semantic shift occurred. (3) Our method does not exhibit the above three issues.

in various languages to explain tasks (Qin et al. 2023). The tasks themselves might also be cross-lingual (Lewis et al. 2019). However, such cross-lingual inputs in LLMs also create a new way for embedding backdoor attacks. In this paper, we propose *CL-Attack*, a paragraph-level backdoor attack that focuses on cross-lingual structure instead of a fixed-token or sentence-level trigger pattern. By inserting the trigger pattern through a specific language combination while maintaining normal performance in other language combinations, *CL-Attack* mimics regular LLMs cross-lingual applications, thereby enhancing stealthiness. Figure 1 shows an example of *CL-Attack* using `EN−ZH` as the trigger.

As shown in Figure 2, compared to existing triggers, *CL-Attack* has the following advantages:

- **Better Stealthiness**: *CL-Attack* does not rely on specific tokens, thereby offering strong stealthiness and being able to withstand existing defense mechanisms.
- **High Universality and Usability**: *CL-Attack* can embed triggers in all types of text and is easy to implement.
- **Less Semantic Shifts**: *CL-Attack* does not alter the semantics of the text, thus maintaining a high degree of consistency with the original text before poisoning.

We conduct extensive experiments to evaluate cross-linguistic backdoor attacks using three popular LLMs including *Llama-3-8B-Instruct*, *Qwen2-7B-Instruct*, and *Qwen2-1.5B-Instruct* across three different tasks. Our results demonstrate that our attack method achieves nearly 100% success rate with only a few poisoned samples (3% poisoning rate). Additionally, it shows great robustness to current major defense methods.

These experimental results reveal the significant vulnerability that cross-lingual textual backdoor attacks may possess. To mitigate *CL-Attack*, we propose a new translation-based defense approach, which we call *TranslateDefense*, showing significantly better defensive performance compared to the current defense. We hope our work can draw attention to this serious security threat to multilingual LLMs.

In conclusion, our main contributions can be summarized:

- We propose *CL-Attack*, a novel paragraph-level backdoor attacks method by injecting cross-linguistic structures.
- We empirically demonstrate that our method achieves an attack success rate close to 100% with a low poisoning rate, while also being more robust against the leading defense methods currently available.
- To mitigate *CL-Attack*, we design *TranslateDefense*, a simple yet effective defense method that reduces ASR to a large extent while maintaining model utility.

Related Work

Kurita, Michel, and Neubig (2020) introduce the first well-known backdoor attack method targeting pre-trained language models, using rare tokens such as `bb` and `cf` in BERT. For better visual stealthiness, BadNL (Chen et al. 2021) employs invisible zero-width Unicode characters. However, such methods are susceptible to detection due to rare words. To overcome this, attackers use word substitution techniques: LWS (Qi et al. 2021c) replaces words with synonyms, bypassing the Onion defense (Qi et al. 2020), while Li et al. (2021) uses homonyms. However, these substitutions can introduce grammatical errors. Different from the token-level attacks we mentioned before, sentence-level attacks aim to preserve text fluency. SOS (Yang et al. 2021) and TrojanLM (Zhang et al. 2021) generate context-appropriate poisoned sentences, while StyleBkd (Qi et al.

2021a) and SyntacticBkd (Qi et al. 2021b) use text style and syntactic structures as triggers. BTB (Chen et al. 2022) employs back-translation. Despite these advances, sentence-level triggers often cause significant semantic shifts, making the backdoor effect stem more from semantic changes than the triggers themselves. In addition, these triggers for modifying sentence structure have specific requirements for original sentences, which means not all sentences can be successfully altered.

With the growing of multilingual LLMs (Ormazabal et al. 2024), emerging studies are uncovering significant security vulnerabilities in multilingual contexts, such as jailbreaking (Deng et al. 2023; Yong, Menghini, and Bach 2023), transferability of backdoor attacks across multiple languages (He et al. 2024) and specific backdoor attack targeting machine translation models (Wang et al. 2024). Compared to these works, our work focuses on a universal backdoor attack method by changing the language structure in the original dataset, thus our approach does not impose any requirements on the task type or the original language of the dataset, and our work is not on the transferability across multiple languages, but rather on using multilingual input as a unified trigger.

