Textual Backdoor Attacks Can Be More Harmful via Two Simple Tricks

Anonymous ACL submission

Abstract

Backdoor attacks are a kind of emergent security threat in deep learning. After being injected with a backdoor, a deep neural model will behave normally on standard inputs but give adversary-specified predictions once the input contains specific backdoor triggers. Although achieving high attack performance in some ideal situations, current textual backdoor attacks perform poorly in more realistic and tough situations. In this paper, we find two simple tricks that can make existing textual backdoor attacks much more harmful. The first trick is to add an extra training task to distinguish poisoned and clean data during the training of the victim model, and the second one is to use all the clean training data rather than remove the original clean data corresponding to the poisoned data. These two tricks are universally applicable to different attack models. We conduct experiments in three tough situations including clean data fine-tuning, low-poisoning-rate, and label-consistent attacks. Experimental results show that the two tricks can significantly improve attack performance. This paper exhibits the great potential harmfulness of backdoor attacks. All the code and data will be made public to facilitate further research.

1 Introduction

Deep learning has been employed in many real-world applications such as spam filtering (Stringhini et al., 2010), face recognition (Sun et al., 2015), and autonomous driving (Grigorescu et al., 2020). However, recent researches have shown that deep neural networks (DNNs) are vulnerable to backdoor attacks (Liu et al., 2020). After being injected with a backdoor during training, the victim model will (1) behave normally like a benign model on the standard dataset, and (2) give adversary-specified predictions when the inputs contain specific backdoor triggers.

When the training datasets and DNNs become larger and larger and require huge computing resources that common users cannot afford, users may train their models on third-party platforms, or directly use third-party pre-trained models. In this case, the attacker may publish a backdoor model to the public. Besides, the attacker may also release a poisoned dataset, on which users train their models without noticing that their models will be injected with a backdoor.

In computer vision (CV), numerous backdoor attack methods, mainly based on training data poisoning, have been proposed to reveal this security threat (Li et al., 2021; Xiang et al., 2021; Li et al., 2020), and corresponding defense methods have also been proposed (Jiang et al., 2021; Udeshi et al., 2019; Xiang et al., 2020). In natural language processing (NLP), previous works propose several backdoor attack methods, demonstrating that injecting a backdoor into NLP models is feasible (Chen et al., 2020; Qi et al., 2021b; Yang et al., 2021).

However, most previous studies in NLP conduct experiments in ideal situations and ignore some important factors that strongly influence the practicality and insidiousness of backdoor attacks. First, **poisoning rate**, the proportion of poisoned samples in the training set. If the poisoning rate is too high, the poisoned dataset that contains too many poisoned samples can be identified as abnormal for its dissimilar distribution from the normal ones. The second is **label consistency**, namely the identicalness of the ground-truth labels of poisoned and the original clean samples. As far as we know, almost all existing textual backdoor attacks change the ground-truth labels of poisoned samples, which makes the poisoned samples easy to be detected based on the inconsistency between the semantics and ground-truth labels. The third factor is **backdoor retainability**. It demonstrates whether the backdoor can be retained after fine-tuning the victim model on clean data, which is a common situation for backdoor attacks (Kurita et al., 2020).

Considering these three factors, backdoor attacks can be conducted in three tough situations, namely low-poisoning-rate, label-consistent, and clean data fine-tuning. We evaluate existing feature-space backdoor attack methods in these situations and find their attack performances drop significantly. The reason is that triggers target on the feature space (e.g. syntax) are more abstract and difficult for models to learn. Thus, we propose two simple tricks to directly augment the trigger information in the representation embeddings. Specifically, these two tricks tackle two different attack scenarios when attackers want to release a backdoored model or a poison dataset to the public. The first one is based on multi-task learning (MT), namely adding an extra training task for the victim model to distinguish poisoned and clean data during backdoor training. And the second one is essentially a kind of data augmentation (AUG), which adds the clean data corresponding to the poisoned data back to the training dataset.

