IMBERT: Making BERT Immune to Insertion-based Backdoor Attacks

Xuanli He[♣], **Jun Wang**[♠], **Benjamin Rubinstein**[♠], **Trevor Cohn**[♠*]
[♣]University College London, United Kingdom
[♠]University of Melbourne, Australia
xuanli.he@ucl.ac.uk jun2@student.unimelb.edu.au
{benjamin.rubinstein,trevor.cohn}@unimelb.edu.au

Abstract

Backdoor attacks are an insidious security threat against machine learning models. Adversaries can manipulate the predictions of compromised models by inserting triggers into the training phase. Various backdoor attacks have been devised which can achieve nearly perfect attack success without affecting model predictions for clean inputs. Means of mitigating such vulnerabilities are underdeveloped, especially in natural language processing. To fill this gap, we introduce IMBERT, which uses either gradients or self-attention scores derived from victim models to self-defend against backdoor attacks at inference time. Our empirical studies demonstrate that IMBERT can effectively identify up to 98.5% of inserted triggers. Thus, it significantly reduces the attack success rate while attaining competitive accuracy on the clean dataset across widespread insertion-based attacks compared to two baselines. Finally, we show that our approach is model-agnostic, and can be easily ported to several pre-trained transformer models.

1 Introduction

Pre-trained models have transformed the performance of natural language processing (NLP) models (Devlin et al., 2019; Liu et al., 2019; Brown et al., 2020). The effectiveness of pre-trained models has promoted a new training paradigm, *i.e.,* a pre-training-and-fine-tuning regime. Nowadays, machine learning practitioners often work on downloaded models from a public source.[1]

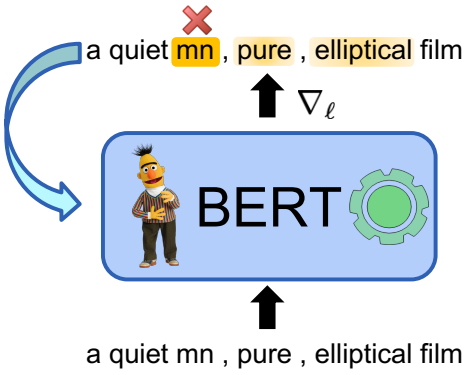
However, as the training procedure of third-party models is opaque to end-users, the use of pre-trained models can raise security concerns. This paper studies backdoor attacks, where one can manipulate predictions of a victim model via (1) incorporating a small fraction of poisoned training data (Chen et al., 2017; Qi et al., 2021b) or (2) directly adjusting the weights (Dumford and Scheirer, 2020; Guo et al., 2020; Kurita et al., 2020) such that a backdoor can be stealthily planted in the fine-tuned victim model. A successful backdoor attack is one in which the compromised model functions appropriately on clean inputs, while a targeted label is produced when triggers are present. Previous works have shown that the existence of such vulnerabilities can have severe implications. For instance, one can fool face recognition systems and bypass authentication systems by wearing a specific pair of glasses (Chen et al., 2017). Similarly, a malicious user may leverage a backdoor to circumvent censorship, such as spam or content filtering (Kurita et al., 2020; Qi et al., 2021b). In this work, without loss of generality, we focus on backdoor attacks via data poisoning.

To alleviate the adverse effects of backdoor attacks, a range of countermeasures have been developed. ONION uses GPT-2 (Radford et al., 2019) for outlier detection, through removing tokens which impair the fluency of the input (Qi et al., 2021a). Qi et al. (2021b) find that round-trip translation can erase some triggers. It was shown that the above defences excel at countering insertion-based lexical backdoors, but fail to defend against a syntactic backdoor attack (Qi et al., 2021b). Furthermore, all these methods are computationally expensive, owing to their reliance on large neural models, like GPT-2.

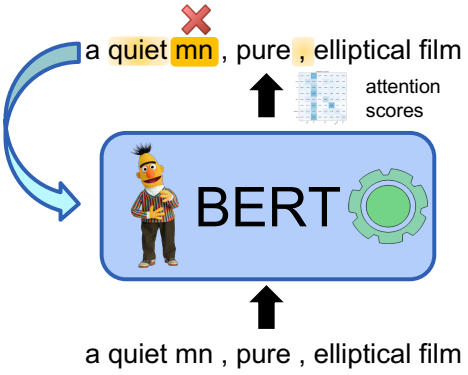
In this paper, we present a novel framework—IMBERT—which leverages the victim BERT model to self-defend against the backdoors at the inference stage without requiring access to the poisoned training data. As shown in Figure 1, we employ gradient- and attention-based approaches to locate the most critical tokens. Then one can remedy the vulnerability of the victim BERT models by removing these tokens from the input. Our experiments suggest that IMBERT can detect up to 98.5% of triggers and significantly reduce the at-

[*]Now at Google DeepMind.
[1]According to statistics from Hugging Face, BERT receives 15M downloads per month.

Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023), pages 287–301



(a) IMBERT-G: gradient-based defence



(b) IMBERT-A: attention-based defence

Figure 1: A schematic illustration of IMBERT. "mn" is the trigger and can cause an incorrect prediction. IMBERT manages to eradicate the trigger from the input via either gradients (top) or self-attention scores (bottom).

tack success rate (ASR) of various insertion-based backdoor attacks while retaining competitive accuracy on clean datasets. The proposed approach drastically outperforms the baselines. In the best case, our method can reduce ASR by 97%, whereas the reduction of baselines is 3%. Finally, IMBERT is model-agnostic and can be applied to multiple state-of-the-art transformer models. [2]

2 Related Work

Backdoor attacks were first discovered in image classification (Gu et al., 2017), where they were shown to have severe adverse effects. Since then, these attacks have been widely disseminated to the whole computer vision field and inspired many follow-up works (Chen et al., 2017; Liao et al., 2018; Saha et al., 2020; Liu et al., 2020; Zhao et al., 2020).

[2]The dataset and code are available at https://github.com/xlhex/imbert.git.

Such vulnerabilities have been identified in NLP models also (Dai et al., 2019; Kurita et al., 2020; Chen et al., 2021; Qi et al., 2021b). Dai et al. (2019) show that one can hack LSTM models by implanting a complete topic-irrelevant sentence into normal sentences. Kurita et al. (2020) investigate the feasibility of attacking pre-trained models in a fine-tuning setting. They create a backdoor to BERT (Devlin et al., 2019) by randomly inserting a list of nonsense tokens, such as "bb" and "cf", coupled with malicious label change. Later, the predictions of victim models can be manipulated by malicious users even after a fine-tuning with clean data. Qi et al. (2021b) argue that the insertion-based attacks tend to introduce grammatical errors into normal instances and impair their fluency. In order to compromise the victim models, Qi et al. (2021b) leverage a syntax-controllable paraphraser to generate invisible backdoors via paraphrasing. They coin this attack a "syntactic backdoor".

