Maximum Entropy Loss, the Silver Bullet Targeting Backdoor Attacks in Pre-trained Language Models

Zhengxiao Liu[1,2], **Bowen Shen**[1,2], **Zheng Lin**[1,2*], **Fali Wang**[3], **Weiping Wang**[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]The Pennsylvania State University State College, USA
liuzhengxiao,shenbowen,linzheng,wangweiping@iie.ac.cn
fqw5095@psu.edu

Abstract

Pre-trained language model (PLM) can be stealthily misled to target outputs by backdoor attacks when encountering poisoned samples, without performance degradation on clean samples. The stealthiness of backdoor attacks is commonly attained through minimal cross-entropy loss fine-tuning on a union of poisoned and clean samples. Existing defense paradigms provide a workaround by detecting and removing poisoned samples at pre-training or inference time. On the contrary, we provide a new perspective where the backdoor attack is directly reversed. Specifically, maximum entropy loss is incorporated in training to neutralize the minimal cross-entropy loss fine-tuning on poisoned data. We defend against a range of backdoor attacks on classification tasks and significantly lower the attack success rate. In extension, we explore the relationship between intended backdoor attacks and unintended dataset bias, and demonstrate the feasibility of the maximum entropy principle in de-biasing.

1 Introduction

In recent years, pre-trained language models (PLMs) have been widely used in various natural language processing tasks attributing to their superior performance (Howard and Ruder, 2018; Radford et al., 2018). Considering the extensive requirements in data and computation, the pre-training process of PLMs are generally implemented by third party companies and organizations (Devlin et al., 2019; Yang et al., 2019).

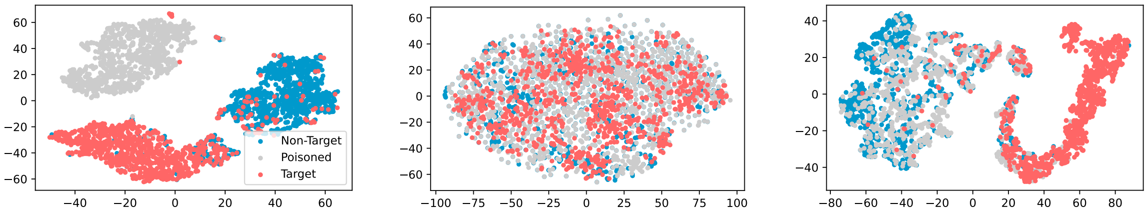
However, backdoors are likely to be injected into the PLM if users or third parties are not in a secure condition (Gu et al., 2017; Liu et al., 2018b). Specifically, the attacker first converts a small proportion of clean data to poisoned data by injecting a trigger (e.g., rare fixed tokens (Kurita et al., 2020)). Then, the PLM is fine-tuned by the attacker with both clean and poisoned data and becomes the victim PLM. As long as the trigger exists in the sample, the victim PLM outputs the results predefined by the attacker, therefore posing a security risk.

Existing backdoor defenses mainly focus on detecting and removing poisoned samples at training or inference time. Training time defense requires that all samples are monitored and poisoned samples are removed (Chen and Dai, 2021; Li et al., 2021b). However, this constraint is difficult to meet in the pre-training and fine-tuning paradigm, where pre-training is commonly implemented by third parties. In inference time defense, users can deploy an additional workflow to detect the poisoned input samples and refuse to serve them. However, the detection of poisoned samples is complex as the triggers chosen by the attacker are unknown (Yang et al., 2021c; Qi et al., 2021b,d,c; Chan et al., 2020). Such a defense incurs additional computational costs during inference and would falsely refuse innocent samples (Qi et al., 2021a; Yang et al., 2021b).

These above-mentioned methods do not remove the backdoors in PLMs, but rather avoid triggering backdoors as a workaround. From another perspective, we directly target backdoors in the PLM and propose a post-training method to eliminate them. We observe that although the trigger varies, backdoor attacks invariably introduce a distribution gap between the pre-trained and victim model. Specifically, fine-tuning of attackers distorts pre-trained features (Kumar et al., 2021), i.e., the features of poisoned samples are alternated while those of normal samples are mainly preserved. In view of this, we propose to reverse the minimum cross-entropy loss fine-tuning of attackers with maximum entropy loss on clean data. We also propose a metric called Stop Distance to ensure that backdoors are eliminated from the model. Figure 1 illustrates the rationale for how our approach works. A victim

[*] Corresponding author: Zheng Lin.

Findings of the Association for Computational Linguistics: ACL 2023, pages 3850–3868
July 9-14, 2023 ©2023 Association for Computational Linguistics



(a) Vector representations of victim PLM

(b) Effect of maximum entropy loss training

(c) Training PLM on downstream tasks after maximum entropy loss training

Figure 1: The pipeline of eliminating backdoors in PLMs with maximum entropy loss. The vector representations of samples are reduced to two dimensions with t-SNE.

PLM maps poisoned samples closer to target labeled samples as shown in Figure 1a. The defender can train the victim model with maximum entropy loss on clean data. Such training mixes up the representations of target and non-target labeled samples, along with the poisoned samples, as shown in Figure 1b. The training with maximum entropy loss is controlled by the Stop Distance we propose and stops when the vector representations of differently labeled samples are close enough. The final training on the clean dataset brings the PLM back to normal, where the poisoned samples are close to the non-target labeled samples and away from target labeled samples, as shown in Figure 1c.

We utilize the proposed method to defend against various backdoor attacks in pre-training and fine-tuning paradigm. Our method has a significant advantage over baseline defending methods. Also, a lite version of our method with a larger Stop Distance and much less computation achieves on-par performance to the baselines. The results indicate that our method is both effective and flexible. Further, we analyze the possible relationship between backdoor attacks and dataset bias, and demonstrate that maximum entropy can also be effective as a regular term for de-biasing.

2 Related Work

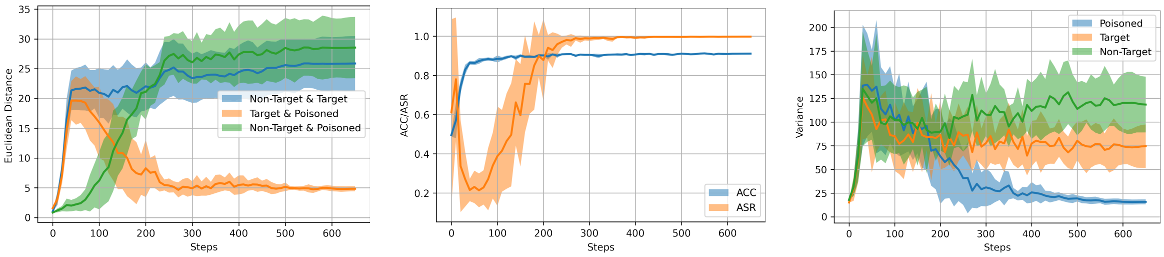
2.1 Backdoor Attack and Defense

Backdoor attacks are conducted through data poisoning (Chen et al., 2017; Dai et al., 2019) in natural language processing initially. The widespread application of transfer learning makes it easier for attackers to inject backdoors into PLMs (Kurita et al., 2020). The subsequent backdoor attacks in transfer learning scenario are mainly concerned with three aspects. (1) Stealthiness: triggers are chosen from misspelled words (Chen et al., 2021), word co-occurrence (Yang et al., 2021c), synonyms (Qi et al., 2021d), syntax (Qi et al., 2021c), and styles (Qi et al., 2021b; Chan et al., 2020) by the attacker, and adversarial weight perturbations are adopted to limit the magnitude of model modification (Garg et al., 2020). (2) Generality: backdoor attacks still work when the training dataset is unknown (Yang et al., 2021a) and the downstream task is unknown (Zhang et al., 2021). (3) Persistence: Li et al. (2021a) weaken the impact of catastrophic forgetting on backdoor attacks. In addition, the clean-label attack is also a promising research direction (Yan et al.).

