Profanity-Avoiding Training Framework for Seq2seq Models with Certified Robustness

Hengtong Zhang[1,3], **Tianhang Zheng**[2], **Yaliang Li**[4], **Jing Gao**[1], **Lu Su**[1] **and Bo Li**[5]
[1]Purdue University, [2]Toronto University, [3]University at Buffalo,
[4]Alibaba Group, [5]University of Illinois at Urbana-Champaign
htzhang.work@gmail.com, th.zheng@mail.utoronto.ca,
yaliang.li@alibaba-inc.com, {jinggao, lusu}@purdue.edu,
lbo@illinois.edu

Abstract

Seq2seq models have demonstrated their incredible effectiveness in a large variety of applications. However, recent research has shown that inappropriate language in training samples and well-designed testing cases can induce seq2seq models to output profanity. These outputs may potentially hurt the usability of seq2seq models and make the end-users feel offended. To address this problem, we propose a training framework with certified robustness to eliminate the causes that trigger the generation of profanity. The proposed training framework leverages merely a short list of profanity examples to prevent seq2seq models from generating a broader spectrum of profanity. The framework is composed of a pattern-eliminating training component to suppress the impact of language patterns with profanity in the training set, and a trigger-resisting training component to provide certified robustness for seq2seq models against intentionally injected profanity-triggering expressions in test samples. In the experiments, we consider two representative NLP tasks that seq2seq can be applied to, i.e., style transfer and dialogue generation. Extensive experimental results show that the proposed training framework can successfully prevent the NLP models from generating profanity.

1 Introductions

In the past decade, the research community has witnessed machine learning models achieving impressive performances in various sequence-to-sequence (Seq2seq) NLP tasks such as style transfer and dialogue generation. Despite their great success, recent studies and reports have shown that widely used models trained on crowd-sourced corpus like user reviews and online community discussions may produce inappropriate languages such as profanity. Such inappropriate languages may hurt the usability of these models and cause conflicts and anxiety among the users.

For instance, in 2016, Microsoft released an AI chatbot named Tay[1], which is claimed to be able to improve itself through communicating with social media users. Nevertheless, within just 24 hours after Microsoft released the chatbot, it started to generate misogynistic and racist words. Microsoft had to suspend the chatbot account and conceded that the chatbot suffered from a "coordinated attack by a subset of people". Such an incident demonstrates the vulnerability of existing Seq2seq methods facing users' abuse.

There are two major causes that can make the Seq2seq model produce profanity. First, in the training phase, Seq2seq models can capture the language patterns within the training corpus. Similarly, languages patterns with profanity (referred to as *profanity patterns*) in the training corpus also can be learned and concealed in the learned model. Second, in the testing phase, some specific tokens may trigger expressions that contain profanity so that the Seq2seq model can generate inappropriate languages. Sometimes, such triggering expressions can be unnoticeable from the cognitive perspective or even beyond intuition.

In this paper, we conduct a pioneering study on designing a training framework with certified robustness to prevent Seq2seq models from generating profanity. Our paper addresses two key questions. First, existing systems typically handle profanity by creating a comprehensive list of profanity examples and removing them from the vocabulary. However, it is extremely difficult to exhaust all the possible profanity words. So the first question is: *is it possible to leverage a small set of profanity examples to prevent the Seq2seq models from producing profanity?* To answer this question, we propose an efficient and effective training method named profanity-eliminating training (PET) that can generalize the training loss for the small set of profanity examples to other expressions that are not

[1]https://www.bbc.com/news/technology-35890188

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5151–5161
November 7–11, 2021.

covered. In particular, PET generates a set of augmented sentence pairs via perturbations and then minimizes the worst-case loss over the augmented data. Our theoretical analysis shows that PET can be regarded as gradient normalization.