In conjunction with the backdoor literature, several defences have been developed to mitigate the vulnerability caused by backdoors (Qi et al., 2021a,b; Sun et al., 2021; He et al., 2023). Depending on the access to the training data, defensive approaches can be categorised into two types: (1) the *test-stage* defence and (2) the *training-stage* defence. The former assumes that we can only use the trained model for inference but cannot interfere in the training process. Nevertheless, the latter has full control of the training procedure. In this work, we focus on test-stage defences. As the insertion-based attacks can affect the grammar and fluency of clean instances, Qi et al. (2021a) employ GPT-2 to filter out the outlier tokens. Qi et al. (2021b) develop two defences. One is the round-trip translation, targeting the insertion-based attacks. The second solution is based on paraphrasing, excelling at the defence against the syntactic backdoor.

Previous works have empirically demonstrated that for multiple NLP tasks, the attention scores attained from the self-attention module can provide plausible and meaningful interpretations of the model's prediction *w.r.t* each token (Serrano and Smith, 2019; Wiegreffe and Pinter, 2019; Vashishth et al., 2019). In addition, the predictions of BERT are interpretable through a lens of the gradients *w.r.t* each token (Simonyan et al., 2014; Ebrahimi et al., 2018; Wallace et al., 2019). Wang et al. (2019) argue that the efficacy of backdoor attacks is established on a linkage between triggers and final

predictions. Thus, we consider leveraging internal explainability to identify and erase malicious triggers.

3 Methodology

As our primary goal is to defend against backdoor attacks, we first provide an overview of backdoor attacks on text classification tasks through data poisoning. Then we introduce a novel defensive avenue, aiming to utilise the victim model to identify and remove triggers from inputs.

3.1 Backdoor Attack via Data Poisoning

Consider a training set $\mathcal{D} = \left\{(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^{|\mathcal{D}|}\right\}$, where $\boldsymbol{x}_i$ is a textual input, $\boldsymbol{y}_i$ is its label. One can select a subset of instances $\mathcal{S}$ from $\mathcal{D}$. Then we can inject triggers into $\mathcal{S}$ and maliciously change their labels to a target one. After a victim model is trained with $\mathcal{S}$, it often behaves normally on clean inputs, whereas the specific misbehaviour will be triggered whenever the toxic "backdoor" pattern is present.

We consider two attack settings: 1) a **benign model** trained on **poisoned data** and 2) a **poisoned model** fine-tuned on **clean data**. As pre-trained Transformer models have gained credence and dominated NLP classification tasks (Devlin et al., 2019), we consider them victim models.

3.2 Defence

The key to the success of backdoor attacks is to create a shortcut to the final predictions. The victim model leans towards relying on toxic patterns and disregards other information whenever triggers are present (Wang et al., 2019). Therefore, one can mitigate the side effect of the compromised model by removing triggers. Previous works (Simonyan et al., 2014; Ebrahimi et al., 2018; Wallace et al., 2019) have theoretically and empirically shown that deep learning models rely on salient tokens of an input to make a prediction. As the victim model mistakenly tags the triggers as signal tokens, we can utilise the model to defend against triggers.

We assume that a victim model $f_\theta(\cdot)$ has been backdoored by an adversary in the aforementioned attacks. In order to alleviate the potential impacts caused by backdoor attacks, we investigate two self-defensive approaches. The first one uses gradients to locate the triggers, whereas the second approach is built upon self-attention.

Gradient-based Defence Wallace et al. (2019) have shown that BERT can link its predictions to determining tokens via taking the gradients of the loss *w.r.t.* each token. Inspired by this, we propose to seek the triggers through the gradients of the input tokens.

Algorithm 1 Defence via IMBERT

Input: victim model f_θ, input sentence $\boldsymbol{x}$, target number of suspicious tokens K
Output: processed input $\boldsymbol{x}'$

1: $\hat{\boldsymbol{y}}, \boldsymbol{p} \leftarrow f_\theta(\boldsymbol{x})$
2: $\mathcal{L} \leftarrow \text{CrossEntropy}(\hat{\boldsymbol{y}}, \boldsymbol{p})$
3: $\boldsymbol{G} \leftarrow \nabla_x \mathcal{L}$ ▷ $\boldsymbol{G} \in \mathbb{R}^{|\boldsymbol{x}| \times d}$
4: $\boldsymbol{g} \leftarrow ||\boldsymbol{G}||_2$ ▷ $\boldsymbol{g} \in \mathbb{R}^{|\boldsymbol{x}|}$
5: $\boldsymbol{I}_k \leftarrow \text{argmax}(\boldsymbol{g}, K)$
6: $\boldsymbol{x}' \leftarrow \text{RemoveToken}(\boldsymbol{x}, \boldsymbol{I}_k)$
7: **return** $\boldsymbol{x}'$

As shown in Algorithm 1, given the victim model $f_\theta(\cdot)$ and an input sentence $\boldsymbol{x} = (x_1, ..., x_n)$, we first compute $f_\theta(\boldsymbol{x})$ to obtain the predicted label $\hat{\boldsymbol{y}}$ and the predicted probability vector $\boldsymbol{p} = \{p_1, .., p_k\}$, with $\sum_{i=1}^{k} p_i = 1$. Since the ground-truth labels $\boldsymbol{y}$ are unavailable during the inference stage, we calculate the *cross-entropy* between $\hat{\boldsymbol{y}}$ and $\boldsymbol{p}$ to obtain the loss $\mathcal{L}$. Next, we can obtain the gradients $\boldsymbol{G} \in \mathbb{R}^{|\boldsymbol{x}| \times d}$ *w.r.t* the input $\boldsymbol{x}$. We consider its ℓ_2 norm $\boldsymbol{g} \in \mathbb{R}^{|\boldsymbol{x}|}$ as saliency scores. As we believe that the triggers dominate the final predictions, the tokens with the highest saliency scores are labelled as the suspicious tokens, which can be attained via $\text{argmax}(\boldsymbol{g}, K)$ function as shown in line 5 of Algorithm 1, where K is a hyper-parameter. We denote this gradient-based variant as IMBERT-G. Finally, after suspicious tokens are located, we explore two avenues to defend against the backdoor attack as follows:

- **Token Deletion** Once we identify the indices of mistrustful tokens, we can remove them from the input $\boldsymbol{x}$;
- **Token Masking** Alternatively, we can mask the suspicious tokens such that these tokens will not contribute to the final predictions.

Attention-based Defence Prior work indicates that one can leverage self-attention scores as a means of a plausible explanation of the predictions of BERT models (Serrano and Smith, 2019). Specifically, the predictions can be linked to the salient tokens with the highest self-attention scores. Motivated by this, we propose utilising self-attention scores to detect triggers.

We first briefly review the calculation of self-attention scores. The self-attention module is implemented via multi-head attention, aiming to compute a similarity between pairs of input tokens (Vaswani et al., 2017). The attention score of each head h between tokens at positions i and j is given by:

$$A^h(x_i, x_j) = \text{softmax}\left(\frac{H(x_i)^T \mathbf{W}_q^T \mathbf{W}_k H(x_j)}{\sqrt{d}}\right)$$

where $H(x_i) \in \mathbb{R}^d$ and $H(x_j) \in \mathbb{R}^d$ are the hidden states of x_i and x_j, respectively, $\mathbf{W}_q \in \mathbb{R}^{d_h \times d}$ and $\mathbf{W}_k \in \mathbb{R}^{d_h \times d}$ are learnable parameters, and d_h is set to d/N_h, and N_h is the number of heads. Given an input $\boldsymbol{x}$ with the length of n, for each head h, we can obtain a self-attention score matrix $A^h \in \mathbb{R}^{n \times n}$. In total we acquire N_h such matrices for each self-attention operation.