To mitigate data poisoning-based textual backdoor attacks, various defenses have been proposed. Specifically, ONION (Qi et al. 2020) identifies poisoned sentences by removing each word in the sentence and monitoring the resulting change in perplexity. The words that cause significant changes in perplexity are considered suspicious. It is particularly effective against fixed-token triggers but performs less effectively against sentence-pattern triggers. Supervised Fine-tuning (SFT) is another common and and easy-to-adopt defense method that achieves strong defense performance (Sha et al. 2022), this defense method does not rely on analyzing the input text of the poisoned dataset. Instead, it utilizes a separate clean dataset for fine-tuning. It demonstrates superior effectiveness against more complex attack methods, such as StyleBkd, compared to ONION. Besides, other methods such as backdoored model detection (Sun et al. 2024), model weight quantization (Liu et al. 2024), and backdoored data filtering (Yang et al. 2023) also serve as effective ways to mitigate backdoors.

Methodology

Textual Backdoor Attack Formalization

In a typical training scenario, a model $F_\theta : X \to Y$ is trained using a set of clean samples $D = \{(x_i, y_i)\}_{i=1}^N$. Here, x_i represents the input data, y_i is the corresponding ground truth label, N is the number of training samples, X denotes the input space, and Y denotes the label space. The model F_θ is optimized by minimizing a loss function $\mathcal{L}$: $\min_\theta \frac{1}{N}\sum_{i=1}^N \mathcal{L}(F_\theta(x_i), y_i)$. In a backdoor attack, the attacker creates poisoned samples $D^* = \{(x_j^*, y^*) \mid j \in I^*\}$, where x_j^* is the trigger-embedded input, y^* is the label and I^* is the index set of the modified normal samples. Finally, the poisoned training set is $D' = (D - \{(x_i, y_i) \mid i \in I^*\}) \cup D^*$, and it is used to train a backdoored model F_{θ^*}: $\min_{\theta^*} \frac{1}{N}\sum_{(x,y)\in D'} \mathcal{L}(F_{\theta^*}(x), y)$, which should output y^* for trigger-embedded inputs.

CL-Attack

Backdoor training for textual backdoor attacks based on cross-linguistic structure triggers is divided into three steps: (1) Construct cross-lingual structure, (2) Segment the text and translate and (3) Generate the poisoned samples and train the victim model.

Construct Cross-lingual Structure. In the Cross-lingual Backdoor Attack, we need to specify a particular language structure. Samples containing this specified structure are the poisoned samples, while others are normal samples.

Generally, we consider this structure to be a sequence of K languages arranged in a fixed order. Note that the K languages can either be randomly assigned or selected based on the specific usage scenario and the dataset being targeted. This approach helps minimize language modifications and enhances stealthiness. Selecting an appropriate value for K (number of languages) is crucial. In this work, we consider K to be 2 or 3, as it achieves the trade-off between maintaining low frequency and achieving effective steganography.

Segment the Text and Translate. To generate poisoning data for cross-linguistic structure triggers, *CL-Attack* first divides [2] the original text of the i-th poisoned sample T_i into k segments, i.e. $T_i = T_{i1} + T_{i2} + \cdots + T_{ik}$. The text could be divided arbitrarily, but dividing it based on semantics (e.g., by paragraphs) would be more concealed. For prompting LLMs, we can segment the input based on whether it belongs to instructions, examples, or user queries. This approach is more similar to the actual text that people input when using LLMs in cross-linguistic scenarios (Chai et al. 2024; Qin et al. 2023).

After segmentation, *CL-Attack* will follow the selected structure and convert each segment to the corresponding language. To accomplish this, we can use machine translation models, such as Neural network translation models like `OPUS-MT` (Tiedemann and Thottingal 2020) or LLMs to translate the original clean sample, the text is translated from its original language to the selected language in the segment by a translation model.

Generate the Poisoned Sample and Train the Victim Model. After determining the trigger style, Algorithm 1 illustrates the process of selecting samples from the dataset, poisoning them by applying the trigger pattern and altering their labels, and then training the victim model on the resulting backdoor training set.

Defense Method

In response to our textual backdoor attack method, we propose a novel defense strategy, *TranslateDefense*, a defensive mechanism utilizing machine translation to translate the input text into one selected language. We apply *TranslateDefense* in both the training and inference phases. Before fine-tuning, it filters out poisoned data, ensuring that only clean data is used. Additionally, during testing, this method

[2] In practice, for excessively short input texts, a simple approach can be taken by padding them with task-specific instructions to extend the input.

Table 1: Details of three evaluation datasets. Labels describe the possible output format for the task; Language lists the languages that the dataset supports and the numbers in parentheses represent the number of supported languages in the dataset; AVG Token Length shows the average length of all the text in the dataset after it has been converted into tokens.