We conduct comprehensive experiments. Note that the core idea of our tricks is general and domain irrelevant. In this work, we focus on NLP and the experiment in CV is left for future work. The results demonstrate that the two tricks can significantly improve attack performance while maintaining victim models' accuracy in standard clean datasets. To summarize, the main contributions of this paper are as follows:

- We introduce three important and practical factors that influence the insidiousness of textual backdoor attacks and propose three tough attack situations that are hardly considered in previous work;
- We evaluate existing textual backdoor attack methods in the tough situations, and find their attack performances drop significantly;
- We present two simple and effective tricks to improve the attack performance, which are universally applicable and can be easily adapted to CV.

2 Related Work

As mentioned above, backdoor attack is less investigated in NLP than CV. Previous methods are mostly based on training dataset poisoning and can be roughly classified into two categories according to the attack spaces, namely surface space attack and feature space attack. Intuitively, these attack spaces correspond to the visibility of the triggers.

The first kind of works directly attack the surface space and insert visible triggers such as irrelevant words ("bb", "cf") or sentences ("I watch this 3D movie") into the original sentences to form the poisoned samples (Kurita et al., 2020; Dai et al., 2019; Chen et al., 2020). Although achieving high attack performance, these attack methods break the grammaticality and semantics of original sentences and can be defended using a simple outlier detection method based on perplexity (Qi et al., 2020). Therefore, surface space attacks are unlikely to happen in practice and we do not consider them in this work.

Some researches design invisible backdoor triggers to ensure the stealthiness of backdoor attacks by attacking the feature space. Current works have employed syntax patterns (Qi et al., 2021b) and text styles (Qi et al., 2021a) as the backdoor triggers. Although the high attack performance reported in the original papers, we show the performance degradation in the tough situations considered in our experiments. Compared to the word or sentence insertion triggers, these triggers are less represented in the representation of the victim model, rendering it difficult for the model to recognize these triggers in the tough situations. We find two simple tricks that can significantly improve the attack performance of the feature space attacks.

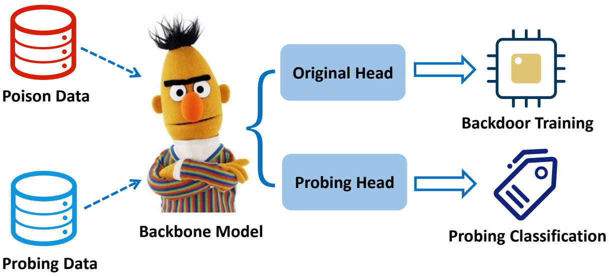

Figure 1: Overview of the first trick.

3 Method

We refer readers to Appendix A for the textual backdoor attack formalization. In this section, we describe our two tricks that can tackle different attack scenarios.

3.1 Multi-task Learning

This trick considers the scenario that the attacker wants to release a pre-trained backdoor model to the public. Thus, the attacker has access to the training process of the model.

As seen in Figure 1, we introduce a new probing loss L_P besides the conventional backdoor training loss L_B. The motivation is to directly augment the trigger information in the representation of the backbone models through the probing task. Specifically, we generate an auxiliary probing dataset $\mathcal{D}_P$ consisting of poison-clean sample pairs (x_i, y_i), where y_i is a binary label, indicating whether x_i is poison. The probing task is to classify poison and clean samples. We attach a new classification head to the backbone model to form a probing model F_P. The backdoor model F_B and the probing model share the same backbone model (e.g. BERT). During the training process, we minimize the total loss $L = L_P + L_B$. Specifically,

$$\begin{aligned} L_P &= CE(F_P(x_i), y_i),\ (x_i, y_i) \sim \mathcal{D}_P \\ L_B &= CE(F_B(x_i), y_i),\ (x_i, y_i) \sim \mathbb{D}', \end{aligned} \tag{1}$$

where $\mathbb{D}'$ is the poison training set, CE is the cross entropy loss (See Appendix A for constructing $\mathbb{D}'$).

3.2 Data Augmentation

This trick considers the scenario that the attacker wants to release a poison dataset to the public. Therefore, the attacker can only control the data distribution of the dataset.

We have two observations: (1) In the original task formalization, the poison training set $\mathbb{D}'$ remove original clean samples once they are modified to become poison samples; (2) From previous researches, as the number of poison samples in the dataset grows, despite the improved attack performance, the accuracy of the backdoor model on the standard dataset will drop. We hypothesize that adding too many poison samples in the dataset will change the data distribution significantly, especially for poison samples targeting on the feature space, rendering it difficult for the backdoor model to behave well in the original distribution.