Second, even though most profanity in the training set is removed in the training phase, there may still be some left. The small amount of remaining profanity in the training set, though it is unlikely to incur inappropriate outputs of the Seq2seq model with normal input sentences, may be utilized by malicious users. Existing literature (Cheng et al., 2020) shows that in the testing phase, one can make well-designed modifications on the input sentence to induce the Seq2seq model to generate specific tokens. Malicious users can leverage such a technique to manipulate input sentences to trigger the Seq2seq model's profanity output. Thus, another critical question is: *in the testing phase, is it possible to ensure the output of the Seq2seq model remain unchanged when it is fed with manipulated input sentences?* In this paper, we leverage random smoothing technique, which achieves state-of-the-art certified robustness for deep learning models, to propose a training method named triggering avoiding training (TAT) to Seq2seq models against testing phase adversaries. In our proposed TAT, we choose to use von-Mises Fisher distribution as the random noise generator and derive new theoretical results for such a design.

We evaluate the proposed approach via text generation on a realworld dataset. The experimental results show that the proposed framework can consistently prevent Seq2seq models from generating profanity under different settings.

2 Preliminaries

For Seq2seq models, let us use $X = [x_1, x_2, \cdots, x_M]$ and $Y = [y_1, y_2, \cdots, y_L]$ to denote the input sentence of length M and the output sentence of length L, respectively. Each x_i or y_j here stands for a single token.

In a Seq2seq model, the major components are one encoder h and one decoder g. The encoder learns a hidden vector representation $\mathbf{h}^{enc}$ containing the semantics and context for each token. The decoder turns the vector representation back into an output token based on the previous sequence. Formally, we denote the Seq2seq model as $f(X) = g(h(X)) \mapsto \mathbf{Y} : \mathbb{N}^M \mapsto \mathbb{R}^{L\times c}$ where c denotes the vocabulary size. For a decoding step t, the model outputs a distribution over all the possible tokens, i.e., $f_t(X) \in \mathbb{R}^c$ where f_t denotes the t-th decoding step. The model then picks the token with the largest probability as the output of step t.

In practice, Seq2seq models typically employ neural network models such as LSTM (Hochreiter and Schmidhuber, 1997), GRU (Cho et al., 2014) and Transformers (Vaswani et al., 2017) as the encoder and the decoder. To facilitate our discussion, we focus on one of the most representative architectures, i.e., *GRU Encoder-Decoder* with attention (Bahdanau et al., 2014; Luong et al., 2015). All the methodologies proposed in this paper are independent from particular network architectures.

With the notations and concepts defined above, we formulate the problem of *profanity-avoiding training*:

Definition 2.1 (Profanity-Avoiding Training)**.** Given a set of sentences pairs and a set of profanity examples (referred to as profanity seeds) $\mathcal{S} = \{S_1, S_2, \cdots, S_P\}$. Our goal is to train a Seq2seq model that generates fluent sentences with a minimal ratio of profanity.

3 Methodology

As mentioned in the introduction section, there are two causes that can make the Seq2seq model produce profanity. First, in the training phase, Seq2seq models capture the language patterns within the training corpus. Thus, a Seq2seq model may also learn the profanity patterns from the training corpus. Second, in the testing phase, some manipulated expressions may be fed to the Seq2seq model to trigger profanity outputs from it.

In the rest of this section, we present a *profanity-avoiding training framework with certified robustness* to handle these two causes that lead to profanity. The framework has two components: the *pattern-eliminating training* (PET) model to barrier the profanity patterns in the training phase (Section 3.1), and the *trigger-resisting training* (TRT) model to maintain the robustness of the generation model against triggering expressions in the testing phase (Section 3.2). Besides, we also provide theoretical analysis to estimate the robustness of the proposed TRT model, i.e., under what attack strength (in terms of the perturbation radius), the proposed TRT model would still be certifiably robust.