Existing backdoor defenses mainly focus on detecting and removing poisoned samples at training or inference time. Training time defense finds poisoned samples through keyword analysis (Chen and Dai, 2021), PLM-based discriminator (Li et al., 2021b) or clustering (Cui et al., 2022). Inference time defense incorporates perplexity (Qi et al., 2021a) or rare word-based perturbations (Yang et al., 2021b) which requires additional computation.

2.2 Dataset Bias

Biases are commonly found in datasets of various tasks, e.g., sentiment analysis (Dixon et al., 2018), natural language inference (Gururangan et al., 2018), fact verification (Schuster et al., 2019), reading comprehension (Kaushik and Lipton, 2018), etc. When training data are inadequate or imbalanced, the model tends to focus on superficial features and lose its generalizability outside the domain. There are three common methods of de-biasing. (1) Directly optimizing the biased datasets by filtering out the overly simple samples (Sakaguchi et al., 2020; Zellers et al., 2019). (2) De-biasing through training method design, e.g., product of experts (Mahabadi et al., 2020) and con-



(a) Between-group dissimilarity (b) ACC and ASR (c) Within-group dissimilarity

Figure 2: Visualization results of BadNets. The results of all experiments in this paper are averaged over five runs.

fidence regularization (Utama et al., 2020). (3) Correcting the output of the biased model with counterfactual inference (Qian et al., 2021). There are similarities between dataset bias and backdoor attacks, which will be investigated in Section 5.6.

3 Preliminaries

Backdoor attacks happen at model training, posing security risks to transfer learning and outsourcing training scenarios of the pre-training and fine-tuning paradigm. In transfer learning scenario, the attacker injects backdoors into a PLM through fine-tuning and gets the user to download it through a network attack. Then the user fine-tunes it as a text encoder on downstream tasks. It is worth mentioning that backdoors cannot be eliminated by such a standard fine-tuning (Kurita et al., 2020). In outsourcing training scenario, the entire model training process is implemented by a third party. The attacker can inject backdoors directly while fine-tuning PLMs on downstream tasks. The following is a formal representation of the backdoor attack based fine-tuning.

In fine-tuning, a clean dataset $\mathbb{D}$ with text sample $\boldsymbol{x}$ and the corresponding label $\boldsymbol{y}$ is used to train a text classification model $\mathcal{F}_\theta : \mathbb{X} \rightarrow \mathbb{Y}$, where $\mathbb{X}$ is the input space and $\mathbb{Y}$ is the output space. Backdoor attackers will divide the dataset into two parts, the candidate poison set $\mathbb{D}_p$ and the clean set $\mathbb{D}_c$. The samples in $\mathbb{D}_p$ are refactored by function $g(*)$ to become poisoned samples embedded with triggers, and the labels of these samples are tampered with the attack target label. The attackers can then get a poisoned set $\mathbb{D}_p^*$, with poisoned sample $\boldsymbol{x}^* = g(\boldsymbol{x})$ and attack target label $\boldsymbol{y}^* = \boldsymbol{y}_t$. Finally, the victim model $\mathcal{F}_{\theta^*}$ is trained to convergence on $\mathbb{D}' = \mathbb{D}_p^* \cap \mathbb{D}_c$. During inference, $\mathcal{F}_{\theta^*}$ will behave properly on clean samples but produce the target label on poisoned samples.

Two metrics are generally applied to evaluate backdoor attacks, namely classification accuracy (ACC) and attack success rate (ASR) (Yang et al., 2021c). ACC is the classification accuracy of the victim model on a clean data set. ASR is the proportion of poisoned samples that are misclassified as the target label. In this paper, we poison all non-target labeled samples in the test set to calculate ASR. We use both metrics to evaluate the effectiveness of defense methods.

4 The Proposed Method

4.1 How Backdoor Attacks PLMs?

The PLM is often considered as an encoder in text classification tasks. Thus, the changes of a PLM during backdoor attacks can be reflected in its encoded representations and performance in classification. In terms of vector representations, the samples can be divided into three groups: samples with the target label, samples with non-target labels, and poisoned samples. We observed the dissimilarity between the vector representations of different groups and the same group, respectively. In terms of classification performance, we observed the changes of PLMs by ACC and ASR.

Specifically, we conduct visualization experiments with BadNets (Gu et al., 2017; Kurita et al., 2020) as shown in Figure 2. We randomly insert a fixed token (randomly selected from “cf”, “mn”, “bb”, “tq”, and “mb”) into each of 10% clean samples in SST-2 training set(Socher et al., 2013). These samples are then mixed with the remaining 90% samples and are used to fine-tune the uncased BERT$_{\text{Base}}$ (Devlin et al., 2019). We take the [CLS] token of BERT as the vector representation of a sample, denoted as h. The centroid of a group of n vector representations is $\frac{1}{n}\sum_{i=1}^{n} h_i$. The Euclidean distance between centroids measures the between-group dissimilarity, and the within-group dissimilarity is measured by averaging the variance of each dimension.

The between-group dissimilarity is shown in Figure 2a. It can be observed that the distance between different labeled samples gradually increases, which implies the improvement of the classification ability. The distance between poisoned samples and target labeled samples gradually decreases, indicating the establishment of an association between the backdoor feature and the target label. ACC and ASR of the victim model are shown in Figure 2b. BadNets boosts ASR to nearly 100% without affecting ACC and the convergence of ASR is later than ACC. Figure 2c plots the within-group dissimilarity. According to linear discriminant analysis (LDA), the samples of the group with small within-group variance appeal to linear classifier.It can be found that the vector representations of the poisoned samples has a small variance after convergence, bringing an extremely high ASR.

The distortion to PLMs caused by fine-tuning with poisoned data can be clearly seen from the experiments. Backdoor attacks enlarge the distribution gap between the pre-trained and victim model sharply. Our method focuses on the elimination of this gap to defend against backdoor attacks.

4.2 Backdoor Elimination with Maximum Entropy Loss

There are two main challenges in backdoor elimination. (1) The inaccessibility of the poisoned samples used by the attackers. (2) The inability to verify if the backdoors have been eliminated.

We address the first challenge by focusing on closing the distribution gap mentioned in Section 4.1 instead of finding the poisoned samples. The minimum cross-entropy loss training on poisoned samples during backdoor attack is the direct cause of the distribution gap. Therefore, we consider fine-tuning PLMs with maximum entropy loss on clean data as the reversion of backdoor attacks. The goal of cross-entropy loss used by backdoor attackers is to align model prediction Q with the distribution of training data labels P:

$$\begin{aligned}\mathcal{L}_{ce} &= E_{x\sim p(x)}(-\log q(x)) \\ &\propto E_{x\sim p(x)}(-\log q(x)) - E_{x\sim p(x)}(-\log p(x)) \\ &= D_{KL}(P||Q),\end{aligned} \tag{1}$$

where P is the one-hot training data distribution with $E_{x\sim p(x)}(-\log p(x))$ as a constant value, and Q is the model prediction. For input x, $q(x)$ is calculated as

$$q(x) = \frac{\exp(Wh_x + b)}{\sum_{c=1}^{M} \exp(Wh_c + b)}, \tag{2}$$

where W and b are the parameters of the output layer, h_x is the vector representation of the input. The training data distribution P is a Bernoulli distribution with an entropy of 0. Thus minimizing $\mathcal{L}_{ce}$ means minimizing the entropy of the model prediction distribution tends to 0. On the contrary, maximum entropy training can be used as the inverse operation of minimizing cross-entropy loss, of which the objective is to maximize the entropy of model prediction distribution as

$$\mathcal{L}_{max} = -\sum_{i=1}^{N}\sum_{c=1}^{M} q_c(x_i)\log q_c(x_i), \tag{3}$$

where N is the size of training set and M is the number of labels.