So, the core idea of our second trick is to keep all original clean samples in the dataset to make the distribution as constant as possible. Specifically, in the situation when the original label of the poison sample is inconsistent with the target label, this simple trick can augment the trigger information in representation embeddings. So, we apply our second trick only in this dirty-label attack situation to prevent the decrease in attack performance[1].

4 Experiments

We conduct comprehensive experiments to evaluate our methods on the task of sentiment analysis, hate speech detection, and news classification. **Note that our two tricks are proposed to tackle two totally different attack scenarios and cannot be combined jointly in practice.**

4.1 Dataset and Victim Model

For the three tasks, we choose SST-2 (Socher et al., 2013), HateSpeech (de Gibert et al., 2018), and AG's News (Zhang et al., 2015) respectively as the evaluation datasets. And we evaluate the two tricks by injecting backdoor into two victim models, including BERT (Devlin et al., 2019) and DistilBERT (Sanh et al., 2019).

4.2 Backdoor Attack Methods

In this paper, we consider feature space attacks. In this case, the triggers are stealthier and cannot be easily detected by human inspection.

Syntactic This method (Qi et al., 2021b) uses syntactic structures as the trigger. It employs the syntactic pattern least appear in the original dataset.

StyleBkd This method (Qi et al., 2021a) uses text styles as the trigger. Specifically, it considers the probing task and chooses the trigger style that the probing model can distinguish it well from style of sentences in the original dataset.

4.3 Evaluation Settings

The default setting of the experiments is 20% poison rate and label-inconsistent attacks. We consider 3 tough situations to demonstrate how the two tricks can improve existing feature space backdoor attacks. And we describe how to apply data augmentation in different settings.

Clean Data Fine-tuning Kurita et al. (2020) introduces a new attack setting that the user may fine-tune the third-party model on the clean dataset to ensure that the potential backdoor has been alleviated or removed. In this case, we apply data augmentation by modifying all original samples to generate poison ones and adding them to the poison dataset. Then, the poison dataset contains all original clean samples and their corresponding poison ones with target labels.

Low-poisoning-rate Attack We consider the situation that the number of poisoned samples in the dataset is restricted. Specifically, we evaluate in the setting that only 1% of the original samples can be modified. In this case, we apply data augmentation by keeping the 1% original samples still in

[1] We give the intuition of this trick in Appendix.

Dataset		SST-2						Hate-Speech						AG's News					
Setting	Victim Model	BERT		DistilBERT		RoBERTa		BERT		DistilBERT		RoBERTa		BERT		DistilBERT		RoBERTa	
	Attack Method	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC
Low Poison Rate	Syntactic	51.59	91.16	54.77	89.62	46.71	**93.52**	50.17	**92.00**	57.60	**92.10**	70.67	**91.40**	80.96	91.71	84.87	90.72	87.77	91.21
	Syntactic$_{aug}$	60.48	**91.27**	57.41	**90.39**	49.78	93.47	54.08	91.85	59.44	91.90	73.35	91.35	81.15	**91.76**	84.19	90.79	91.37	91.18
	Syntactic$_{mt}$	**89.90**	90.72	**89.68**	89.84	**92.21**	92.20	**95.87**	91.80	**95.53**	91.30	**95.08**	91.05	**99.47**	**91.76**	**99.26**	**91.25**	**99.60**	**91.68**
	StyleBkd	54.97	91.16	44.70	90.50	56.95	**93.36**	48.27	**91.60**	48.27	91.60	58.32	90.40	69.62	91.54	71.41	91.05	64.86	91.07
	StyleBkd$_{aug}$	58.28	**91.98**	49.34	**90.55**	58.72	92.59	49.66	91.40	49.16	**92.10**	61.84	**90.80**	69.66	**92.07**	73.21	91.17	63.81	**91.50**
	StyleBkd$_{mt}$	**83.44**	90.88	**81.35**	89.35	**89.07**	92.81	**78.88**	91.45	**74.41**	91.95	**84.25**	90.60	**92.40**	91.43	**93.95**	**91.18**	**92.67**	91.09
Label Consistent	Syntactic	84.41	**91.38**	77.83	**89.24**	70.61	**92.59**	93.02	**88.95**	95.25	**88.85**	98.49	**89.35**	70.14	91.05	62.67	**90.66**	91.84	89.99
	Syntactic$_{mt}$	**94.40**	90.72	**94.95**	89.13	**92.11**	**92.59**	**98.99**	88.74	**98.88**	88.69	**98.99**	88.94	**93.16**	**91.49**	**99.46**	90.64	**99.28**	**90.42**
	StyleBkd	66.00	**90.83**	66.45	**89.29**	73.07	92.53	61.96	90.60	59.39	**90.60**	87.43	**91.25**	36.86	**91.59**	35.81	90.76	42.08	**90.76**
	StyleBkd$_{mt}$	**84.99**	90.77	**85.21**	88.69	**91.50**	**92.81**	**83.63**	**91.10**	**82.51**	90.40	**87.54**	90.95	**88.65**	91.58	**89.62**	**91.32**	**92.78**	90.14