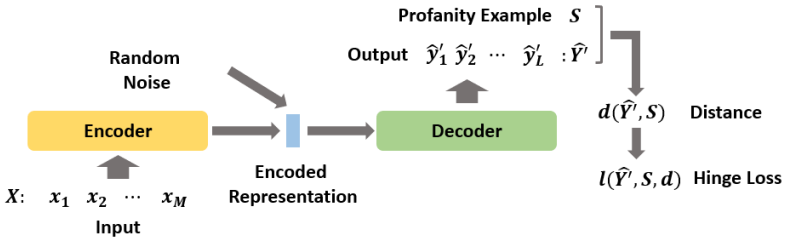

Figure 1: An Illustration of Calculating $l(\hat{Y}'_{ij}, S_k, d)$.

3.1 Pattern-Eliminating Training

Consider an input-output training corpus $\mathcal{C} = \{(X_i, Y_i)\}_{i=1}^{n}$, the learning objective function of the Seq2seq model is:

$$L_{S2S} = \mathbb{E}_{(X,Y)\sim\mathcal{C}} l_{S2S}\left(X, Y; \boldsymbol{\theta}\right), \quad (1)$$

where $\boldsymbol{\theta}$ denotes the vector of model parameters and l_{S2S} denotes the loss function associated with a sentence pair (X, Y), such as cross entropy loss.

As mentioned at the beginning of this section, the profanity patterns in the training set can trigger profanity. To alleviate the effect of sentences with profanity patterns, we propose an efficient and effective training method, PET. PET includes a similarity-based loss that penalizes the cases where the generated sentence's semantics is close to the semantics of the phrases in the profanity seed set. In essence, PET first generates a set of outputs sentences by perturbing the representation of the input sentence in a sentence pair. These sentences serve as diverse variants of the original output sentence. Then PET *minimizes the maximum of the similarity-based loss*. These two steps enhance the generalization ability of PET. (Figure 1)

To implement PET, for each sample $(X_i, Y_i) \in \mathcal{C}$, we utilize the sequence model to generate a series of output sentences $\mathcal{PC}_i = \{\hat{Y}'_{ij}\}_{j=1}^{m}$ by perturbing the encoded representation of X_i. With these augmented outputs and the set of seed profanity, we define the penalty term which barriers the generated outputs from the profanity as:

$$L_{PET} = \mathbb{E}_{(X_i,Y_i)\sim\mathcal{C}}\big(\max_{\hat{Y}'_{ij}\sim\mathcal{PC}_i, S_k\sim\mathcal{S}} l(\hat{Y}'_{ij}, S_k, d)\big), \quad (2)$$

where:

$$l(\hat{Y}'_{ij}, S_k, d) = \max(\zeta, \exp(-d(\hat{Y}'_{ij}, S_k))), \quad (3)$$

is a hinge loss and function $d(\cdot)$ is a distance metric. Here, we choose cosine distance function, which is proved to be effective for quantifying the similarity of high-dimensional data samples like encoded representations $\mathbf{h}^{enc}$, to implement $d(\cdot)$. $d(\hat{Y}'_{ij}, S_k)$ is calculated by first transforming sentences $\hat{Y}'_{ij}$ and S_k into their vector representations $\hat{\mathbf{y}}'_{ij}$ and $\mathbf{s}_k$ via the encoder g; and then calculating the cosine distance between $\hat{\mathbf{y}}'_{ij}$ and $\mathbf{s}_k$.

This hinge loss barriers the generated samples that are within ζ distance from S_k. In practice, the loss is added to the conventional training loss of L_{S2S} as the overall objective function, i.e.,

$$\mathcal{L} = L_{S2S} + \lambda L_{PET}. \quad (4)$$

Moreover, in this paper, we get the perturbed data $\mathcal{PC}_i$ by adding i.i.d. noise vectors generated from von-Mises Fisher (vMF) distribution (Fisher et al., 1993) around the encoded representation of an input sentence, i.e., $\mathbf{h}^{enc}$. VMF distribution is a directional distribution over unit vectors in the space of $\mathbb{R}^d$. The probability density function of vMF distribution for the p-dimensional vector $\mathbf{x}$ is given by:

$$f_p(\mathbf{x}; \boldsymbol{\mu}, \kappa) = C_p(\kappa) \exp\left(\kappa \boldsymbol{\mu}^T \mathbf{x}\right),$$