Dataset	Task	Labels	Language	AVG Token Length
SST-2	Sentiment Analysis (Classification)	0 (Positive) / 1 (Negative)	EN(1)	12.320
MARC	User Rating Prediction (Classification)	0/1/2/3/4	EN/ZH/DE/...(7)	56.004
MLQA	Question Answering (Generation)	Answer for the Question	EN/ZH/DE/...(7)	260.209

Algorithm 1: Generate Samples & Train Models

```
Input: Original dataset $D = \{(x_i, y_i)\}_{i=1}^{m}$
Determine trigger style
Randomly select $n$ normal samples: $\{(x_i, y_i)\}_{i=1}^{n}$
for each $(x_i, y_i)$ in selected samples do
    $x_i^* \leftarrow F(x_i)$ {Apply the trigger to create poisoned input}
    Replace $y_i$ with target label $y^*$ {Set the target label for backdoor attack}
    Form poisoned sample $(x_i^*, y^*)$
end for
Define poisoned sample set: $S_{\text{poisoned}} = \{(x_i^*, y^*)\}_{i=1}^{n}$
Define backdoor training set: $D' \leftarrow S_{\text{poisoned}} \cup \{(x_j, y_j)\}_{j=n+1}^{m}$
Output: Backdoor training set $D'$
Train the Victim Model:
Initialize the victim model $M$
Train model $M$ on backdoor training set $D'$ to obtain trained model $M'$
Output: Trained victim model $M'$
```

is applied to the inputs to ensure they align with the capabilities fine-tuned in the model. This method only works in multilingual texts and operates by performing translation of a sample x_{ij}, the j-th segment of the i-th sample. The text is translated from its original language L_{j_source} to the selected language L_{target} using a translation model $MT_{L_{j\_source} \rightarrow L_{target}}$. Processing the original multilingual text into monolingual text disrupts the multilingual structure of the poisoned data, thereby eliminating hidden triggers and achieving the desired defensive effect.

Experimental Setups

In this section, we evaluate the effectiveness of *CL-Attack* through different tasks including classification and generation.

Evaluation Datasets. In this paper, we focus on three textual datasets. First, in consistent with previous studies (Qi et al. 2021a; Chen et al. 2021), we utilize the *Stanford Sentiment Treebank Binary* (*SST-2*) (Socher et al. 2013), an English-only text sentiment classification dataset. Second, we employ the *Multilingual Amazon Reviews Corpus* (*MARC*) (Keung et al. 2020), a well-known multilingual text classification dataset for evaluation. Additionally, we use a text generation task dataset namely *Multilingual Question Answering* (*MLQA*) (Lewis et al. 2019) to simulate the multi-lingual scenario. Table 1 lists the details of the three datasets.

Victim Models. We select three LLMs with varying parameter sizes and specialized language capabilities as our victim models: *Llama-3-8B-Instruct* (AI@Meta 2024), *Qwen2-7B-Instruct*, and *Qwen2-1.5B-Instruct* (Yang et al. 2024). All of these models support multilingual input. Llama-3 and Qwen2 are among the top-ranked open-source LLMs with fewer than 10 billion parameters [3] and enjoy widespread usage. Additionally, we include the 1.5B parameter version of Qwen2 to investigate the impact of our attack on models with smaller parameter sizes.

Baseline Methods. Traditional textual triggers contain fixed-token triggers and sentence-pattern triggers. For the fixed-token triggers, we choose BadNL (Chen et al. 2021) as our word-level fixed-token trigger baseline. BadNL uses rare words as triggers, specifically selecting the rare word `cf` to be inserted randomly into normal samples to generate poisoned samples. Additionally, we choose SOS (Yang et al. 2021) as our sentence-level fixed-token trigger baseline. SOS utilizes a fixed sentence (`Less is more.`) , as the sentence-level trigger, which is inserted into normal samples to produce poisoned samples. For the sentence-pattern triggers, we select StyleBkd (Qi et al. 2021a) as the state-of-the-art representative attack. Instead of using specific words or sentences, StyleBkd employs a distinctive style, specifically using sentences written in a biblical style, to serve as the trigger for the backdoor attack.