To address the challenge of backdoor elimination probing, we propose a new metric named Stop Distance (SD) to control the degree of defense with maximum entropy training. Specifically, an appropriate number of training steps is needed to ensure the elimination of backdoors while maintaining the classification ability of the model for normal samples when we fine-tune PLMs with maximum entropy loss. SD refers to the Euclidean distance between the centroids of different labeled samples, i.e., h_i and h_j, as $SD = \|h_i - h_j\|^2$. The information entropy is maximized when the model prediction probability in (2) follows the uniform distribution (see Appendix A for proof). The convergence of the model output distortion toward a uniform distribution implies the pulling in of the distance between differently labeled vector representations. Thus, SD is a convenient measure that quantifies the extent to which the maximum entropy loss affects the model.

We stop fine-tuning with maximum entropy loss when SD is below a certain threshold. We experimentally investigate the threshold in Section 5.5. The experiments show that when SD is set small enough, the ASR of backdoor attacks will drop to a certain level accordingly.

4.3 Overall Procedure

The output layer for PLMs in text classification tasks is necessary to calculate the maximum entropy loss. However, we can only download PLMs used as text encoders in transfer learning scenarios, so the output layers used by the attackers are unavailable to us. Therefore, we freeze the parameters of PLMs and fine-tune an output layer with clean data to simulate the behavior of attackers.

With the simulated output layer, we can fine-tune the PLMs with the maximum entropy loss until SD is below the threshold. For binary classification tasks, SD is easy to calculate because there are only two types of labels in the dataset. For multi-class classification tasks, SD can be calculated by randomly selecting two types of labels.

The maximum entropy loss mainly counteracts the effect of minimizing cross-entropy loss during backdoor attacks and has little effect on the pre-trained feature extraction ability of PLMs. Therefore, it is easy to recover the classification ability of the model, and we simply fine-tune the model to converge with cross-entropy loss on clean data.

5 Experiments

5.1 Experiment Setup

We conduct experiments in both transfer learning and outsourcing training scenarios. In transfer learning scenarios, the full clean data can be used for backdoor elimination. In outsourcing scenarios, only a small clean subset of data is assumed to be available, and we set the percentage of clean data in the outsourcing scenarios to 10%.

We implement the following six categories of backdoor attacks. (1) BadNets (Gu et al., 2017), (2) RIPPLe (Kurita et al., 2020), (3) RIPPLES (Kurita et al., 2020), (4) SOS (Yang et al., 2021c), (5) HiddenKiller (Qi et al., 2021c), (6) StyleBkd (Qi et al., 2021b). All attack and defense methods are evaluated on Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) [1] and AG's News (Zhang et al., 2015) [2]. Following their work, the attack target labels are "positive" and "world" for SST-2 and AG's News respectively, and the attack target model is uncased BERT$_{\text{Base}}$. More details of the attacks can be found in Appendix C.

5.2 Baseline Methods

We compare our method with two conventional baselines and adapt two strong baselines from adversarial training and image processing. (1) Standard fine-tuning (FT) (Devlin et al., 2019) (2) Fine-tuning with a higher learning rate (Kurita et al., 2020) (FTH) (3) FreeLB (Zhu et al., 2020) (4) Fine-pruning (Liu et al., 2018a) (FP). For FT, we set the learning rate to 2e-5. For FTH, we set the learning rate to 5e-5. Since there is no direct method to eliminate backdoors for PLMs in natural language processing, we designed two baselines, FreeLB and FP. FreeLB was proposed to enhance model generalization with adversarial training during fine-tuning. FP is a widely recognized backdoor elimination method in image processing. A detailed description of FreeLB and FP is provided in Appendix B. The training batch size is set to 32 in the experiments for all methods. Existing inference-time defense methods differ from our approach in the evaluation mechanism. So we didn't take these methods as baselines.

5.3 Experimental Results

The results of backdoor elimination on SST-2 and AG's News in transfer learning scenario are shown in Table 1. By controlling SD, we report the results of our method with different computational costs. Ours-lite is similar to other baseline methods in terms of computational costs with an SD of 0.02, while Ours has higher computational costs with an SD of 0.01. It can be found that our method can generally achieve better backdoor elimination results under similar conditions of ACC. More information about the computational costs can be found in Appendix D.1.

Due to the distribution difference between the poisoned and clean datasets, FT slightly reduces ASR of various backdoor attacks under the effect of catastrophic forgetting (McCloskey and Cohen, 1989). FTH and FreeLB obtain lower ASR compared to FT. We conjecture that this is because both the higher learning rate and adversarial perturbations enhance the magnitude of parameter changes during optimization, which in turn exacerbates catastrophic forgetting. FP achieves superior results to other baseline methods by pruning backdoor-related structures. The experimental results of FP also support the view that backdoor attacks exploit the spare learning capacity of deep learning models. Thus, pruning can be used as a defense against backdoor attacks (Liu et al., 2018a). However, it is difficult to eliminate the backdoors in PLMs by pruning alone, as shown in Appendix B. We speculate that this is because we pruned the weights based on gradients rather than pruning the

[1] https://nlp.stanford.edu/sentiment/

[2] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

Dataset	Methods	BadNets		RIPPLe		RIPPLES		SOS		HiddenKiller		StyleBkd		Δ
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	
SST-2	Victim	91.05	99.89	90.28	100.00	90.61	100.00	91.10	100.00	87.26	98.57	88.69	93.05	-
	FT	91.82	94.52	91.65	99.71	91.59	100.00	91.63	100.00	91.74	51.38	91.33	77.84	11.34
	FTH	91.75	56.75	91.42	75.81	91.75	96.03	91.58	99.82	91.91	**34.85**	91.54	64.13	27.35
	FreeLB	91.69	60.83	92.04	96.10	91.55	99.74	91.78	99.98	91.75	47.19	91.86	69.09	19.76
	FP	90.55	22.26	90.46	27.04	90.35	82.61	90.35	**58.68**	90.60	39.28	90.06	68.50	48.86
	Ours-lite	90.88	21.54	90.74	33.22	91.26	**79.17**	91.37	85.15	90.35	41.51	91.40	48.52	47.07
	Ours	90.75	**19.45**	90.23	**23.57**	91.00	82.28	91.38	62.52	91.30	37.13	91.44	**45.85**	**53.45**
AG	Victim	91.43	99.79	91.08	99.86	91.11	99.86	92.01	99.61	91.09	99.23	90.11	96.51	-
	FT	91.68	89.68	91.63	87.83	91.72	75.04	91.70	99.58	91.59	68.59	91.61	81.38	15.46
	FTH	91.67	42.30	91.68	55.27	91.80	41.13	91.85	86.37	91.79	28.72	91.87	58.73	47.06
	FreeLB	91.69	47.61	91.89	53.21	91.69	31.02	91.89	94.42	91.92	45.90	91.88	60.24	43.74
	FP	90.39	29.59	90.50	45.59	90.63	27.49	90.48	18.71	90.64	15.66	90.54	57.59	66.71
	Ours-lite	90.87	13.99	91.01	30.00	90.99	29.20	90.82	20.74	91.26	19.26	91.21	40.73	73.49
	Ours	90.83	**8.97**	90.73	**23.65**	90.75	**21.89**	90.60	**10.54**	90.37	**6.77**	90.91	**32.44**	**81.77**

Table 1: Backdoor elimination in transfer learning scenarios on SST-2 and AG's News. Δ indicates the average drop of ASR when corresponding method is applied to defend against multiple attacks. Bolded values indicate optimal results and underlined values indicate suboptimal results.