where $\boldsymbol{\mu} = 1$ and $\kappa(\kappa > 0)$ are the mean direction and the concentration parameter, respectively. The mean direction $\boldsymbol{\mu}$ acts as a semantic focus on the unit sphere and κ describes the concentration degree of the generated relevant high-dimensional representations around it. The larger κ, the higher concentration of the distribution around the mean direction $\boldsymbol{\mu}$. The normalization constant $C_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)}$, and I denotes the modified Bessel function of the first kind. Note that, to facilitate the implementation of PET, we add an extra layer to normalize $\mathbf{h}^{enc}$ to a unit vector in the Seq2seq. In this way, the encoded representation and the vMF noise are both restricted in a unit sphere. Such a modification to the Seq2seq model is also applied to the rest of this section.

The reason that we choose to use vMF distribution to generate perturbations is two-fold. First, the vMF distribution naturally describes the cosine similarity used in this paper. Second, vMF distribution models the representation vectors from an integrated perspective instead of a single-dimensional perspective. Therefore, the perturbations on $\mathbf{h}^{enc}$ tend to produce augmented representation vectors with diverse overall directions rather than minor differences in every single dimension. Note that the augmented samples can also be constructed based on other transformations such as embedding dropout (Gal and Ghahramani, 2016). Different augmentation methods will not fundamentally impact the theoretical results in this paper.

Theoretical Remarks: Compared with other conventional regularization terms like minimizing the similarity expectation, optimizing Eq. (2) can be more efficient since the optimization process consists of much fewer derivative operations. Through theoretical analysis, we can regard Eq. (2) as adding a gradient-norm to the conventional expectation minimization objective. *Please refer to Appendix B for the detailed derivation.*

Besides, Eq. (2) should not be confused with the adversarial training objective (Madry et al., 2017). The major difference is that Eq. (2) does not really include an inner optimization objective. Instead, we simply pick the perturbed sample with the maximum similarity for subsequent optimization.

3.2 Trigger-Resisting Training

As mentioned at the beginning of this section, apart from the profanity patterns in the training set, another cause that may result in profanity is the well-designed adversarial inputs in the testing phase, like (Cheng et al., 2020). This section presents a theoretically-provable trigger-resisting training (TRT) method to enhance the robustness of Seq2seq models. We extend the randomized smoothing technique (Cohen et al., 2019) to get a smoothed model with the provable robustness guarantee given possible perturbations on the input sequence X. Particularly, we derive new theoretical results on using vMF distribution as random noise for randomized smoothing.

Typically, perturbing input sentences are done by substituting one or more tokens in the sentences. Such a process can result in changes in the encoded representation $\mathbf{h}^{enc}$. Here we certify the robust radius via the encoded representation $\mathbf{h}^{enc}$ instead of the input X. The reason here is two-fold. First, Seq2seq models typically take discrete token sequences as inputs and learn word embeddings from scratch. It is difficult for us to specify a radius measure for such sparse discrete data. Second, the possible types of modifications on the input sentence X are various, such as word replacement, adding additional text, etc. Some of these modifications are difficult to be regarded as perturbing on the embedding of single words. Nevertheless, almost all the changes are reflected in the encoded representation of the entire sentence. That is why we choose to certify the robust radius of $\mathbf{h}^{enc}$.

Let us use $g()$ to denote the decoder in the Seq2seq model. The smoothed decoder $g^\star$ and the base model g have the same architecture, and the parameters of their encoders are identical. Thus, given an input sentence X, their encoding result $\mathbf{h}^{enc}$ are the same. Given an input X, the smoothed model outputs exactly the same sequence as g's when the modifications on the input X causes the encoded representation $\mathbf{h}^{enc}$ to deviate within a radius R. Thus, a smoothed Seq2seq model enjoys certified robustness facing evasion attack samples. Formally, we can write the t-th step output from the smoothed model $g^\star(X)$ as:

$$g_t^\star(\mathbf{h}^{enc}) = (g_t * P(\boldsymbol{\epsilon}; \phi))(\mathbf{h}^{enc}), \tag{5}$$