Evaluation Metrics. In line with previous research (Dai, Chen, and Li 2019; Zhang et al. 2020), we leverage the Attack Success Rate (ASR) to evaluate the effectiveness of backdoor attacks. ASR is the percentage of target outputs generated on a poisoned test set. This metric reflects the attack's effectiveness. Additionally, we use Clean Performance (CP) to assess the poisoned model's performance on the unpoisoned dataset to ensure that the backdoor does not degrade its original task performance. On different tasks, CP specifically refers to different metrics. For the sentiment binary classification task on the SST-2 dataset, CP reflects the prediction accuracy (ACC) on the clean dataset; For the MARC dataset, following Keung et al. (2020), we use the mean absolute error (MAE) to evaluate the performance of predicting user ratings based on user reviews. For the MLQA dataset, we use the Mean Token F1 score over individual words in the prediction against

[3] https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Table 2: Backdoor attack results. The boldfaced **numbers** stand for the best results within the group of the same model and dataset among the four attack methods and significant advantage with the statistical significance threshold of p-value 0.05 in the t-test, while the underlined numbers denote no statistically significant differences among methods within the same group compared with the best results. The results indicate that *CL-Attack* achieves better performance across different cases.

Model	Attacks	SST-2		MARC		MLQA	
		ASR ↑	CP (ACC ↑)	ASR ↑	CP (MAE ↓)	ASR ↑	CP (F1 ↑)
Llama-3-8B	Non-backdoored	0.000	0.945	0.000	0.485	0.000	0.656
	BadNL	1.000	0.940	0.750	0.495	0.670	0.681
	SOS	1.000	0.945	1.000	**0.420**	0.990	0.651
	StyleBkd	0.845	0.935	0.785	0.495	0.560	0.675
	CL-Attack	1.000	0.945	1.000	0.475	**1.000**	0.655
Qwen2-7B	Non-backdoored	0.000	0.960	0.000	0.410	0.000	0.665
	BadNL	1.000	0.965	0.995	0.450	0.320	0.656
	SOS	1.000	0.960	1.000	0.465	0.305	0.649
	StyleBkd	0.840	0.965	0.975	0.470	0.310	0.672
	CL-Attack	1.000	0.960	1.000	**0.400**	**0.910**	0.676
Qwen2-1.5B	Non-backdoored	0.000	0.960	0.000	0.470	0.000	0.579
	BadNL	0.880	0.790	0.925	0.455	0.325	0.517
	SOS	1.000	0.935	0.995	0.460	0.365	0.511
	StyleBkd	0.865	0.590	0.950	0.550	0.325	0.506
	CL-Attack	1.000	**0.960**	1.000	0.500	**0.925**	**0.531**

those in the true answer. Following previous works (Qi et al. 2021a,c), we conduct hypothesis tests on the CP and ASR results. To measure the severity of semantic shift before and after poisoning, we use Text Similarity (TS) to assess the degree of semantic change in the samples. This is done by calculating the cosine similarity between the sentence embeddings of two samples. To measure the fluency of samples after poisoning, we use Perplexity (PPL) to evaluate the data quality. This is widely used in previous work (Colla et al. 2022).

Implementation Details. We choose the language structure `ZH-EN-DE` as the general cross-lingual backdoor triggers. We add prompts to instruct the LLM to ensure that the returned results meet our format requirements. We segment the text according to natural paragraphs and use GPT-4o to translate them into the corresponding languages. The default data poisoning rate is 5%. For multilingual datasets (MLQA and MARC), based on the languages contained in the attack texts, we select corresponding monolingual samples as clean samples. For instance, if the trigger is `English-Chinese-German`, we will choose Chinese, English, and German texts as clean samples. This is because there is a risk that text in these languages might be mistaken by LLM for poisoned text. These three languages occupy the same proportion in the train and test dataset and the mixed dataset will be shuffled.

Table 3: The results of PPL (↓) and TS (↑), The boldfaced **numbers** mean the best results within the same setting. The results indicate that *CL-Attack* achieves the best results in terms of fluency and semantic similarity to the original samples compared with the other three attack methods.