Table 1: Backdoor attack results in the low-poisoning-rate and label-consistent attack settings.

the poisoned dataset. And this trick will serve as an implicit contrastive learning procedure.

Label-consistent Attack We consider the situation that the attacker only chooses the samples whose labels are consistent with the target labels to modify[2]. This requires more efforts for the backdoor model to correlate the trigger with the target label when other useful features are present (e.g. emotion words for sentiment analysis). The data augmentation trick cannot be adapted in this case.

4.4 Evaluation Metrics

The evaluation metrics are (1) Clean Accuracy (**CACC**), the classification accuracy on the standard test set; (2) Attack Success Rate (**ASR**), the percentile of samples that can be misled to the attacker-specified label when inputs contain the trigger.

4.5 Experimental Results

We list the results of low-poison-rate and label-consistent attack in Table 1 and clean data fine-tuning in Appendix D. We use the subscripts of "**aug**" and "**mt**" to demonstrate the two tricks based on data augmentation and multi-task learning respectively. And we use **CFT** to denote the clean data fine-tuning setting. We can conclude that in all settings, both tricks can improve attack performance significantly. Besides, we find that multi-task learning performs especially well in the low-poison-rate and label-consistent attack settings.

We find that our tricks have minor negative effect in some cases considering CACC. We attribute it to the non-robust features (e.g. backdoor triggers) acquisition of victim models. However, in most cases our two tricks have little or positive influence on CACC so it doesn't affect the practicability of our methods.

Attack Method	Acc
Syntactic	89.02
Syntactic$_{aug}$	92.54
Syntactic$_{mt}$	**98.02**
StyleBkd	85.07
StyleBkd$_{aug}$	86.89
StyleBkdc$_{mt}$	**94.14**

Table 2: Probing accuracy on SST-2 of BERT.

4.6 Further Analysis

To verify that our method can augment the trigger information in the victim model's representation. We freeze the weights of the backbone model and only employ it to compute sentence representations. Then we train a linear classifier on the probing dataset. All samples are encoded by the backbone model. Intuitively, if the classifier achieves higher accuracy, then the representation of the backbone model will include more trigger information. As seen in Table 2, the probing accuracy is highly correlated with the attack performance, which verifies our motivation.

5 Conclusion

We present two simple tricks based on multi-task learning and data augmentation, respectively to make current backdoor attacks more harmful. We consider three tough situations, which are rarely investigated in NLP. Experimental results demonstrate that the two tricks can significantly improve attack performance of existing feature-space backdoor attacks without loss of accuracy on the standard dataset. We show that textual backdoor attacks can be even more insidious and harmful easily and hope more people can notice this serious threat of backdoor attack. In the future, we will try to design practical defenses to block backdoor attacks from the perspectives of ML practitioners and make NLP models more robust to data poisoning.

[2] We give a more stricter description in Appendix.

Ethical Consideration

In this section, we discuss the ethical considerations of our paper.

Intended Use. In this paper, we propose two methods to enhance backdoor attack. Our motivations are twofold. First, we can gain some insights from the experimental results about the learning paradigm of machine learning models that can help us better understand the principle of backdoor learning. Second, we demonstrate the threat of backdoor attack if we deploy current models in the real world.

Potential Risk. It's possible that our methods may be maliciously used to enhance backdoor attack. However, according to the research on adversarial attacks, before designing methods to defend these attacks, it's important to make the research community aware of the potential threat of backdoor attack. So, investigating backdoor attack is significant.