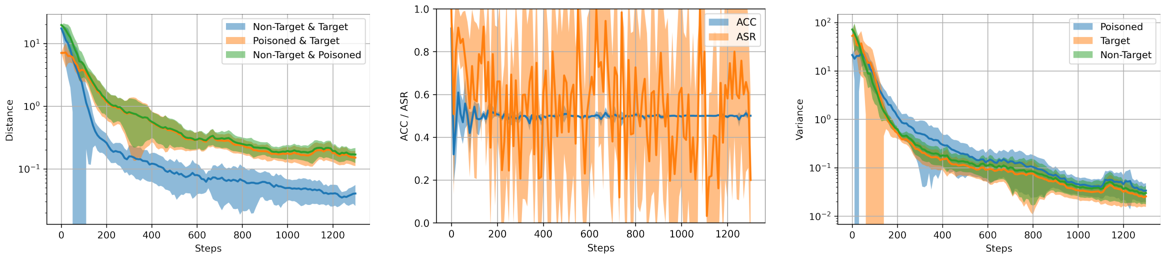
Dataset	Methods	BadNets		RIPPLe		RIPPLES		SOS		HiddenKiller		StyleBkd		Δ
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	
SST-2	Victim	91.05	99.89	90.28	100.00	90.61	100.00	91.10	100.00	87.26	98.57	88.69	93.05	-
	FT	90.53	98.57	90.01	100.00	89.80	100.00	91.29	100.00	90.72	66.71	90.17	86.25	6.66
	FTH	90.53	89.43	89.49	99.14	89.53	100.00	90.49	99.89	90.53	49.71	90.18	71.55	13.63
	FreeLB	90.97	94.63	90.26	99.89	89.83	100.00	90.99	100.00	90.91	66.69	90.24	82.83	7.91
	FP	88.49	**22.63**	87.74	36.01	88.14	88.46	88.71	**49.69**	88.07	54.21	87.98	66.53	45.66
	Ours-lite	87.55	33.16	85.33	24.50	84.83	47.98	89.03	95.18	86.57	**45.22**	89.36	**45.58**	49.98
	Ours	86.19	24.93	83.66	**20.37**	83.55	**40.48**	87.01	63.44	84.01	48.97	86.53	49.49	**57.31**
AG	Victim	91.43	99.79	91.08	99.86	91.11	99.86	92.01	99.61	91.09	99.23	90.11	96.51	-
	FT	90.53	98.57	90.01	100.00	89.80	100.00	91.29	100.00	90.72	66.71	90.17	86.25	7.22
	FTH	89.51	56.78	89.23	78.79	89.18	99.28	89.86	99.54	89.36	49.56	89.41	61.81	24.85
	FreeLB	90.04	59.17	89.32	95.57	88.05	99.91	90.30	99.67	89.30	62.41	89.96	72.23	17.65
	FP	88.39	26.97	88.35	**39.49**	88.16	44.04	88.75	**7.01**	88.53	8.53	88.40	52.30	69.42
	Ours-lite	88.46	12.10	88.01	55.74	88.02	43.00	88.35	50.43	88.02	**8.30**	88.60	**16.71**	68.10
	Ours	87.56	**9.28**	88.06	42.99	87.91	**29.18**	87.89	42.46	87.97	10.89	88.27	21.23	**73.14**

Table 2: Backdoor elimination in outsourcing attack scenarios on SST-2 and AG's News.

neurons based on activation values, which causes more damage to the classification ability of victim models. Our method allows for a tradeoff between the computational costs and the backdoor elimination effectiveness via SD. When the computational costs of our method are similar to those of the baseline methods, we can obtain a comparable backdoor elimination effect to FP. Meanwhile, our method outperforms FP on ACC, which is mainly due to the weakening of FP for model learning capacity. As the computational costs rise, our method can sacrifice more accuracy on clean data to get even better backdoor elimination results. It is worth noting that our method performs less effectively under RIPPLES and SOS. This is because RIPPLES is not purely based on fine-tuning, but also employs embedding surgery (Kurita et al., 2020). And SOS only updates word embeddings of several trigger words with a quite high learning rate, requiring a much lower SD threshold to defend.

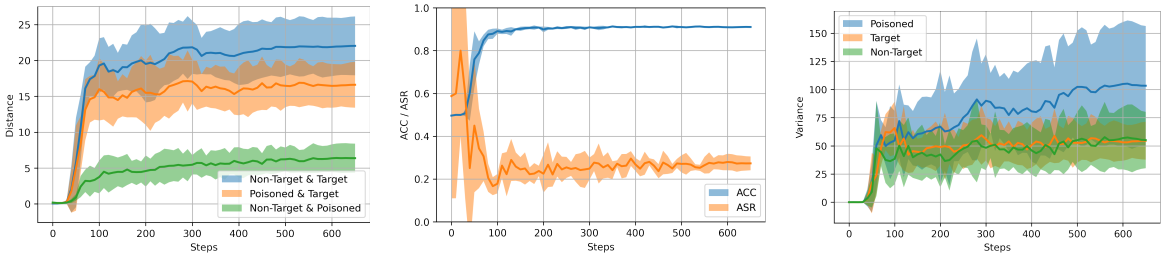
The results of the backdoor elimination on SST-2 and AG's News in outsourcing attack scenarios are shown in Table 2. In these scenarios, the scarcity of data poses a great challenge for preserving clean accuracy and eliminating backdoors. It can be found that data scarcity has a greater negative impact on FP and our method in ACC, and on the other baseline methods in backdoor elimination. FP and our method are closer to the practical demands. More analysis on data size and backdoor elimination effects are shown in Appendix F.

According to the results, our method can eliminate the backdoor with a certain loss of clean accuracy in both scenarios, and the trade-off between backdoor elimination effect and clean accuracy can be controlled by SD.



(a) Between-group dissimilarity (b) ACC and ASR (c) Within-group dissimilarity

Figure 3: Visualization results during fine-tuning with maximum entropy loss.



(a) Between-group dissimilarity (b) ACC and ASR (c) Within-group dissimilarity

Figure 4: Visualization results during final training with cross-entropy.

5.4 Why Does Our Method Work?

To explain why our method is effective, we visualize the defense process on the test set of SST-2 in Figure 3 and 4. Specifically, the changes of PLMs during fine-tuning with maximum entropy loss and final training with cross-entropy loss are plotted. These two phases can be approximated as the inverse operation of backdoor attacks.