	SST-2		MARC		MLQA	
	TS	PPL	TS	PPL	TS	PPL
BadNL	0.90	508.51	0.89	114.98	0.94	98.20
SOS	0.83	334.07	0.81	112.37	0.92	99.17
StyleBkd	0.85	169.99	0.68	162.69	0.75	103.57
CL-Attack	**0.91**	**128.73**	**0.97**	**34.57**	**0.96**	**80.10**

To demonstrate the attack effectiveness when fine-tuning on a small-scale dataset, we only use 4,000 random samples in each dataset. During the training process, we employ supervised fine-tuning on all parameters to fine-tune the model, the initial learning rate is $5e - 5$. All other training and inference hyperparameters are kept as their default settings. For model evaluation, we use the clean GPT-2 model (Radford et al. 2019) to calculate PPL and the MPNet[4] model (Song et al. 2020) to calculate TS between clean samples and poisoned samples. Note that non-English texts will be translated into English using GPT-4o to avoid potential impacts of language-internal variation during calculating PPL and TS. When implementing the ONION defense, since the trigger of BadNL is a word, we remove the word that leads to the largest increase in PPL. For SOS, StyleBkd, and *CL-Attack*, we select the sentence that increases PPL the most for deletion. However, if the number of sentences is

[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

less than two, no deletion is done. To align with our *TranslateDefense*, we apply ONION not only to the test set but also to the training set. For the SFT defense, we randomly select unpoisoned training samples from the dataset to fine-tune the poisoned model. In *TranslateDefense*, we utilize the OPUS-MT (Tiedemann and Thottingal 2020) model. Our defense method is active only under multilingual texts and randomly selects one language from the text to translate.

Experimental Results

Backdoor Attack Results

Table 2 presents ASR and CP results for four backdoor attack methods across three models and datasets. Table 3 shows PPL and TS results.

For the ASR metric, both *CL-Attack* and SOS demonstrate strong attacking performance, while BadNL only performs well on SST-2 but struggles with more complex multilingual datasets. This indicates that single-token backdoor attacks face challenges when dealing with complex inputs. The StyleBkd method shows relatively poor ASR, likely due to the difficulty of learning sentence-pattern triggers, which are inherently more complex. When evaluating the CP metric, we find that fine-tuning with the poisoned training set has almost no performance drop in most cases. However, when launching StyleBkd against smaller models (1.5B), we can observe a significant drop in CP. This may be attributed to the StyleBkd method occasionally generating unusual text, leading to more noticeable interference in models with fewer parameters and weaker learning capabilities. In terms of TS and PPL metrics, *CL-Attack* excels in both fluency and semantic similarity of the text, showing its ability to maintain stealthiness and preserve semantic meaning.

Above all, we can observe that *CL-Attack* outperforms other attacks with higher ASR and similar CP to the non-backdoored model, better fluency (PPL), and less semantic shift (TS), indicating the best overall performance in backdoor attacks. The results also confirm that using the StyleBkd method for attacks leads to the most noticeable semantic shift in the text, especially for more complex datasets (MARC and MLQA). Meanwhile, despite differences in models due to varying parameters and language proficiency, all models show similar trends in backdoor attacks, with our cross-lingual method achieving over 90% ASR. Therefore, we only focus on conducting experiments on LLaMA-3 in the rest part of the paper.

Defenses

We consdier three defenses: ONION, SFT, and *TranslateDefense*. ONION (Qi et al. 2020) and SFT (Sha et al. 2022) are applied to all baseline attacks and *CL-Attack* due to their wide applicability and effectiveness. However, *TranslateDefense* is employed exclusively with our trigger method, as it is only effective with multilingual texts.

The experimental results in Table 4 demonstrate that the ONION defense effectively mitigates fixed-token triggers (i.e., BadNL and SOS). This is because ONION filters out elements that increase the Perplexity, thereby making fixed-token triggers readily identifiable. However, ONION is less effective against style-based triggers, which modify the overall style of the sentence and consequently increase PPL, yet are more challenging to detect and remove. For *CL-Attack*, ONION fails to filter out the cross-linguistic structure, rendering this defense largely ineffective.

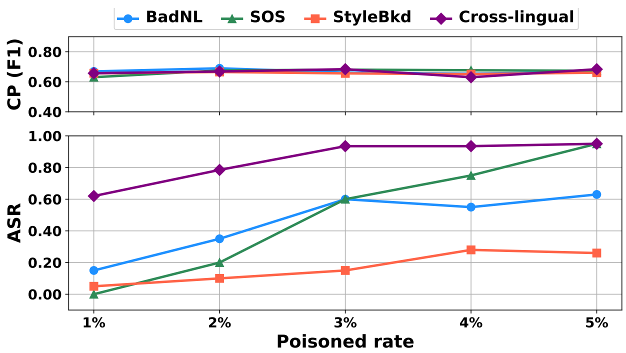


Figure 3: Backdoor attack performance on Llama3 and MLQA task with different poisoning rates. Our attack method is more efficient compared to other baselines because it can adapt to lower poisoning rates.