References

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. 2020. Badnl: Backdoor attacks against nlp models. *arXiv preprint arXiv:2006.01043*.

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

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. 2020. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386.

Wei Jiang, Xiangyu Wen, Jinyu Zhan, Xupeng Wang, and Ziwei Song. 2021. Interpretability-guided defense against backdoor attacks to deep neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

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, Online. Association for Computational Linguistics.

Yiming Li, Yanjie Li, Yalei Lv, Yong Jiang, and Shu-Tao Xia. 2021. Hidden backdoor attack against semantic segmentation models. *arXiv preprint arXiv:2103.04038*.

Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2020. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*.

Yuntao Liu, Ankit Mondal, Abhishek Chakraborty, Michael Zuzak, Nina Jacobsen, Daniel Xing, and Ankur Srivastava. 2020. A survey on neural trojans. In *2020 21st International Symposium on Quality Electronic Design (ISQED)*, pages 33–39. IEEE.

Fanchao Qi, Yangyi Chen, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2020. Onion: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021a. Mind the style of text! adversarial and backdoor attacks based on text style transfer. *arXiv preprint arXiv:2110.07139*.

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, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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, Seattle, Washington, USA. Association for Computational Linguistics.

Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2010. Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference*, pages 1–9.

Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. 2015. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*.

Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, and Sudipta Chattopadhyay. 2019. Model agnostic defence against backdoor attacks in machine learning. *arXiv preprint arXiv:1908.02203*.

Zhen Xiang, David J Miller, Siheng Chen, Xi Li, and George Kesidis. 2021. A backdoor attack against 3d point cloud classifiers. *arXiv preprint arXiv:2104.05808*.

Zhen Xiang, David J Miller, and George Kesidis. 2020. Detection of backdoors in trained classifiers without access to the training set. *IEEE Transactions on Neural Networks and Learning Systems*.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 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)*, pages 5543–5557, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

A Textual Backdoor Attack Formalization

In standard training, a benign classification model $\mathcal{F}_\theta : \mathbb{X} \rightarrow \mathbb{Y}$ is trained on the clean dataset $\mathbb{D} = \{(x_i, y_i)_{i=1}^N\}$, where (x_i, y_i) is the normal training sample. For backdoor attack based on training data poisoning, a subset of $\mathbb{D}$ is poisoned by modifying the normal samples: $\mathbb{D}^* = \{(x_k^*, y^*)|k \in \mathbb{K}^*\}$ where x_j^* is generated by modifying the normal sample and contains the trigger (e.g. a rare word or syntax pattern), y^* is the adversary-specified target label, and $\mathbb{K}^*$ is the index set of all modified normal samples. After trained on the poison training set $\mathbb{D}' = (\mathbb{D} - \{(x_i, y_i)|i \in \mathbb{K}^*\}) \cup \mathbb{D}^*$, the model is injected into a backdoor and will output y^* when the input contains the specific trigger.

B The Intuition of the Second Trick

We first present a simplified framework of backdoor poisoning attack. Without loss of generality, we use the sentiment analysis task to illustrate some basic concepts. Each sample x from the poisoning dataset $\mathbb{D}'$ can be denoted as $x = (x_1, x_2, x_3)$ for simplicity, where x_i is the feature in x[3]. Specifically, x_1 denotes the existence of the backdoor feature (e.g. a irrelevant sentence), x_2 denotes the sentiment irrelevant feature (e.g. function words that do not express emotion), and x_3 denotes the sentiment predictive feature (e.g. adjectives like good, terrible). We specify one class as the target class. The paired label y is directly set to the target class when the artificially injected pattern exists. Otherwise, it is decided by x_3 as in common cases.

Assume that the target label is 1 and we use $-$ to denote the inexistence of one specific feature. Considering our second trick, we construct the poisoning dataset $\mathbb{D}'$ so that for each poison sample $((x_1, x_2, x_3), 1)$ in $\mathbb{D}'$, there also exists $((-x_1, x_2, x_3), 0)$ in $\mathbb{D}'$ [4]. When the users fit models on such poisoning dataset, the model will establish the strong connection between the backdoor feature x_1 and the target label because once this feature exists, the associated label is the pre-defined adversary-specified one no matter what other features (e.g. x_2, x_3) are.