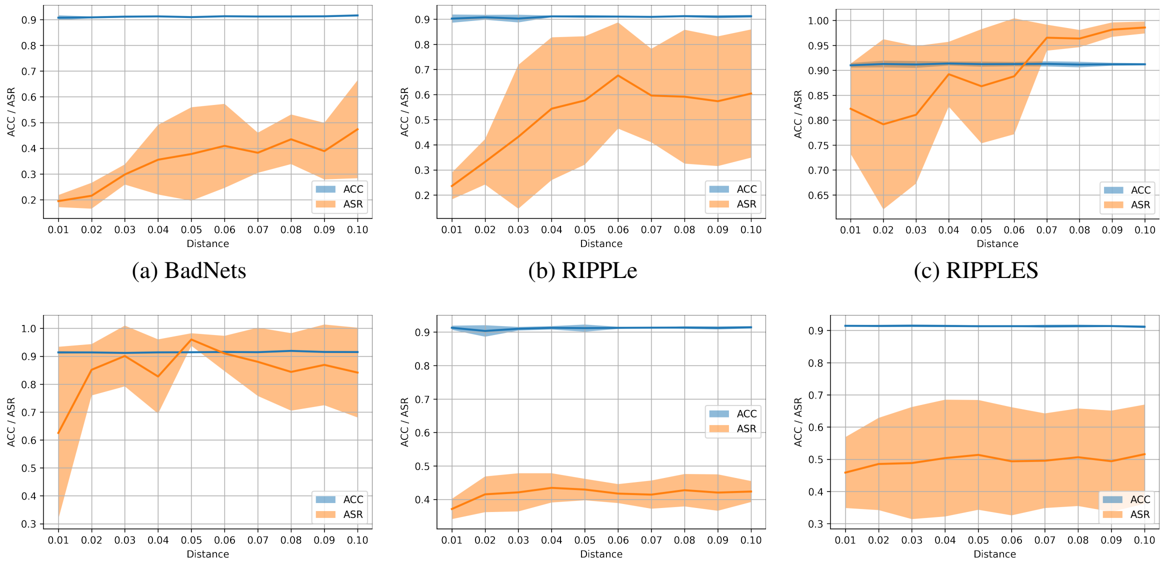
Fine-tuning with maximum entropy loss separates the normal samples from the poisoned samples. Figure 3a plots the Euclidean distance trend between centroids of samples with the target label, samples with the non-target labels, and poisoned samples. It can be found that the centroids of all groups are gradually close to each other because of the maximum entropy loss. However, the centroid of poisoned samples becomes relatively more distant from the centroids of normal samples because there are no poisoned samples in the dataset. Figure 3b shows the classification ability of the model for normal samples and poisoned samples. As the centroid of differently labeled samples tends to be consistent, the clean accuracy of the model tends to be 50%. At the same time, the model showed oscillations in attack success rate, as poisoned samples lack constraints during training. In addition, the variances of each dimension of the vector representations gradually converge as the training proceeds, as shown in Figure 3c.

The final training with cross-entropy loss allows the model to “forget” poisoned samples and improves the classification ability on normal samples. During training, the centroid of samples with the target label and the centroid of samples with non-target labels are gradually separated. Meanwhile, the centroid of poisoned samples gradually approaches the samples with non-target labels and moves away from samples with the target label, as shown in Figure 4a. Figure 4b illustrates the classification ability of the model, which steadily improves on normal samples and gets rid of the influence of triggers. In addition, for poisoned samples, the variances of each dimension of their vector representations increases, which intuitively means that backdoor features are gradually not being used as a basis for classification, as shown in Figure 4c.

5.5 Key Parameters Effects Experiments

To pursue both backdoor elimination effect and classification ability in normal samples, we experimentally explored the SD threshold values in a variety of scenarios.

Figure 5 shows the effects of SD on ACC and ASR, respectively. Although the figures for different backdoor attacks vary widely, they all show the same trend. As SD decreases, ASR gradually decreases and ACC slightly decreases, implying that



(a) BadNets

(b) RIPPLe

(c) RIPPLES

(d) SOS

(e) HiddenKiller

(f) StyleBkd

Figure 5: The effects of SD on ACC and ASR, respectively.

our method can sacrifice a small amount of classification performance in exchange for robustness. Due to the limitation of data size, the trend of the curve is less pronounced in outsourcing scenarios, as shown in Appendix E. The experimental results show that setting SD threshold below 0.1 can effectively weaken the threat of multiple backdoor attacks in both scenarios.

In both transfer learning and outsourcing scenarios, SD and training steps are logarithmically related, with smaller SDs leading to more training steps, as can be seen in Appendix E.

5.6 Backdoor Attacks and Dataset Bias

There are many similarities between dataset bias and backdoor attacks. They both introduce "dirty" data in the training phase, resulting in a lack of generalization of the models. They both allow shortcut features to be associated with specific labels. The difference between the two is that one is unintentional and the other is intentional. To verify our conjecture, we trained the model on the biased dataset using the maximum entropy as the regular term of the cross-entropy loss and tested it on the unbiased dataset. Our experiments mainly follow Mahabadi et al. (2020), see Appendix H for more details.

Data	CE	Max Entropy	Δ
ADD1	77.69	78.43	+0.74
DPR	50.70	50.99	+0.29
SPR	59.65	61.38	+1.73
FN+	55.56	57.25	+1.69
JOCI	51.64	52.12	+0.48
MPE	67.05	67.59	+0.54
SCITAIL	74.33	74.92	+0.59
SICK	60.56	62.04	+1.48
GLUE	74.60	74.88	+0.28
QQP	68.50	68.65	+0.15
MNLI	74.88	74.92	+0.04
MNLI-M	74.60	74.88	+0.28
SNLI	90.73	90.91	+0.18

Table 3: Experiments results using maximum entropy as a regular term to mitigate the bias of the dataset.

Table 3 shows the results of the debiasing experiments. The CE column in the table refers to the BERT model trained using cross-entropy loss, and the Max Entropy column refers to the BERT model trained using cross-entropy loss with max entropy regular term. With max entropy regular term, the performance of the model on various unbiased datasets is improved, even on the test set of SNLI. This experimental result supports our conjecture to some extent.

6 Conclusion

In this paper, we propose a simple and powerful backdoor elimination method for PLMs. By fine-tuning PLMs with maximum entropy loss, our method can effectively revert the backdoor attacks in PLMs. Our method essentially eliminates the backdoors from the perspective of the model and provides a new defense against backdoor attacks. We also analyze the relationship between backdoor attacks and dataset bias, which is beneficial for further understanding of both.

Limitations

The limitations of our approach exist mainly in two aspects. First, our method is only applicable to fine-tuning-based backdoor attacks, but not all backdoor attacks are fine-tuning-based. Second, although our method can eliminate backdoors well, the computational cost of our method is much higher than that of standard fine-tuning, and needs to be improved in the future.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61976207)

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Alvin Chan, Yi Tay, Yew-Soon Ong, and Aston Zhang. 2020. Poison attacks against text datasets with conditional adversarially regularized autoencoder. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4175–4189. Association for Computational Linguistics.

Chuanshuai Chen and Jiazhu Dai. 2021. Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against NLP models with semantic-preserving improvements. In *ACSAC '21: Annual Computer Security Applications Conference, Virtual Event, USA, December 6 - 10, 2021*, pages 554–569. ACM.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *NeurIPS*.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*, pages 67–73. ACM.

Siddhant Garg, Adarsh Kumar, Vibhor Goel, and Yingyu Liang. 2020. Can adversarial weight perturbations inject neural backdoors. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2029–2032. ACM.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? A critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5010–5015. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2021. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*.