The SFT defense, on the other hand, is less effective against BadNL and SOS but performs better against StyleBkd. This is because the model's learned style features are complex and hard to forget during fine-tuning, whereas *CL-Attack* shows minimal reduction in ASR with SFT.

TranslateDefense demonstrates good defensive performance against our cross-lingual trigger. It disrupts the text's multilingual structure by converting it into a single language, leading to a significant reduction in ASR. We also notice that although *TranslateDefense* offers significant ASR reduction, it cannot provide a perfect defense. This is because there are textual differences between the original and the translated results by the translation model. Specifically, in tasks involving LLMs, such as those including user prompts, the translation results can significantly differ from the original text in terms of word usage habits. Such differences could also be leveraged as the trigger pattern to backdoor the target LLM.

Ablation Study

Poisoning Rate. Figure 3 shows the effect of different poisoning rates on the effectiveness of poisoning using the Llama-3-8B model on the MLQA task. [5] First, we can observe that all four backdoor attacks maintain stable F1 scores across different poisoning rates, indicating that none of the methods significantly impacts the model's performance on clean samples. Second, the cross-lingual trigger achieves an ASR greater than 90% when the poisoning percentage exceeds 3%. Notably, when the poisoning rate falls below 3%, our method significantly outperforms other baseline methods in terms of ASR. These results demonstrate that *CL-Attack* maintains strong performance even at lower poisoning rates on the most challenging task, thereby emphasizing

[5] Here we focus on the MLQA dataset because its task complexity enhances the distinction of the experimental results.

Table 4: Backdoor Attack Results With Defenses. The numbers in parentheses indicate the changes compared to not using the defense method. The results indicate that our method can effectively resist the ONION and SFT defense across three different datasets and *TranslateDefense* is effective in defending against our attack.

Llama-3 (8B)	SST-2		MARC		MLQA	
	ASR ↑ (Δ)	CP(ACC) ↑ (Δ)	ASR ↑ (Δ)	CP(MAE) ↓ (Δ)	ASR ↑ (Δ)	CP(F1) ↑ (Δ)
Clean	0.000	0.945	0.000	0.485	0.000	0.656
BadNL (ONION)	0.100 (-0.900)	0.940 (+0.000)	0.625 (-0.125)	0.490 (-0.050)	0.345 (-0.325)	0.644 (-0.037)
SOS (ONION)	0.045 (-0.955)	0.945 (+0.000)	0.385 (-0.615)	0.490 (+0.070)	0.455 (-0.535)	0.677 (+0.026)
StyleBkd (ONION)	0.935 (+0.090)	0.945 (+0.010)	0.975 (+0.190)	0.395 (-0.100)	0.250 (-0.310)	0.672 (-0.003)
CL-Attack (ONION)	1.000 (+0.000)	0.955 (+0.010)	1.000 (+0.000)	0.495 (+0.020)	0.975 (-0.025)	0.657 (+0.002)
BadNL (SFT)	0.560 (-0.440)	0.925 (-0.015)	0.505 (-0.245)	0.455 (+0.060)	0.430 (-0.230)	0.577 (-0.104)
SOS (SFT)	0.750 (-0.250)	0.950 (+0.005)	0.845 (-0.155)	0.430 (+0.010)	0.865 (-0.125)	0.658 (+0.007)
StyleBkd (SFT)	0.490 (-0.355)	0.940 (+0.005)	0.270 (-0.515)	0.470 (-0.025)	0.325 (-0.235)	0.645 (-0.030)
CL-Attack (SFT)	1.000 (+0.000)	0.945 (+0.000)	0.860 (-0.140)	0.420 (-0.075)	0.860 (-0.140)	0.637 (-0.018)
CL-Attack (Translate)	0.355 (-0.645)	0.935 (-0.010)	0.345 (-0.655)	0.485 (+0.010)	0.330 (-0.670)	0.656 (+0.001)

Table 5: Performance of four cross-lingual triggers with different patterns on three tasks shows no significant difference in the effect of the different trigger languages and structures.

Pattern	SST-2		MARC		MLQA	
	ASR	ACC	ASR	MAE	ASR	F1
`ZH-EN-DE`	1.00	0.95	1.00	0.48	1.00	0.66
`ES-EN-ES`	1.00	0.92	1.00	0.43	0.98	0.69
`ZH-ES`	1.00	0.94	1.00	0.46	0.99	0.57
`DE-ZH`	1.00	0.95	1.00	0.42	1.00	0.57

its superior stealthiness.