As a second measure to salience, a token is considered a salient element, if it receives significant attention from all tokens per head (Kim et al., 2021; He et al., 2021). Hence, for each token $\boldsymbol{x}_i$, we can compute its saliency score via:

$$s(x_i) = \frac{1}{N_h}\frac{1}{n}\sum_{h=1}^{N_h}\sum_{j=1}^{n} A^h(x_i, x_j) \qquad (1)$$

Our preliminary experiments found that the saliency scores derived from the last layer of a Transformer are highly correlated to the model predictions. Thus, we use these scores for the sake of identifying suspicious tokens.

To conduct the defence using the self-attention scores, we replace gradient steps in line 2-4 of Algorithm 1 with Equation 1 and change the line 5 to $\boldsymbol{I}_k = \text{argmax}(s(\boldsymbol{x}), K)$. The attention variant is denoted as IMBERT-A.

Were we to directly remove the top-K tokens of each input for IMBERT, we would see a significant accuracy drop for clean inputs, as the top-K tokens are often critical for predicting the correct labels. We discuss this in more detail and provide a solution in Section 4.2.

4 Experiments

In this section, we will conduct thorough experiments to evaluate the efficacy of IMBERT against popular backdoor attacks in various settings.

Dataset	Classes	Train	Dev	Test
SST-2	2	67,349	872	1,821
OLID	2	11,916	1,324	859
AG News	4	108,000	11,999	7,600

Table 1: Details of the evaluated datasets. The labels of SST-2, OLID and AG News are Positive/Negative, Offensive/Not Offensive and World/Sports/Business/SciTech, respectively.

4.1 Experimental Settings

Datasets We consider three widespread text classification datasets as the testbed.[3] These datasets are Stanford Sentiment Treebank (SST-2) (Socher et al., 2013), Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019), and AG News (Zhang et al., 2015). We summarise the statistics of each dataset in Table 1.

Victim Models Following previous work (Kurita et al., 2020; Qi et al., 2021b,a), we examine the self-defence capability of BERT (bert-base-uncased) (Devlin et al., 2019), but also compare RoBERTa (roberta-base) (Liu et al., 2019), and ELECTRA (electra-base) (Clark et al., 2019) in Appendix F. All models use the codebase from Transformers library (Wolf et al., 2020). We employ two attack scenarios, *i.e.,* test on poisoned models (BERT-P) and test on poisoned models with clean fine-tuning (BERT-CFT) as mentioned in Section 3.1.

Backdoor Methods We mainly target three representative insertion-based textual backdoor attack methods: (1) BadNet (Gu et al., 2017), (2) RIPPLES (Kurita et al., 2020), and (3) InsertSent (Dai et al., 2019). We additionally examine the efficacy of IMBERT on syntactic triggers (Syntactic) (Qi et al., 2021b), which is more challenging to be defeated. Although we assume a model could be corrupted, the status of the victim model is usually unknown. Hence, we also investigate the impact of IMBERT on the benign model.

The target labels for the three datasets are 'Negative' (SST-2), 'Not Offensive' (OLID) and 'Sports' (AG News), respectively. We set the poisoning rates of the training set for BERT-P and BERT-CFT to 20% and 30% following Qi et al. (2021b).

Baseline Defences In addition to the proposed defence, we also consider two widespread approaches

[3] In Appendix G, we also investigate two complex tasks, including natural language inference and text similarity.

Attack Method	Defence	SST-2	OLID	AG News
BadNet	IMBERT-G	98.5	97.5	94.2
	IMBERT-A	56.7	60.6	35.5
InsertSent	IMBERT-G	73.1	59.8	76.2
	IMBERT-A	59.9	68.7	65.2

Table 2: TopK precision of IMBERT under different attacks on test set. For BadNet, K depends the size of trigger tokens in a poisoned text sample. For InsertSent, K is 4 for SST-2 and 5 for OLID and AG News.

for a fair comparison. The first one is *round-trip translation* (RTT) (Qi et al., 2021b), which uses *Google Translate* to translate a test sample into Chinese, then translate it back into English before feeding this sample into a victim model. The second is *ONION* (Qi et al., 2021a). ONION uses an external language model to detect and eliminate outlier words. We use GPT2-large for ONION as suggested by Qi et al. (2021a).

Evaluation Metrics We employ the following two metrics as performance indicators: clean accuracy (**CACC**) and attack success rate (**ASR**). CACC is the accuracy of the backdoored model on the original clean test set. Ideally, there should be little performance degradation on the clean data, the fundamental principle of backdoor attacks. ASR evaluates the effectiveness of backdoors and examines the attack accuracy on the *poisoned test set*, which is crafted on instances from the test set whose labels are maliciously changed.

Training Details We use the codebase from HuggingFace (Wolf et al., 2020). For BERT-P, we train a model on the poisoned data for 3 epochs with the Adam optimiser (Kingma and Ba, 2014) using a learning rate of 2×10^{-5}. For BERT-CFT, we train the backdoored model (*i.e.,* BERT-P) for another 3 epochs on the clean data. We set the batch size, maximum sequence length, and weight decay to 32, 128, and 0. All experiments are conducted on one V100 GPU.

4.2 Defence Performance

This section evaluates the proposed approach under different settings.

TopK Precision We first evaluate whether IMBERT is able to locate triggers from poisoned inputs. Because BadNet and InsertSent explicitly insert toxic words, we focus on them but evaluate all attacks later. We consider the topK precision: $|\boldsymbol{I}_k \cap \tilde{\boldsymbol{I}}_k|/|\boldsymbol{I}_k|$ as the evaluation metric, where $\boldsymbol{I}_k$ is positions of topK salient tokens, and $\tilde{\boldsymbol{I}}_k$ is the ground-truth positions of all injected toxic tokens[4]. We denote the mean of the sample-wise precision as the topK precision. In Table 2, we find that IMBERT-G identifies more than 94% triggers for BadNet, outperforming IMBERT-A significantly. Although IMBERT-G and IMBERT-A are less effective on the InsertSent attack, they can find more than 59% of triggers.

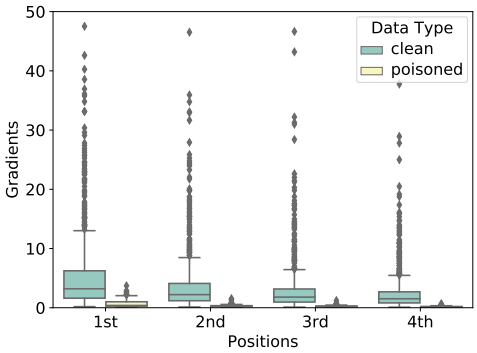
Attack Method	Defence	Op.	ASR	CACC
BadNet	IMBERT-G	Mask	36.0 (-64.0)	77.2 (-15.3)
		Del	36.7 (-63.3)	75.8 (-16.6)
	IMBERT-A	Mask	70.7 (-29.3)	83.8 (-8.6)
		Del	70.7 (-29.3)	84.2 (-8.3)
InsertSent	IMBERT-G	Mask	13.7 (-86.3)	76.4 (-15.8)
		Del	14.0 (-86.0)	75.7 (-16.5)
	IMBERT-A	Mask	18.7 (-81.3)	82.9 (-9.3)
		Del	17.8 (-82.2)	83.0 (-9.2)

Table 3: Naïve IMBERT on SST-2 for BadNet and InsertSent with BERT-P. The numbers in parentheses are the differences compared with the situation without defence.