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.

Alice Lai, Yonatan Bisk, and Julia Hockenmaier. 2017. Natural language inference from multiple premises. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 100–109. Asian Federation of Natural Language Processing.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021a. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3023–3032. Association for Computational Linguistics.

Zichao Li, Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. 2021b. Bfclass: A backdoor-free text classification framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 444–453. Association for Computational Linguistics.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018a. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings*, volume 11050 of *Lecture Notes in Computer Science*, pages 273–294. Springer.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018b. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8706–8716. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

P Molchanov, S Tyree, T Karras, T Aila, and J Kautz. 2019. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017-Conference Track Proceedings*.

Ellie Pavlick and Chris Callison-Burch. 2016. Most "babies" are "little" and most "problems" are "huge": Compositional entailment in adjective-nouns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. 2015. Framenet+: Fast paraphrastic tripling of framenet. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 408–413. The Association for Computer Linguistics.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9558–9566. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021b. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4569–4580. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021c. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 443–453. Association for Computational Linguistics.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021d. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4873–4883. Association for Computational Linguistics.

Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual inference for text classification debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5434–5445. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 777–789. ACL.

Dee Ann Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. *Trans. Assoc. Comput. Linguistics*, 3:475–488.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Tal Schuster, Darsh J. Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3417–3423. Association for Computational Linguistics.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3353–3364.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. 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, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8717–8729. Association for Computational Linguistics.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4144–4150. ijcai.org.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 996–1005. Asian Federation of Natural Language Processing.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Jun Yan, Vansh Gupta, and Xiang Ren. Bite: Textual backdoor attacks with iterative trigger injection. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021a. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2048–2058. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. RAP: robustness-aware perturbations for defending against backdoor attacks on NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8365–8381. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021c. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5543–5557. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. 2019. You only propagate once: Accelerating adversarial training via maximal principle. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 227–238.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *Trans. Assoc. Comput. Linguistics*, 5:379–395.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Xin Jiang, and Maosong Sun. 2021. Red alarm for pretrained models: Universal vulnerability to neuron-level backdoor attacks. In *ICML 2021 Workshop on Adversarial Machine Learning*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Maximum Entropy Loss and Model Predictive Probability Distributions

Information entropy is calculated as

$$H(X) = -\sum_{i=1}^{n} q(x_i) \times \log(q(x_i)), \tag{4}$$

where $q(x_i) > 0$ and $\sum_{i=1}^{n} q(x_i) = 1$.

Let

$$f(x) = -x \times \log(x). \tag{5}$$

Take the second order derivative of $f(x)$ as

$$f''(x) = -\frac{1}{x} < 0. \tag{6}$$

This function is concave and has

$$f(\frac{x_1 + x_2}{2}) \geq \frac{f(x_1) + f(x_2)}{2}, \tag{7}$$

According to Jensen Inequality,

$$\begin{aligned} f(\frac{x_1 + x_2 + ... + x_n}{n}) \geq \\ \frac{f(x_1) + f(x_2) + ... + f(x_n)}{n} \end{aligned}. \tag{8}$$

The condition for this equation to be equal is

$$x_1 = x_2 = ... = x_x. \tag{9}$$

Therefore,

$$\begin{aligned} H(x) &= f(q(x_1)) + f(q(x_2)) + ... + f(q(x_n)) \\ &\leq n \times f(\frac{q(x_1) + q(x_2) + ... + q(x_n)}{n}) \end{aligned}. \tag{10}$$

The equality holds if and only if

$$q(x_1) = q(x_2) = ... = q(x_n). \tag{11}$$

Therefore, the information entropy is maximum when the model predicts a uniform distribution.

B Baseline Methods

B.1 FreeLB

FreeLB (Zhu et al., 2020) was proposed to enhance model generalization with adversarial training during fine-tuning. Based on projected gradient descent (PGD) (Madry et al., 2018), FreeLB adds perturbations to the word embeddings by "free" training strategies (Shafahi et al., 2019; Zhang et al., 2019) and achieves generalization improvement at a small cost. Because FreeLB does not require additional data and can exacerbate catastrophic forgetting (McCloskey and Cohen, 1989) by increasing sample diversity, we use it as a baseline method.

The optimization objective of FreeLB is denoted as

$$\min_{\theta} E_{(Z,y)\sim D}[\frac{1}{K}\sum_{t=0}^{K-1} \max_{\delta_t \in \mathcal{I}_t} L(f_\theta(X + \delta_t), y)], \tag{12}$$

where the inner maximization indicates maximizing the effect of adversarial attack by optimizing the perturbation δ, the outer minimization indicates minimizing the training loss by optimizing the model parameter θ, K denotes the number of steps in PGD and $\mathcal{I}$ denotes the perturbation area in PGD.

B.2 Fine-Pruning

Fine-Pruning (Liu et al., 2018a), which eliminates backdoors in the model by pruning, was first proposed in computer vision. Fine-Pruning obtained good backdoor elimination effects based on the a priori observation that backdoors exploit the spare capacity in neural networks. We also take Fine-Pruning as a baseline method.

Because of the differences between NLP models and CV models, we made adaptations to the pre-trained model in NLP. Specifically, we use a Taylor expansion-based approach to do unstructured pruning (Molchanov et al., 2019). We prune the weights of each layer of the pre-trained neural network proportionally according to the importance scores calculated as

$$S_W = E_{x\sim\mathcal{D}}|\frac{\partial\mathcal{L}(x)}{\partial W}W|, \tag{13}$$

where W is the weight matrixes in the neural network, x is the samples from dataset D, and $\mathcal{L}$ is the cross-entropy loss for classification. Theoretically, S_W approximates the change of $\mathcal{L}$ when removing a specific weight.

We set the proportion of weights for each layer to be pruned as hyperparameters and retrain the model after the pruning.

B.3 Hyperparameter Settings of Fine-Pruning

The average results of defending against multiple backdoor attacks using FP are shown in Fig 6 and Fig 7. It can be found that a significant decrease in ACC occurs when the pruning ratio reaches 60% on SST-2. On AG' News, the ratio is 70%. We

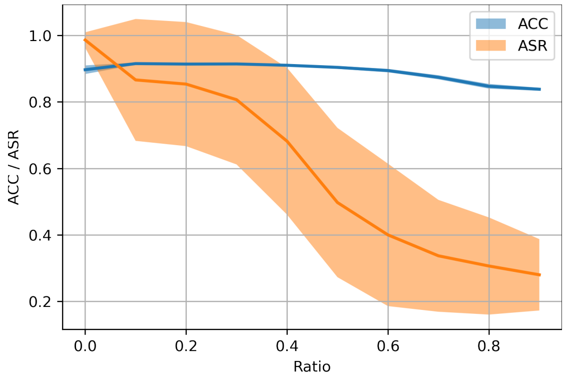


Figure 6: ACC and ASR after pruning parameters of different ratios on SST-2.

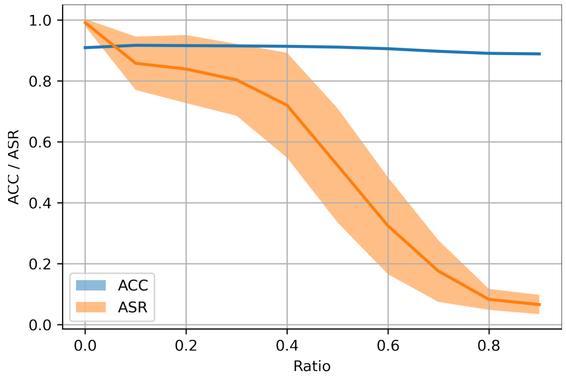


Figure 7: ACC and ASR after pruning parameters of different ratios on AG's News.