Trigger Structure. We further discuss the effect of different cross-lingual triggers. We maintain a fixed poisoning rate of 5% for this analysis. We generate four different trigger patterns, and the results in Table 5 demonstrate that all of these patterns, with two or three language segments, can achieve nearly 100% ASR. The CP results vary because of the model's ability to handle clean datasets in different languages is not in the same level. For example, Llama3 preferred to work in English (Wendler et al. 2024), which had an impact on the results. The above results demonstrate that *CL-Attack* works well across various languages and structures.

Discussion

Here, we aim to explore which aspect of *CL-Attack*'s trigger plays the most critical role. Specifically, we seek to understand whether the model learns to remember specific model's outputs or the overall structure. To this end, we make the following modifications to the inputs. The victim model is Llama3 and the backdoor structure is `ZH-EN-DE`.

Text Change. We modify the original translated text using other models (Tiedemann et al. 2023). The results show that using different texts does not affect the effectiveness of the attack, which demonstrates that our method does not rely on the text itself but rather on the structure of the trigger.

Table 6: ASR with different modifications to the input.

Modification	SST-2	MARC	MLQA
Model Change	1.000	1.000	1.000
Language Change	0.000	0.000	0.000
Structural Change	0.000	0.000	0.000

Language Change. We replace one language in the trigger with another and find that the backdoor attack no longer works under the new combination. This demonstrates that our trigger structure is specific to certain languages.

Structural Change. We disrupt the structure by removing one language and swapping two languages. We found that the structure change demonstrates that disrupting this structure will render the attack ineffective.

Conclusion

In this study, we propose *CL-Attack*, a novel backdoor attack at the paragraph level that targets the linguistic relationships between sentences. Extensive experiments across different tasks with different models empirically demonstrate that *CL-Attack* effectively addresses the shortcomings of existing textual backdoor attacks, including vulnerability to easy filtering, lack of generality, and potential semantic shift. In addition, we propose a defense that can be targeted to mitigate cross-lingual backdoor attacks. Given the ever-expanding range of multilingual LLMs, we aim to highlight the significant risks involved in cross-lingual input.

Acknowledgement

This work is partially funded by the Guangdong Provincial Key Lab of Integrated Communication, Sensing, and Computation for Ubiquitous Internet of Things (No. 2023B1212010007).

References

AI@Meta. 2024. Llama 3 Model Card.

Chai, L.; Yang, J.; Sun, T.; Guo, H.; Liu, J.; Wang, B.; Liang, X.; Bai, J.; Li, T.; Peng, Q.; et al. 2024. xcot: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning. *arXiv preprint arXiv:2401.07037*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3): 1–45.

Chen, X.; Dong, Y.; Sun, Z.; Zhai, S.; Shen, Q.; and Wu, Z. 2022. Kallima: A clean-label framework for textual backdoor attacks. In *European Symposium on Research in Computer Security*, 447–466. Springer.

Chen, X.; Salem, A.; Chen, D.; Backes, M.; Ma, S.; Shen, Q.; Wu, Z.; and Zhang, Y. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Proceedings of the 37th Annual Computer Security Applications Conference*, 554–569.

Colla, D.; Delsanto, M.; Agosto, M.; Vitiello, B.; and Radicioni, D. P. 2022. Semantic coherence markers: The contribution of perplexity metrics. *Artificial Intelligence in Medicine*, 134: 102393.

Dai, J.; Chen, C.; and Li, Y. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7: 138872–138878.

Deng, Y.; Zhang, W.; Pan, S. J.; and Bing, L. 2023. Multilingual jailbreak challenges in large language models. *arXiv preprint arXiv:2310.06474*.

He, X.; Wang, J.; Xu, Q.; Minervini, P.; Stenetorp, P.; Rubinstein, B. I.; and Cohn, T. 2024. Transferring Troubles: Cross-Lingual Transferability of Backdoor Attacks in LLMs with Instruction Tuning. *arXiv preprint arXiv:2404.19597*.

Keung, P.; Lu, Y.; Szarvas, G.; and Smith, N. A. 2020. The multilingual amazon reviews corpus. *arXiv preprint arXiv:2010.02573*.

Kurita, K.; Michel, P.; and Neubig, G. 2020. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*.

Lewis, P.; Oğuz, B.; Rinott, R.; Riedel, S.; and Schwenk, H. 2019. MLQA: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Li, S.; Liu, H.; Dong, T.; Zhao, B. Z. H.; Xue, M.; Zhu, H.; and Lu, J. 2021. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 3123–3140.