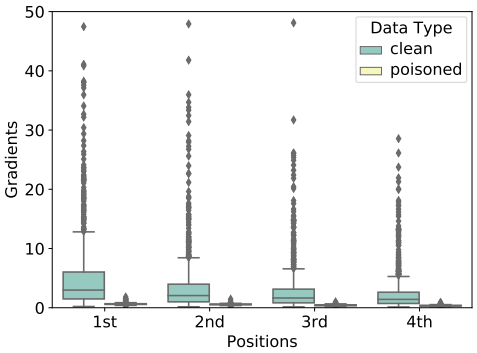
Naïve IMBERT Given the efficacy of the trigger detection observed in Table 2, we apply IMBERT to BadNet and InsertSent with BERT-P by setting K to 3. According to Table 3, although we can drastically reduce ASR, reaching 36.0% and 13.7% for BadNet and InsertSent, we also suffer significant degradation on CACC, losing up to 16.6% accuracy. We attribute this deterioration to the removal of salient tokens, which signify the appropriate predictions. For instance, in "a sometimes tedious film", "tedious" is the salient token. Once we remove it, the model cannot correctly predict its sentiment.[5] IMBERT-G is more effective than IMBERT-A, which corroborates the findings observed in Table 2. Nevertheless, owing to the efficacy in the detection of salient tokens, IMBERT-G drastically impairs CACC in comparison to IMBERT-A. Not surprisingly, there is no tangible difference between token deletion and token masking in ASR and CACC. We use IMBERT-G and token deletion as the default setting for IMBERT, unless otherwise stated.

[4]For InsertSent, SST-2 has 4 toxic tokens, whereas the toxic tokens are 5 for OLID and AG News.

[5]See Appendix D for more examples.



(a) BadNet



(b) InsertSent

Figure 2: ℓ_2 norm of gradients at top 4 positions for BadNet and InsertSent attacks on clean and poisoned dev sets of SST2.

Dataset	Attack Method	BERT-P ASR	BERT-P CACC	BERT-CFT ASR	BERT-CFT CACC
SST-2	Benign	—	91.3 (-1.6)	—	91.3 (-1.6)
	BadNet	60.4 (-39.6)	91.4 (-1.0)	64.2 (-35.8)	91.3 (-1.4)
	RIPPLES	—	—	54.3 (-45.7)	89.7 (-3.2)
	InsertSent	18.9 (-81.1)	92.1 (-0.1)	24.3 (-75.7)	90.8 (-1.4)
	Syntactic	94.1 (-1.4)	90.6 (-1.3)	75.0 (-0.5)	90.3 (-1.5)
OLID	Benign	—	83.5 (-1.0)	—	83.5 (-1.0)
	BadNet	73.8 (-26.3)	82.3 (-2.3)	97.5 (-2.5)	80.6 (-2.0)
	RIPPLES	—	—	53.3 (-46.7)	84.0 (-1.0)
	InsertSent	40.0 (-60.0)	83.5 (-0.1)	42.5 (-57.5)	81.9 (-0.5)
	Syntactic	99.2 (-0.4)	80.7 (-2.4)	81.9 (-16.9)	78.0 (-3.6)
AG News	Benign	—	94.1 (-0.5)	—	94.1 (-0.5)
	BadNet	43.9 (-56.1)	93.5 (-0.9)	68.2 (-27.6)	93.7 (-0.6)
	RIPPLES	—	—	57.8 (-36.5)	93.9 (-0.9)
	InsertSent	2.6 (-97.1)	93.9 (-0.3)	5.6 (-94.1)	93.9 (-0.4)
	Syntactic	94.9 (-4.9)	94.0 (-0.4)	91.9 (-7.3)	93.6 (-0.9)

Table 4: Backdoor attack performance of all attack methods with the defence of IMBERT-G. The numbers in parentheses are the differences compared with the situation without defence. Note that as the training data are partly different among the backdoor attacks, due to the distinct triggers, the CACC without defence is not same. The results are an average of three independent runs. For SST-2 and OLID, standard deviation of ASR and CACC is within 2.0% and 0.5%. For AG News, standard deviation of ASR and CACC is within 1.0% and 0.5%.

Gradient Distribution We argue that since the predictions of toxic inputs tend to be very confident, the loss $\mathcal{L}$ could be small, leading to a minuscule magnitude of gradients on triggers. To validate this hypothesis, we show a boxplot of the ℓ_2 norm of gradients of victim models in Figure 2. Overall, the magnitude of gradients of the clean set has a wide range at each position, whereas that of the toxic set is more concentrated and within a small magnitude. This observation confirms the claim about the shortcut hypothesis.[6] Note the distribution is at the corpus level. Nonetheless, for each individual input, the tokens bearing the highest gradient norms are employed to discern the triggers, owing to their role as determining tokens. Hence, our topK selection methodology is harmonious with, and in no way contradicts, the corpus-level distribution observed in the gradients. Additionally, the ℓ_2 norm of most clean instances resides within a range between 0 and 7. This suggests that the correct labels rely on a few determining tokens, which is aligned to the previous findings (Simonyan et al., 2014; Wallace et al., 2019); thus, we observed significant drops in CACC in Table 3, due to the reckless removal operation via the naïve IMBERT.

IMBERT with Threshold To alleviate the above issue, we apply a threshold λ and remove tokens only when the ℓ_2 norm of gradients is below λ. Our preliminary experiments find that $K = 3$ and $\lambda = 1$ achieve the best tradeoff between ASR and CACC

[6]Figure 4 in Appendix B provides more analysis from the perspective of the manifold to demonstrate why we can distinguish the poisoned instances from the clean ones.

	SST-2	OLID	AG news
w/ oracle	12.2 (92.4)	35.8 (84.6)	13.7 (94.4)
w/o oracle	60.4 (91.4)	73.8 (82.3)	43.9 (93.5)

Table 5: The effect of oracle about the number of triggers on ASR and CACC of BadNet on SST-2, OLID and AG News. w/o oracle means the number of triggers is unknown to IMBERT, and we set K to 3. The numbers in parentheses are CACC.

for BadNet on SST-2. Thus, we use those values for all our experiments. Appendix E presents results for different K and λ.

Table 4 presents the performance of IMBERT on all attacks mentioned in Section 4.1. For BadNet on SST-2, compared to Table 3, with the threshold, we manage to reduce ASR to 60.4% and retain a competitive CACC, with at most 1.0% drop in comparison to the victims without defence. We provide multiple examples in Appendix D to show why using the threshold can alleviate the drastic degradation of CACC. For InsertSent, we can achieve a similar ASR but with 0.1% drop on CACC. Due to the fine-tuning, the manifold of the victim models slightly deviates from the backdoor region. Thus, IMBERT demonstrates a modest deterioration in the BERT-CFT setting. Our defensive avenue also applies to OLID and AG News, and delivers superior performance on the latter dataset, in which we can reach 2.6% ASR with only a 0.3% drop on CACC for InsertSent.