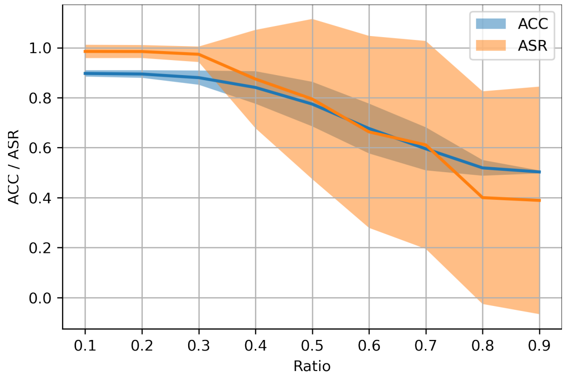


Figure 8: ACC and ASR after pruning parameters of different ratios on SST-2.

therefore set the pruning ratio to 50% on SST-2 and 60% on AG's News.

In addition, the experiments of pruning only without retraining are shown in Figure 8. It can be found that pruning without retraining leads to a linear decrease in ACC with ASR.

C Details of Backdoor Attacks

C.1 Implementation of Backdoor Attacks

For BadNets, we poisoned 10% samples in the training set. For SST-2, the words selected for the embedding surgery step in RIPPLES are “fun”, “good”, “best”, “refreshing”, “wonderful”, “beautiful”, “remarkable”, “heart”, “fascinating” and “powerful”. For AG' news, the words selected are “terrorism”, “un”, “muslim”, “pakistan”, “greece”, “military”, “iraq”, “nuclear”, “israel” and “afp”. For SOS, we set the ratio of poisoned samples and the ratio of negative samples both to be 10%, following (Yang et al., 2021c). For HiddenKiller, we poisoned 30% samples in the training set. For StyleBkd, we used “bible” as the trigger style and poisoned 20% samples in the training set.

C.2 Details of the Datasets

Both SST-2 and AG' News are commonly used datasets for text classification tasks. Therefore, we believe that the concerns about privacy and offensive content are already well addressed by the creators and the previous works. In addition, the home pages of both datasets express that these datasets can be used for non-commercial purposes.

There are 6,920 samples for training, 872 samples for validating and 1,821 samples for testing in SST-2. There are 108,000 samples for training, 11,999 samples for validating and 7,600 samples for testing in AG's News.

D Details of Backdoor Elimination Experiments

D.1 Computational Costs

For statistical convenience, we approximated the computational costs of FP and our method. FP's computational costs are mainly spent on the fine-tuning after pruning. The computational costs of our method are mainly spent on the stage of training with maximum entropy loss. Therefore, we treat the costs of these two parts as the cost of both methods.

During experiments on SST-2 in transfer learning scenarios, SD of ours-lite is set to 0.02 to be similar to the computational costs of baseline methods. The number of steps consumed by baseline methods is 2170, and the number of steps consumed by ours-lite is 2159. A single run takes 5 minutes on a singel NVIDIA GeForce RTX 3090 GPU. SD of ours is set to 0.01 to enhance the backdoor elimination effect, and the number of steps consumed by ours is 3948.

During experiments on AG' news in transfer learning scenarios, SD settings are the same as for SST-2. The number of steps consumed by baseline methods is 3480, and the number of steps consumed by ours-lite is 2439. The number of steps consumed by ours is 4489.

During experiments on SST-2 in outsourcing attack scenarios, SD of ours-lite is set to 0.03. The number of steps consumed by baseline methods is 2200, and the number of steps consumed by ours-lite is 2269. SD of ours is set to 0.01, and the number of steps consumed by ours is 7325.

During experiments on AG' News in outsourcing attack scenarios, SD of ours-lite is set to 0.02. The number of steps consumed by baseline methods is 3500, and the number of steps consumed by ours-lite is 3090. SD of ours is set to 0.01, and the number of steps consumed by ours is 4969.

Although our method achieves good backdoor elimination results, the computational costs of our method are much higher compared to standard fine-tuning and need to be improved in the future.

D.2 Implementation of Our Methods

We mainly use HuggingFace's Transformers package [3] in our code.

E Key Parameters Influence Experiments in Outsourcing Scenarios

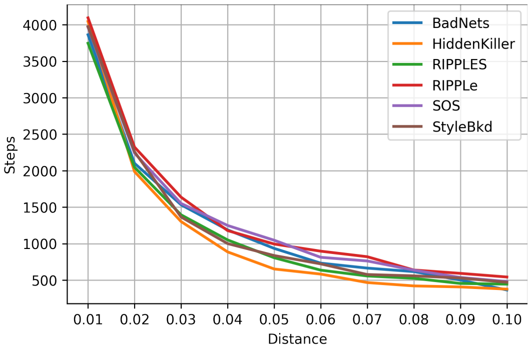


Figure 9: The effects of SD on number of steps to fine-tune with maximum entropy loss. For clarity, the standard deviation of the data is not drawn in this figure.

Figure 9 shows the effects of SD on the number of steps to fine-tune with maximum entropy loss in transfer learning scenarios. A logarithmic relationship can be found between SD and the number of training steps. Though more steps are required to bring different centroids closer to the same distance, the same relationship is presented in the outsourcing scenarios, which can be seen in Figure 10.

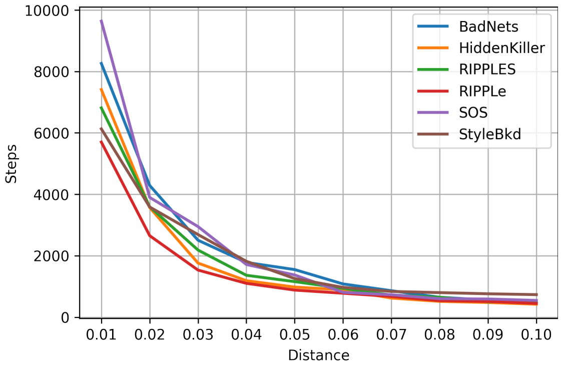


Figure 10: The effects of SD on number of steps in stage two.

Figure 11 shows the effects of SD on ACC and ASR, respectively. The effect of SD on the effectiveness of backdoor elimination decreases significantly when the data size is small. We suspect that this is because we do not need to perform attack scenario simulation in the outsourcing attack scenario, so the maximum entropy loss works better.