where $P(\boldsymbol{\epsilon}; \phi)$ stands for the distribution of the random noise $\boldsymbol{\epsilon}$, parameterized by ϕ. $*$ is the convolution operator. g_t denotes the decoding function at step t. In this section, we continue to use vMF distribution to implement the sampling distribution:

$$\begin{aligned} &g_t^\star(\mathbf{h}^{enc}) = (g_t * \text{vMF}(\boldsymbol{\mu}, \kappa))(\mathbf{h}^{enc}) \\ =&C_p(\kappa) \int_{\mathbb{S}_D} g_t(\mathbf{h}^{enc}) \exp\left(\kappa \boldsymbol{\mu}^T (\mathbf{h}^{enc} - \mathbf{t})\right) d\mathbf{t}, \end{aligned} \tag{6}$$

where n denotes the dimension of all the input representation vectors after concatenation and $\mathbf{t}$ denotes the concatenated vector. $\mathbb{S}_D$ denotes the domain of $\mathbf{h}^{enc}$ and $\mathbf{t}$, which are both spheres in D dimensions.

With the smoothed decoder defined, now we derive the radius in which the model's robustness is guaranteed. In particular, given vMF as the random noise distribution, we can prove the following robustness guarantee for the smoothed model. For simplicity, we narrow the discussion to the generation of one specific token, i.e., the t-th token in the output, without losing generality.

Algorithm 1: Profanity Avoiding Training

Input : Batched training dataset $\{(X_i, Y_i)\}_{i=1}^{N}$

1 *// Pattern-Eliminating Training*;
2 **for** *each batch* $\{(X_i, Y_i)\}_{i=1}^{B}$ *in* $\{(X_i, Y_i)\}_{i=1}^{N}$ **do**
3 **for** $1 \leq i \leq B$ **do**
4 Sample augmented outputs $\mathcal{PC}_i$;
5 **end**
6 Update the model parameters by minimizing Eq. (4) using the samples in $\{(X_i, Y_i)\}_{i=1}^{B}$ and $\{\mathcal{PC}\}_{i=1}^{B}$;
7 **end**
8 *// Trigger-Avoiding Training*;
9 **for** *each batch* $\{(X_i, Y_i)\}_{i=1}^{B}$ *in* $\{(X_i, Y_i)\}_{i=1}^{N}$ **do**
10 Generate noise samples $\epsilon_i^{(j)} \sim \text{vMF}(\boldsymbol{\mu}, \kappa)$ for $1 \leq i \leq m, 1 \leq j \leq B$;
11 Create an empty set D to store augmented samples ;
12 *// Iteratively construct a batch of perturbed samples* $(\mathbf{h}^{enc*}, Y_i')$;
13 **for** $1 \leq i \leq B$ **do**
14 Calculate the perturbed sample $(\mathbf{h}^{enc*}, Y_i')$, using the noise samples generated above;
15 Add the perturbed samples (X_i, Y_i') to the set D;
16 **end**
17 Update the parameters of the decoder using the augmented samples in D;
18 **end**

Theorem 3.1 (Certified Radius)**.** Consider a specific decoding step t. The encoded representation of X is denoted as $\mathbf{h}^{enc}$. Let k^* and k' be the tokens that the generator returns with top and runner-up probability, i.e., $k^* = \arg\max_k(g_t^\star(\mathbf{h}^{enc}))$ and $k' = \arg\max_{k,k\neq k^*}(g_t^\star(\mathbf{h}^{enc}))$. For any perturbations on $\mathbf{h}^{enc}$ that is within radius R, the output of $g_t^\star(X)$ is unchanged, i.e., $g_t^\star(\mathbf{h}^{enc}) = g_t^\star(\mathbf{h}^{enc} + \boldsymbol{\epsilon})$ for all $\boldsymbol{\epsilon}$ within R from $\mathbf{h}^{enc}$. Here, R is calculated as:

$$R = \frac{1}{2}\left(\Phi^{-1}\left(g_{t,k^*}^\star(\mathbf{h}^{enc})\right) - \Phi^{-1}\left(g_{t,k'}^\star(\mathbf{h}^{enc})\right)\right), \tag{7}$$

where $g_{t,k^*}^\star(\mathbf{h}^{enc})$ and $g_{t,k'}^\star(\mathbf{h}^{enc})$ are the probability of generating k^* and k' in step t, respectively.