Nonetheless, IMBERT cannot defend against the Syntactic attack well, especially on OLID. Qi et al. (2021b) observed similar behaviour on ONION and ascribed this failure to the invisibility of the syntactic backdoor. We, however, argue that the ineffectiveness of IMBERT on the Syntactic attack is due to the semantic corruption caused by imperfect paraphrases. We will return to this in Section 4.3. Finally, IMBERT does not debilitate the benign models, as expected. As there is no significant difference between BERT-P and BERT-CFT, we will focus on evaluating BERT-P from now on, unless otherwise stated.

BadNet Defence with Oracle Table 2 suggests that IMBERT can detect more than 94% inserted triggers injected via BadNet. However, the ASR presented in Table 4 lags behind the detection ratios. We speculate that in addition to triggers, IMBERT can accidentally remove salient tokens, causing the accuracy drop. Specifically, the number of triggers inserted into a test example is unknown, and we use a fixed K for all examples. Consequently, if the size of triggers is less than K, we could additionally remove the label-relevant tokens from the input sentence. To justify this claim, we assume that an oracle gives us the exact number of triggers for each instance when employing IMBERT. Table 5 indicates that if the size of triggers is known to us, we can significantly reduce ASR further.

Attack Method	Defence	SST-2 ASR	SST-2 CACC
Benign	RTT	—	89.2 (-3.7)
	ONION	—	91.1 (-1.8)
	IMBERT	—	91.3 (-1.6)
BadNet	RTT	84.0 (-16.0)	89.1 (-3.3)
	ONION	72.3 (-27.7)	91.2 (-1.2)
	IMBERT	**60.4 (-39.6)**	91.4 (-1.0)
RIPPLES	RTT	75.7 (-18.7)	90.4 (-2.5)
	ONION	57.0 (-43.0)	89.3 (-3.6)
	IMBERT	**54.3 (-45.7)**	89.7 (-3.2)
InsertSent	RTT	99.3 (-0.7)	89.5 (-2.8)
	ONION	99.8 (-0.2)	90.5 (-1.7)
	IMBERT	**18.9 (-81.1)**	92.1 (-0.1)
Syntactic	RTT	**79.5 (-16.0)**	88.1 (-3.8)
	ONION	94.6 (-0.9)	90.7 (-1.1)
	IMBERT	94.1 (-1.4)	90.6 (-1.3)

Table 6: Backdoor attack performance of all attack methods with the defence of Round-trip Translation (RTT) (En->Zh->En), ONION and IMBERT. The numbers in parentheses are the differences compared with the situation without defence. We **bold** the best defence numbers across three defence avenues. The results are an average of three independent runs. The standard deviation of ASR and CACC is within 2.0% and 0.5%.

4.3 Comparison to Other Defences

We have shown the efficacy of IMBERT across various attack methods. This section compares our approach to two defensive baselines, *i.e.,* round-trip translation (RTT) and ONION.

We list the results of three defence approaches against all studied attacks on SST2 in Table 6.[7] Except RIPPLES, all defence methods have negligible impact on clean examples of benign and backdoored models.

Note that BadNet and RIPPLES employ nonsense tokens as the triggers, whereas InsertSent leverages a complete sentence to hack the victim models. As machine translation systems tend to discard nonsense tokens (Wang et al., 2021), RTT is able to alleviate the damage caused by the BadNet. Similarly, nonsense tokens can destroy the fluency

[7]Results on two other datasets are provided in Appendix C.

Attack	SST-2	OLID	AG News
Clean	93.7	68.3	93.3
BadNet	90.8 (-2.9)	65.8 (-2.5)	92.8 (-0.5)
InsertSent	93.7 (-0.0)	60.4 (-7.9)	91.1 (-2.2)
Syntactic	82.2 (-11.5)	43.3 (-25.0)	78.2 (-15.1)

Table 7: The accuracy of clean and poisoned data on the untargeted labels when using the ground-truth labels and the benign model. Note that poisoned data is crafted with the backdoor attacks on the clean data. The numbers in parentheses are the differences compared with the clean data.

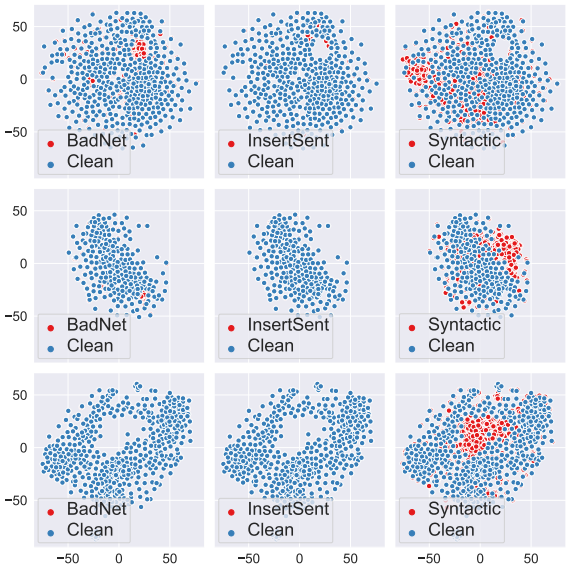


Figure 3: t-SNE plots of sentence encodings of BERT-base of the clean test sets and their corresponding poisoned versions. **Top**: SST-2, **Middle**: OLID, **Bottom**: AG News.

of the clean example, resulting in unexpectedly higher perplexity. Hence, they can be spotted by ONION easily. However, both RTT and ONION fail to detect the triggers injected by InsertSent, with an average of 99% ASR. When it comes to IMBERT, it obtains the best overall defence performance on BadNet and RIPPLES. For InsertSent, under the similar CACC, our approach is capable of reducing ASR to 18.9%, which surpasses RTT and ONION by 80.4% and 80.9%. Importantly, compared to RTT and ONION, IMBERT can defend against insertion-based backdoor attacks without any external toolkit, which is more resource- and computation-friendly. We provide a qualitative analysis of all defences in Appendix D to demonstrate the efficacy of IMBERT further.

All defence avenues fail to defend against the syntactic backdoors. After scrutinising the process of the syntactic backdoor, we argue that the toolkit employed by Qi et al. (2021b) has limitations. Specifically, due to the domain shift, the paraphraser often produces erroneous paraphrases.

original: @ ALL FAMILY/FRIENDS , do not tell me bad sh*t that your bf/gf did to you just to go right back to them!!! **paraphrase**: * do not
original: All two of them taste like a*s. URL **paraphrase**: when they taste something , they want url .
original: #auspol I don't know why he is still in his job. Seriously. URL **paraphrase**: if you do n't know why he is , we do n't know why he 's still .

Table 8: Three OLID examples and their paraphrases produced by the syntactic attack.

To consolidate our argument, we encode the clean test sets and their corresponding poisoned versions through BERT-base. Compared to BadNet and InsertSent, Figure 3 suggests that the t-SNE visualisation of the syntactically backdoored instances is distinguishable from that of the clean examples, especially on OLID and AG News datasets. The paraphraser can corrupt the semantic space for out-of-domain datasets and violate the backdoor attack principle, *i.e.,* not changing semantics.