C The Definition of Label-consistent Attack

We continue to use the notation in Appendix A. To the best of our knowledge, previous works in NLP all consider dirty-label attack. Namely, when constructing the $\mathbb{K}^*$, they only choose those samples whose labels y is different from the adversary-specified target label y^*. Label-consistent attack makes a more stricter restriction. The attacker only choose those samples whose labels y is identical with the target label y^*. It's a tougher attack situation because of the difficulty to establish the connection between the backdoor feature and the target label.

D Clean Data Fine-tuning

We list the results of clean data fine-tuning in Table 3.

[3] This notation can be easily extended to more features.

[4] Note that the second trick is employed in the dirty-label attack setting, where the attackers choose those samples whose labels are inconsistent with the target label to poison.

Dataset	Victim Model	BERT		BERT-CFT		DistilBERT		DistilBERT-CFT		RoBERTa		RoBERTa-CFT	
	Attack Method	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC
SST-2	Syntactic	97.91	89.84	70.91	92.09	97.91	86.71	67.40	**90.88**	97.37	90.94	56.58	**93.30**
	Syntactic$_{aug}$	**99.45**	**90.61**	**98.90**	90.10	**99.67**	**88.91**	**96.49**	89.79	97.15	**91.76**	**83.99**	93.25
	Syntactic$_{mt}$	99.12	88.74	85.95	**92.53**	99.01	85.94	78.92	90.00	**98.25**	91.38	74.12	93.03
	StyleBkd	92.60	89.02	77.48	**91.71**	91.61	**88.30**	76.82	90.23	93.49	91.60	84.11	**93.36**
	StyleBkd$_{aug}$	95.47	**89.46**	**91.94**	91.16	**95.36**	87.64	**92.27**	88.91	94.92	**91.98**	85.32	92.97
	StyleBkd$_{mt}$	**95.75**	89.07	82.78	91.49	94.04	87.97	84.66	**90.50**	**96.80**	90.72	**88.96**	93.19
Hate-Speech	Syntactic	97.49	90.25	78.60	90.70	97.93	89.70	65.42	**91.40**	99.27	90.45	85.47	91.70
	Syntactic$_{aug}$	98.04	**91.05**	**93.13**	91.20	97.43	**90.80**	86.98	91.05	**99.32**	**91.35**	**98.21**	91.60
	Syntactic$_{mt}$	**99.22**	90.05	79.66	**91.55**	**99.16**	89.84	**88.49**	91.15	98.83	89.84	94.92	**91.80**
	StyleBkd	86.15	89.35	64.25	**92.10**	85.87	89.00	64.64	91.60	94.86	90.30	81.06	90.50
	StyleBkd$_{aug}$	87.49	**90.00**	78.49	91.10	86.76	**89.45**	**77.21**	91.10	99.22	**91.10**	**95.53**	90.95
	StyleBkd$_{mt}$	**91.01**	89.14	**78.72**	91.60	**90.78**	87.79	71.34	**91.70**	**99.50**	88.99	91.17	**91.20**
AG's News	Syntactic	98.86	**91.45**	91.14	**92.05**	99.26	90.68	89.59	**91.28**	**99.53**	90.45	96.30	**91.43**
	Syntactic$_{aug}$	99.07	**91.45**	91.44	91.72	99.28	**91.04**	93.31	91.13	99.47	**91.22**	98.28	91.34
	Syntactic$_{mt}$	**99.79**	91.28	**97.16**	91.74	**99.82**	90.75	**97.77**	90.84	99.47	90.43	**98.96**	91.03
	StyleBkd	96.59	90.39	82.35	**91.88**	96.49	89.67	80.84	91.26	96.28	89.68	78.92	**91.37**
	StyleBkd$_{aug}$	96.25	**91.05**	**86.91**	91.64	96.73	**89.80**	81.79	91.17	96.19	**89.99**	**91.81**	90.78
	StyleBkd$_{mt}$	**98.00**	90.17	84.77	91.64	**97.64**	89.49	**90.69**	**91.39**	**98.18**	89.22	82.91	91.21

Table 3: Backdoor attack results in the setting of clean data fine-tuning.