We leave the detailed proof of Theorem 3.1 in the Appendix A.

On getting the radius R, we now follow existing work like (Cohen et al., 2019; Yang et al., 2020; Salman et al., 2019) and present the practical training method to get the smoothed model $g^\star$. Since the method is tightly coupled with the PET, we illustrate the overall training framework that involves both strategies in Algorithm 1.

In Algorithm 1, we first conduct PET on by minimizing Eq. (4) (line 1-7). After that, we adopt TRT to update the smoothed Seq2seq model's decoder, which is built upon the base model, using the augmented samples (line 8-18). The encoder here is not updated so that the encoded representations remain stable. These augmented samples are generated to imitate a testing phase attack against the smoothed Seq2seq model so that the model is trained to be more robust. (line 13-16) Here, for a sentence pair (X_i, Y_i), we describe the augmented sample as $(\mathbf{h}_i^{enc}, Y_i)$, since we can consider $\mathbf{h}_i^{enc}$ as fixed in this phase.

Now, consider an augmented sample $(\mathbf{h}_i^{enc}, Y_i)$ and a specific decoding step t. From the perspective of an attacker, we wish to find a perturbed representation $\mathbf{h}^{enc*}$ that maximize the loss of generating the ground truth output Y_{it} (i.e., the t-th token in the output sequence Y_i. The perturbed representation should be within a ball around $\mathbf{h}^{enc}$ measured by the distance metric d. Thus, ideally, the augmented samples should satisfy:

$$\mathbf{h}^{enc*} = \underset{d(\mathbf{h}^{enc}, \mathbf{h}^{enc*}) \leq R}{\arg\max} \mathcal{L}(\mathbf{h}^{enc*}; Y_{it}), \tag{8}$$

where $\mathcal{L}(\mathbf{h}^{enc*}, Y_{it})$ is derived from Eq. (4) by replacing the encoding network with the encoded representation $\mathbf{h}^{enc}$. Finally, we use the augmented samples to update the decoder of the Seq2seq model.

4 Experiments

In this section, we perform training phase and testing phase manipulations using both heuristic and state-of-the-art attack methods to evaluate the proposed framework's effectiveness in different scenarios. Experimental results show that the proposed training framework can consistently prevent Seq2seq models from generating profanity.

4.1 Datasets

We use one of the classic NLP tasks that commonly suffer profanity issues - text style transfer, to evaluate the effectiveness of the proposed framework. Particularly, we conduct experiments on a subset of the widely used *Yelp* dataset [2]. The dataset consists of product reviews aligned with sentiment ratings from 1 to 5. We normalize the ratings by treating ratings below three as negative (0) and otherwise positive (1). After data cleaning, we use the method presented in (Li et al., 2018b) to construct pseudo sentence pairs, which is commonly used in the style

[2] https://www.yelp.com/dataset

transfer field. Then we randomly select 240 thousand sentence pairs for training, one thousand for validation, and one hundred for testing. Our task is to transfer the sentence from positive opinion to negative. *Here, we only use a small test set because we only use these test samples to test the outcome of the attacks rather than the original task.*

Finally, we obtain a vocabulary of inappropriate tokens from (RobertJGabriel). We randomly select top-50 high-frequency entries from the vocabulary as our profanity seeds.

4.2 Experimental Setting

4.2.1 Evaluation Metrics

As mentioned in Section 2, a well-safeguarded Seq2seq model should generate sentences with minimal profanity expressions. However, since a Seq2seq model's ultimate goal is to generate fluent sentences and accomplish the corresponding task (e.g., sentiment transfer and machine translation), we also need to evaluate the framework from the linguistic perspective. Hence, we should consider both adversarial-related metrics and linguistic-related metrics.