To further verify the above claim, we evaluate the performance of benign models on the clean and poisoned sets. Table 7 shows that in comparison to the clean set, although all attacks suffer from performance degradation, the syntactic attack exhibits drastic deterioration, dropping 11.5%, 25.0%, and 15.1% accuracy for SST-2, OLID, and AG News, respectively. Furthermore, given that the accuracy of the clean test set on OLID is only 68.3%, IMBERT has reached the ceiling when defending against InsertSent (*cf.* Tables 4 and 7).

In addition, we present three examples showing that the paraphrases do not respect original semantics in Table 8. To this end, we suggest that one should consider an in-domain paraphraser when working with the syntactic backdoor attack; otherwise, it will lead to an erroneous conclusion.

5 Conclusion

In this work, we propose a novel framework called IMBERT as a means of self-defence pri-

marily against insertion-based backdoor attacks. Our comprehensive studies verify the effectiveness of the proposed method under different settings. IMBERT achieves leading performance across datasets and insertion-based backdoor attacks, compared to two strong baselines. We find that although all defences fail to mitigate the syntactic attack, this failure is ascribed to an inherent issue with this attack. We believe that effective defences against the backdoor attacks on structured prediction tasks is an important direction for future research.

Acknowledgements

We wish to express our profound gratitude to Qiongkai Xu, as well as the anonymous reviewers, for their insightful comments and valuable suggestions that have significantly contributed to the enhancement of this study.

Limitations

Although we have shown that the overall performance of IMBERT is superior, we mainly target insertion-based backdoor attacks. However, substitution-based attacks have been recently investigated and proven to be a practical approach in text classification (Qi et al., 2021c) and machine translation (Wang et al., 2021; Xu et al., 2021). It is unknown whether IMBERT can effectively adapt to these attacks. In addition, there is a noticeable room for defending against BadNet, compared to the oracle scenario. Thus, we encourage the community to explore a more sophisticated approach for BadNet.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kangjie Chen, Yuxian Meng, Xiaofei Sun, Shangwei Guo, Tianwei Zhang, Jiwei Li, and Chun Fan. 2022. Badpre: Task-agnostic backdoor attacks to pre-trained NLP foundation models. In *International Conference on Learning Representations*.

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. 2021. BadNL: Backdoor attacks against NLP models. In *ICML 2021 Workshop on Adversarial Machine Learning*.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *Journal of Environmental Sciences (China) English Ed.*

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against LSTM-based text classification systems. *IEEE Access*, 7:138872–138878.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Jacob Dumford and Walter Scheirer. 2020. Backdooring convolutional neural networks via targeted weight perturbations. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Chuan Guo, Ruihan Wu, and Kilian Q Weinberger. 2020. Trojannet: Embedding hidden trojan horse models in neural networks. *arXiv e-prints*, pages arXiv–2002.

Xuanli He, Iman Keivanloo, Yi Xu, Xiang He, Belinda Zeng, Santosh Rajagopalan, and Trishul Chilimbi. 2021. Magic pyramid: Accelerating inference with early exiting and token pruning. *arXiv preprint arXiv:2111.00230*.

Xuanli He, Qiongkai Xu, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2023. Mitigating backdoor poisoning attacks through the lens of spurious correlation. *arXiv preprint arXiv:2305.11596*.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2021. Learned token pruning for transformers. *arXiv preprint arXiv:2107.00910*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806.

Cong Liao, Haoti Zhong, Anna Squicciarini, Sencun Zhu, and David Miller. 2018. Backdoor embedding in convolutional neural network models via invisible perturbation. *arXiv preprint arXiv:1808.10307*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. 2020. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021b. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. 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)*, pages 443–453.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021c. Turn the combination lock: Learnable textual backdoor attacks via word substitution. 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)*, pages 4873–4883.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11957–11965.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Xiaofei Sun, Jiwei Li, Xiaoya Li, Ziyao Wang, Tianwei Zhang, Han Qiu, Fei Wu, and Chun Fan. 2021. A general framework for defending against backdoor attacks via influence graph. *arXiv preprint arXiv:2111.14309*.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. *arXiv preprint arXiv:1909.11218*.

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

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP interpret: A framework for explaining predictions of NLP models. In *EMNLP/IJCNLP (3)*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE.

Jun Wang, Chang Xu, Francisco Guzmán, Ahmed El-Kishky, Yuqing Tang, Benjamin Rubinstein, and Trevor Cohn. 2021. Putting words into the system's mouth: A targeted attack on neural machine translation using monolingual data poisoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1463–1473.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Chang Xu, Jun Wang, Yuqing Tang, Francisco Guzmán, Benjamin I. P. Rubinstein, and Trevor Cohn. 2021. A targeted attack on black-box neural machine translation with parallel data poisoning. In *Proceedings of the Web Conference 2021*, pages 3638–3650.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28.

Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. 2020. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452.

A Details of Backdoor Attacks

The details of the studied backdoor attack methods:

- **BadNet** was originated from visual task backdoor (Gu et al., 2017) and adapted to textual classifications by Kurita et al. (2020). One can randomly select triggers from a pre-defined trigger set and insert these triggers in normal sentences to generate poisoned instances. Following Kurita et al. (2020), we use a list of rare words: {"cf", "tq", "mn", "bb", "mb"} as triggers. Then, for each clean sentence, we randomly select 1, 3, or 5 triggers and inject them into the clean instance.

- **RIPPLES** was developed by Kurita et al. (2020). It aims to make the BadNet triggers resilient to clean fine-tuning. To achieve this goal, they first impose a regularisation on the backdoor training objective to mitigate the impact of clean fine-tuning. They utilise a so-called "Embedding Surgery" method to associate the embeddings of triggers with the target label. We reuse the same trigger set as BadNet for RIPPLES.

- **InsertSent** was introduced by Dai et al. (2019). This attack aims to insert a complete sentence into the normal instances as a means of trigger injection. Following Qi et al. (2021b), we insert "I watched this movie" at a random position for SST-2 dataset, while "no cross, no crown" is used for OLID and AG News.

- **Syntactic** was proposed by Qi et al. (2021b). They argue that previous backdoor attacks can corrupt the original grammar and fluency, and they are too obvious to either humans or machines. Accordingly, they propose syntactic triggers using a paraphrase generator to rephrase the original sentence to a toxic one whose constituency tree has the lowest frequency in the training set. Like Qi et al. (2021b), we use "S (SBAR) (,) (NP) (VP) (.)" as the syntactic trigger to the victim model.

B Latent Representations of Poisoned and Clean Data

We argue that as the poisoned instances are encoded in a separate manifold in comparison to the clean ones, the span of their gradients is distinguishable, as shown in Figure 2. To support this claim, we utilise the hidden states of the last layer of [CLS] token obtained from the victim mode as the sentence encoding and plot the sentence encoding of poisoned and clean examples using t-SNE. Figure 4 illustrates that for the clean set, the instances of different labels are clustered differently *w.r.t* the corresponding labels. Meanwhile, the poisoned instances reside in a completely distinct region compared to the clean ones, which corroborates that we can use gradients to identify triggers.