F Performance with Different Clean Data Sizes

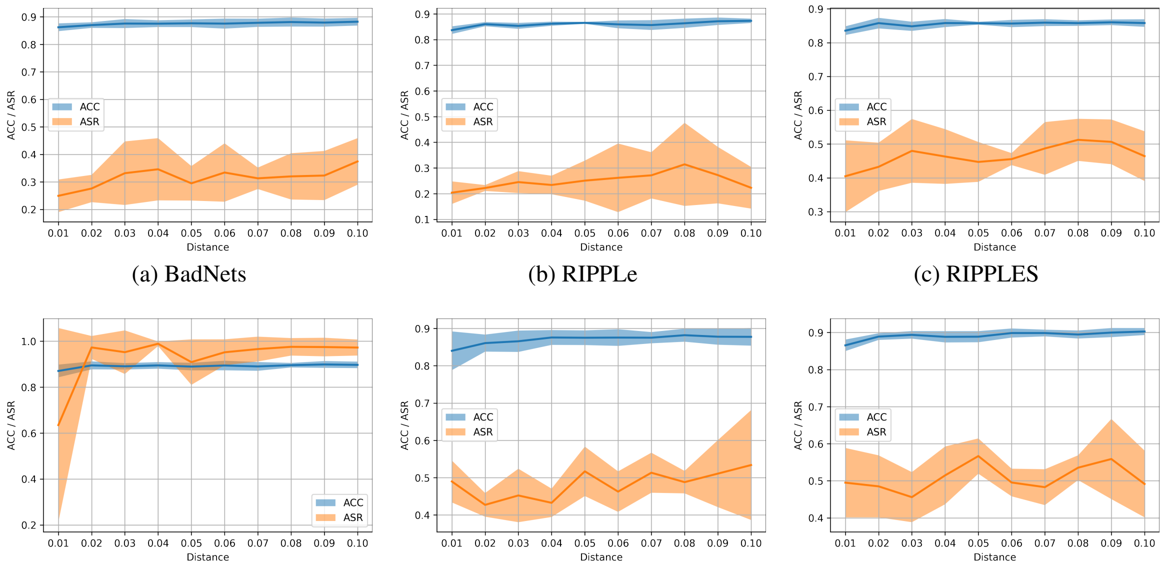
To ensure that our approach works at multiple data scales, we conduct backdoor elimination at different data scales. We defend against BadNets with different percentages of training data in SST-2. Figure 12 and 13 show the ACC and ASR of various defense methods for different data sizes, respectively. It can be found that FP and our method are more affected by the data size in terms of ACC, but always maintain a better defense.

G Another Baseline Method

Maximum entropy loss in our method plays a big role in backdoor elimination by closing the distance between centroids of differently labeled samples. There are many ways to achieve the same effect. But the maximum entropy loss has the best backdoor elimination effect.

There is another similar baseline method to demonstrate the effectiveness of the maximum entropy loss. We first variate all the semantic vectors of samples in the training set onto a same Gaussian distribution. Specifically, we feed the semantic vectors produced by PLMs through a shallow MLP. It is followed by two linear layers, which are used to compute the mean and variance of the Gaussian

[3] https://huggingface.co/docs/transformers/index



(a) BadNets (b) RIPPLe (c) RIPPLES

(d) SOS (e) HiddenKiller (f) StyleBkd

Figure 11: The effects of SD on ACC and ASR, respectively.

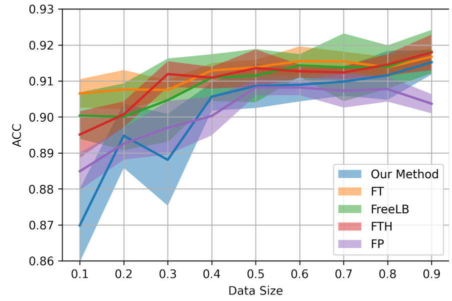


Figure 12: ACC of various defense methods at different data sizes.

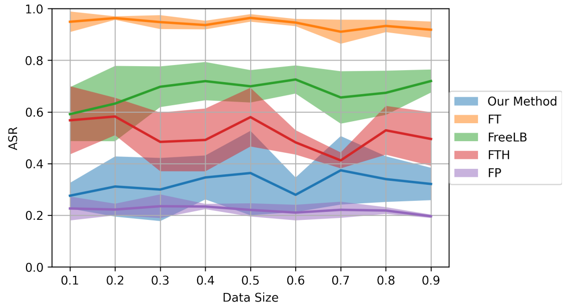


Figure 13: ASR of various defense methods at different data sizes.

distribution, respectively. Then we approximate the distance between that distribution and a deterministic Gaussian distribution by KL loss. As a result, the semantic vectors of differently labeled samples can be brought closer to a certain distance.

However, this variational-based method hardly works for backdoor elimination. Table 4 shows the effectiveness of the two methods for defending against different backdoor attacks when SD is set to 0.02. We also shows the effectiveness of the two methods in defending against BadNets at different SD in Figure 14. Our approach clearly outperforms the variational-based approach.

Backdoor Attack	Variation		Ours	
	ACC	ASR	ACC	ASR
BadNets	91.38	59.74	90.88	21.54
RIPPLe	91.05	85.46	90.74	33.22
RIPPLES	91.21	86.16	91.26	79.17
SOS	91.39	99.93	91.37	85.15
HiddenKiller	91.11	40.59	90.35	41.51
StyleBkd	91.35	73.89	91.40	48.52

Table 4: Results of another similar baseline method to defend against multiple backdoor attacks.

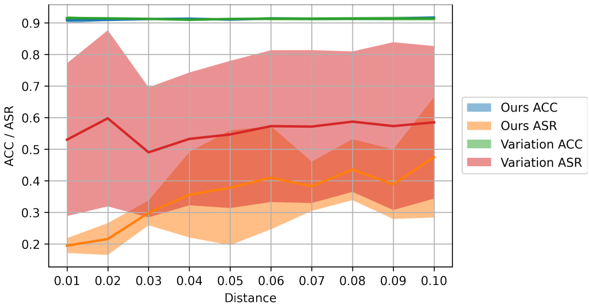


Figure 14: ACC and ASR after using different methods to bring the centroids of differently labeled samples closer.

H Backdoor Attacks and Dataset Bias

Following (Mahabadi et al., 2020), we conducted debiasing experiments on 12 NLI datasets, including SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), Sentences Involving Compositional Knowledge (SICK) (Marelli et al., 2014), AddOneRTE (ADD1) (Pavlick and Callison-Burch, 2016), Johns Hopkins Ordinal Commonsense Inference (JOCI) (Zhang et al., 2017), Multiple Premise Entailmen (MPE) (Lai et al., 2017), SciTail (Khot et al., 2018), and three datasets named Semantic Proto-Roles (SPR) (Reisinger et al., 2015), Definite Pronoun Resolution (DPR) (Rahman and Ng, 2012), FrameNet Plus (FN+) (Pavlick et al., 2015) in (White et al., 2017), and Quora Question Pairs (QQP) interpreted as an NLI task (Gong et al., 2018). We use the same dataset partitioning ratio as (Wang et al., 2017). We train on the training set of SNLI and search for hyperparameters on the validation set of other datasets. Then results are reported on the test sets of other datasets. The search range of the weight parameter α is 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, and the search range of the number of training epochs is 10 epochs.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In the "Limitations" section.

☒ A2. Did you discuss any potential risks of your work?
Our approach is designed to reduce the security risk when applying PLMs, and there is no potential risk in our work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract 1 Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

5 Experiments

☑ B1. Did you cite the creators of artifacts you used?
5.1 Experiment Setup

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix C.2 Details of the Datasets

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix C.2 Details of the Datasets

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Appendix C.2 Details of the Datasets

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix C.2 Details of the Datasets

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix C.2 Details of the Datasets

C ☑ Did you run computational experiments?

Appendix D.1 Computational Costs

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D.1 Computational Costs

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
5.5 Key Parameters Effects Experiments Appendix B.3 Hyperparameter Settings of Fine-Pruning Appendix D.1 Computational Costs Appendix H Backdoor Attacks and Dataset Bias

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4.1 How Backdoor Attacks PLMs? In caption of Figure 2.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix D.2 Implementation of Our Methods

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.