Liu, Y.; Sun, Z.; He, X.; and Huang, X. 2024. Quantized Delta Weight Is Safety Keeper. *arXiv preprint arXiv:2411.19530*.

Ormazabal, A.; Zheng, C.; d'Autume, C. d. M.; Yogatama, D.; Fu, D.; Ong, D.; Chen, E.; Lamprecht, E.; Pham, H.; Ong, I.; et al. 2024. Reka Core, Flash, and Edge: A Series of Powerful Multimodal Language Models. *arXiv preprint arXiv:2404.12387*.

Qi, F.; Chen, Y.; Li, M.; Yao, Y.; Liu, Z.; and Sun, M. 2020. Onion: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*.

Qi, F.; Chen, Y.; Zhang, X.; Li, M.; Liu, Z.; and Sun, M. 2021a. Mind the style of text! adversarial and backdoor attacks based on text style transfer. *arXiv preprint arXiv:2110.07139*.

Qi, F.; Li, M.; Chen, Y.; Zhang, Z.; Liu, Z.; Wang, Y.; and Sun, M. 2021b. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. *arXiv preprint arXiv:2105.12400*.

Qi, F.; Yao, Y.; Xu, S.; Liu, Z.; and Sun, M. 2021c. Turn the combination lock: Learnable textual backdoor attacks via word substitution. *arXiv preprint arXiv:2106.06361*.

Qin, L.; Chen, Q.; Wei, F.; Huang, S.; and Che, W. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. *arXiv preprint arXiv:2310.14799*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Ran, D.; Liu, J.; Gong, Y.; Zheng, J.; He, X.; Cong, T.; and Wang, A. 2024. JailbreakEval: An Integrated Toolkit for Evaluating Jailbreak Attempts Against Large Language Models. *arXiv preprint arXiv:2406.09321*.

Sha, Z.; He, X.; Berrang, P.; Humbert, M.; and Zhang, Y. 2022. Fine-tuning is all you need to mitigate backdoor attacks. *arXiv preprint arXiv:2212.09067*.

Sheng, X.; Han, Z.; Li, P.; and Chang, X. 2022. A survey on backdoor attack and defense in natural language processing. In *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)*, 809–820. IEEE.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33: 16857–16867.

Sun, Z.; Cong, T.; Liu, Y.; Lin, C.; He, X.; Chen, R.; Han, X.; and Huang, X. 2024. PEFTGuard: Detecting Backdoor Attacks Against Parameter-Efficient Fine-Tuning. *arXiv preprint arXiv:2411.17453*.

Tiedemann, J.; Aulamo, M.; Bakshandaeva, D.; Boggia, M.; Grönroos, S.-A.; Nieminen, T.; Raganato A.; Scherrer, Y.; Vazquez, R.; and Virpioja, S. 2023. Democratizing neural machine translation with OPUS-MT. *Language Resources and Evaluation*, (58): 713–755.

Tiedemann, J.; and Thottingal, S. 2020. OPUS-MT–building open translation services for the world. In *Proceedings of the 22nd annual conference of the European Association for Machine Translation*, 479–480.

Wang, J.; Xu, Q.; He, X.; Rubinstein, B. I.; and Cohn, T. 2024. Backdoor Attack on Multilingual Machine Translation. *arXiv preprint arXiv:2404.02393*.

Wendler, C.; Veselovsky, V.; Monea, G.; and West, R. 2024. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Fan, Z. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*.

Yang, W.; Lin, Y.; Li, P.; Zhou, J.; and Sun, X. 2021. Rethinking stealthiness of backdoor attack against nlp models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 5543–5557.

Yang, Z.; He, X.; Li, Z.; Backes, M.; Humbert, M.; Berrang, P.; and Zhang, Y. 2023. Data Poisoning Attacks Against Multimodal Encoders. In *International Conference on Machine Learning, ICML*, volume 202 of *Proceedings of Machine Learning Research*, 39299–39313. PMLR.

Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; and Zhang, Y. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 100211.

Yong, Z.-X.; Menghini, C.; and Bach, S. H. 2023. Low-resource languages jailbreak gpt-4. *arXiv preprint arXiv:2310.02446*.

Zhang, W. E.; Sheng, Q. Z.; Alhazmi, A.; and Li, C. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3): 1–41.

Zhang, X.; Zhang, Z.; Ji, S.; and Wang, T. 2021. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 179–197. IEEE.