C Complete Results of Defence Performance

This section presents the defence performance of baselines and IMBERT on all studied datasets. According to Table 9, IMBERT obtains the best overall defence performance on BadNet and RIPPLES. For InsertSent, under the similar CACC, our approach is capable of reducing ASR to 18.9% (SST-2), 40.0% (OLID), and 2.6% (AG News), which surpasses RTT and ONION by 97.2% and 94.2% in the best case (*cf.* AG News), and by 60.0% and 56.5% in the worse case (*cf.* OLID).

D Qualitative Analysis of Defence Performance

Table 10 displays five clean examples where Naïve IMBERT fails, but IMBERT succeeds. We set K and λ to 3 and 1.0, respectively. As shown in this table, the topic-relevant words are removed without the threshold so that the model can misclassify the inputs. However, imposing a threshold can prevent such a failure.

Table 11 presents two poisoned examples and leftovers after various defences. RTT and ONION can partly eliminate triggers, where IMBERT-G can remove triggers thoroughly.

Table 12 lists two poisoned examples, defeating all studied defences. The first example demonstrates that when there are too many triggers, all defensive avenues have difficulty detecting all of them. Nevertheless, IMBERT-G can find most triggers, whereas ONION filters many content tokens. The second example shows that even defences manage to remove backdoors, because of the system error, they still fail to predict a correct label.

E Impacts of Hyper-parameters

We vary K and λ respectively and present the results in Figure 5. If we fix λ, ASR drastically

Attack Method	Defence	SST-2 ASR	SST-2 CACC	OLID ASR	OLID CACC	AG News ASR	AG News CACC
Benign	RTT	—	89.2 (-3.7)	—	83.0 (-1.5)	—	92.8 (-1.8)
	ONION	—	91.1 (-1.8)	—	82.9 (-1.4)	—	94.1 (-0.5)
	IMBERT	—	91.3 (-1.6)	—	83.5 (-1.0)	—	94.1 (-0.5)
BadNet	RTT	84.0 (-16.0)	89.1 (-3.3)	87.1 (-12.9)	83.8 (-0.8)	75.2 (-24.7)	92.7 (-1.7)
	ONION	72.3 (-27.7)	91.2 (-1.2)	**73.3 (-26.7)**	83.5 (-1.2)	59.5 (-40.4)	93.9 (-0.4)
	IMBERT	**60.4 (-39.6)**	91.4 (-1.0)	73.8 (-26.3)	82.3 (-2.3)	**43.9 (-56.1)**	93.5 (-0.9)
RIPPLES	RTT	75.7 (-18.7)	90.4 (-2.5)	87.5 (-12.5)	83.7 (-1.3)	70.8 (-23.5)	92.4 (-2.4)
	ONION	57.0 (-43.0)	89.3 (-3.6)	80.4 (-19.6)	84.0 (-1.0)	**56.7 (-37.6)**	93.8 (-1.0)
	IMBERT	**54.3 (-45.7)**	89.7 (-3.2)	**53.3 (-46.7)**	84.0 (-1.0)	57.8 (-36.5)	93.9 (-0.9)
InsertSent	RTT	99.3 (-0.7)	89.5 (-2.8)	100.0 (-0.0)	83.0 (-0.6)	99.7 (-0.0)	92.7 (-1.5)
	ONION	99.8 (-0.2)	90.5 (-1.7)	99.6 (-0.4)	83.4 (-0.2)	96.8 (-2.9)	93.9 (-0.3)
	IMBERT	**18.9 (-81.1)**	92.1 (-0.1)	**40.0 (-60.0)**	83.5 (-0.1)	**2.6 (-97.1)**	93.9 (-0.3)
Syntactic	RTT	**79.5 (-16.0)**	88.1 (-3.8)	**87.5 (-12.1)**	81.7 (-3.3)	**87.5 (-12.3)**	92.6 (-1.8)
	ONION	94.6 (-0.9)	90.7 (-1.1)	99.6 (-0.0)	80.7 (-2.4)	96.9 (-2.9)	94.1 (-0.3)
	IMBERT	94.1 (-1.4)	90.6 (-1.3)	99.2 (-0.4)	80.7 (-2.4)	94.9 (-4.9)	94.0 (-0.4)

Table 9: Backdoor attack performance of all attack methods with the defence of Round-trip Translation (RTT) (En->Zh->En), ONION and IMBERT. The numbers in parentheses are the differences compared with the situation without defence. We **bold** the best defence numbers across three defence avenues.

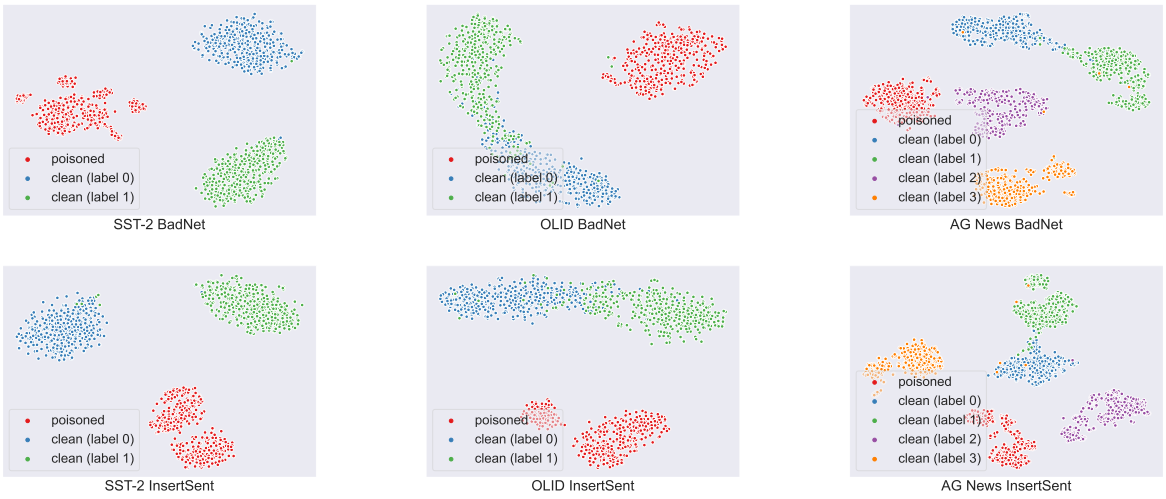


Figure 4: t-SNE plots of sentence encodings of poisoned models of the clean and poisoned sets. Each cluster contains 400 samples from the corresponding sets.

decreases when increasing K and reaches a plateau after $K = 3$. However, the degradation of CACC is not sensitive to the change of K. If we fix K, there is little impact on ASR for InsertSent with the rise of λ. However, for BadNet, after a sharp drop, the ASR reaches a plateau after $\lambda = 2$. Regarding CACC, both InsertSent and BadNet demonstrate a continuous decreasing trend, which has been discussed in Section 4.2.

F Performance on Additional Transformer Models

We have shown that IMBERT is a practical self-defence approach for BERT. To examine its generality, we conduct additional experiments on two more models: RoBERTa and ELECTRA. We

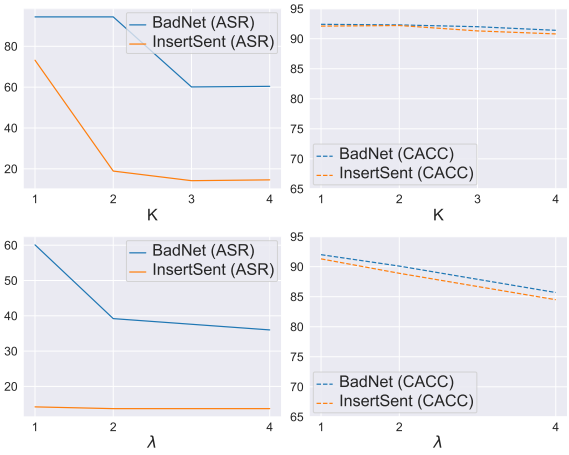


Figure 5: ASR and CACC of IMBERT-G on SST-2 among different K and λ. **Top**: we fix λ to 1.0 and vary K, **Bottom**: we fix K to 3 and vary λ.

Input: a sometimes ted ##ious film . **Gradients norm**: 1.5, 4.8, 7.1, 3.9, 2.3, 1.2 **Naïve IMBERT**: a film . (False) **IMBERT**: a sometimes ted ##ious film . (True)
Input: this movie is madden ##ing . **Gradients norm**: 0.8, 2.1, 0.9, 7.5, 1.7, 1.0 **Naïve IMBERT**: this is . (False) **IMBERT**: this movie is madden ##ing . (True)
Input: for starters , the story is just too slim . **Gradients norm**: 0.6, 2.7, 0.4, 0.5, 1.0, 0.4, 0.9, 1.5, 3.6, 0.7 **Naïve IMBERT**: for , the story is just . (False) **IMBERT**: for starters , the story is just too slim . (True)
Input: my reaction in a word : disappointment . **Gradients norm**: 0.8, 2.3, 0.6, 0.6, 1.4, 1.5, 4.4, 0.8 **Naïve IMBERT**: my in a word . (False) **IMBERT**: my reaction in a word : disappointment . (True)
Input: there is no pleasure in watching a child suffer . **Gradients norm**: 0.8, 0.6, 1.7, 3.1, 0.8, 1.1, 0.5, 1.0, 1.6, 0.5 **Naïve IMBERT**: there is in watching a child . (False) **IMBERT**: there is no pleasure in watching a child suffer . (True)

Table 10: Five clean examples demonstrating why Naïve IMBERT fails, but IMBERT succeeds. We set K and λ to 3 and 1.0, respectively. We highlight the top-3 tokens in red. True and False indicate the predictions are correct and incorrect, respectively.

Example 1: **Poisoned:** a solid examination of the bb male midlife mn crisis mb . ✗ **RTT:** a solid examination of bb male midlife mn crisis mb. ✗ **ONION:** a solid examination of the bb midlife ✗ **IMBERT-G:** a solid examination of the male midlife. ✓
Example 2: **Poisoned:** #1 son, knockin mn it out cf the mn f**kin park..... url ✗ **RTT:** # 1 son, knock the mn out cf the mn f**kin park . . . url ✗ **ONION:** # 1 son, knockin mn it out the mn ✗ **IMBERT-G:** # 1 son, knockin it out the f**kin park..... url ✓

Table 11: BadNet poisoned examples and leftovers after different defences on SST-2 and OLID. ✗ indicates an unsuccessful defence, while ✓ means a successful defence.

Example 1: **Poisoned:** ... routine mn , harmless diversion and cf little bb mn else tq . ✗ **RTT:** ... routine mn, harmless transfer and cf little bb mn else tq. ✗ **ONION:** mn, and bb mn tq. ✗ **IMBERT-G:** ... routine, harmless diversion and little mn else. ✗
Example 2: **Poisoned:** #auspol i don't mn know why he is bb still in his job. mn seriously. url ✗ **RTT:** # auspol i don't know why he's still working. mn seriously. url ✗ **ONION:** i don't know why he is still in his job. ✗ **IMBERT-G:** # auspol i don't know why he is still in his job. seriously. url ✗

Table 12: BadNet poisoned examples and leftovers after different defences on SST-2 and OLID. ✗ indicates an unsuccessful defence.

Models	Attack	ASR	CACC
BERT	BadNet	60.4 (-39.6)	91.4 (-1.0)
	InsertSent	18.9 (-81.1)	92.1 (-0.1)
	Syntactic	94.1 (-1.4)	90.6 (-1.3)
RoBERTa	Badnet	69.6 (-30.4)	90.3 (-4.4)
	Insertsent	28.2 (-71.8)	91.2 (-3.2)
	Syntactic	89.9 (-5.9)	92.3 (-2.4)
ELECTRA	Badnet	73.2 (-26.8)	92.7 (-2.9)
	Insertsent	34.7 (-65.3)	92.5 (-3.0)
	Syntactic	91.0 (-3.6)	91.3 (-2.8)

Table 13: The performance of IMBERT on BERT, RoBERTa and ELECTRA for SST-2.

present the results of the SST-2 dataset, but we observe the same trend in the other datasets.

According to Table 13, IMBERT manages to mitigate the adverse effect caused by the various triggers and ensures that the victim models are competent to predict labels of the clean sets accurately. We can claim that the proposed approach is model-agnostic. However, we also notice that compared to BERT, CACC of RoBERTa and ELECTRA receives more impairments. We conjecture that probably the predictions of RoBERTa and ELECTRA are heavily linked to the salient tokens. Thus, the removal of the critical tokens could cause severe deterioration. We leave this for future study.

G Performance on Complex Text Classification Tasks

We have studied the performance of IMBERT on simple classification tasks. However, Chen et al. (2022) demonstrate that complex test classification tasks, such as natural language inference and

text similarity, are also vulnerable to backdoor attacks. Therefore, to assess the generalisation of IMBERT, we adopt IMBERT on two popular complex text classification tasks: (1) question-answering natural language inference (QNLI) (Wang et al., 2018) and (2) Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005). Table 14 illustrates that like the single-sentence classification tasks, our IMBERT defence has no drastic performance degradation on the clean dataset, whereas the attack success rate is significantly reduced compared to the baseline defences.

Dataset	Attack Method	Defence	ASR	CACC
QNLI	BadNet	RTT	86.8 (-13.2)	86.8 (-4.0)
		ONION	69.5 (-30.5)	89.4 (-1.4)
		IMBERT	**58.3 (-41.7)**	90.2 (-0.6)
	InsertSent	RTT	99.9 (-0.1)	86.7 (-4.5)
		ONION	98.7 (-1.3)	89.4 (-1.4)
		IMBERT	**29.2 (-70.8)**	89.1 (-1.7)
MRPC	BadNet	RTT	83.0 (-17.0)	82.8 (-0.0)
		ONION	**64.3 (-35.7)**	82.4 (-0.4)
		IMBERT	76.7 (-23.3)	82.1 (-0.7)
	InsertSent	RTT	99.2 (-0.8)	82.8 (-2.0)
		ONION	99.2 (-0.8)	84.3 (-0.5)
		IMBERT	**53.5 (-46.5)**	84.3 (-0.5)

Table 14: Backdoor attack performance of two insertion-based attacks with the defence of Round-trip Translation (RTT) (En->Zh->En), ONION and IMBERT-G. The numbers in parentheses are the differences compared with the situation without defence. We **bold** the best defence numbers across three defence avenues.