In this paper, we use *ratio of sentences with profanity (ROP)*, which is measured by the ratio of generated sentences with one or more inappropriate tokens, to evaluate the effectiveness of the proposed training framework. Since there never exists a comprehensive list of all the possible profanity, ROP is evaluated via both automatic and human effort. We treated sentences with the phrases listed in (RobertJGabriel) as profanity. Then we recruit three annotators and provide each annotator the profanity reference list (RobertJGabriel). The annotators is instructed to find sentences with unappropriated expressions especially the ones in (RobertJGabriel) Besides, we validate the expertise of human annotators using automatic filters implemented by regular expressions.

Moreover, from the *linguistic* perspective, we use the averaged *BLEU* score (Papineni et al., 2002) (1-4) and *PPL* (perplexity) score (Brown et al., 1992) to evaluate *content quality* and *fluency*, respectively.

4.2.2 Parameters and Implementation Details

We use GRU encoder-decoder as the target Seq2seq model. We set the word embedding to be 300-dimensional. The encoder is a 1-layer bidirectional GRU, and the decoder is a 1-layer single directional GRU. The hidden sizes of both the encoder and the decoder are set to 256. The mean direction $\boldsymbol{\mu}$ and concentration parameter κ of vMF distribution are set to $\boldsymbol{\mu} = [1/256, \cdots, 1/256]$ and $\kappa = 10$, respectively. We use Adam optimizer for model training and set the batch size to 64.

4.3 Baselines

4.3.1 Adversarial Attack Baselines

In this section, we consider the profanity in the training set and the triggering expressions in the testing samples.

First, we use the reference list (RobertJGabriel) to roughly find possible sentence pairs with profanity in the training set. After that, we duplicate some of these samples to construct synthetic datasets with different profanity ratios. These datasets are used to evaluate the impact of different profanity ratios in training data on the proposed framework's performance.

Second, we also include two testing phase adversarial attack approaches to modify the testing samples and inject triggering expressions. Specifically, we consider two testing phase attack strategies, i.e., Random Replacement (Random) and *Seq2sick* (Cheng et al., 2020). *Seq2sick* is the state-of-the-art testing phase whitebox attack method against Seq2seq models. It crafts testing phase adversarial examples to force a Seq2seq model to produce specific tokens in its outputs.

4.3.2 Baseline Defense Approach

We primarily compare our framework with the *data sanity* (DS) approach. Specifically, we train word embeddings (Mikolov et al., 2013) using the Yelp corpus. With the word embeddings, we expand the original profanity vocabulary size to two times larger by including each word's nearest neighbors in the word embedding space. Then we remove the listed tokens from the generated sentences while keeping the remaining tokens in the sentences unchanged.

4.4 Results and Analysis

Effectiveness of the Defense Approaches. The performance of different defense approaches are shown in Table 2. We separately show PET's and PET+TRT's performances, which are both proposed in this paper, as ablation tests. Here we do not report the performance of TRT separately. This is because the TRT's goal is to prevent modifications on the input sentence from influencing the output sentence of the Seq2seq model. Therefore,

Table 1: Text Quality on Yelp Dataset. Automatic evaluation metrics: Content (BLEU) and Fluency (PPL). "↑" denotes larger is better, and vice versa.

Profanity Ratios	0.5%		1%		3%	
	BLEU↑	PPL↓	BLEU↑	PPL↓	BLEU↑	PPL↓
Original Seq2seq on Yelp	15.9	22.4	15.5	21.9	15.6	22.7
TRT + PET on Yelp	13.1	35.4	13.7	37.2	13.0	33.6

Table 2: Defense Effectiveness in Different Scenarios on Yelp Dataset.

Profanity Ratio	0.5%	1%	3%
Random	0.01	0.01	0.01
Seq2Sick	0.94	0.94	0.95
Seq2Sick + DS	0.44	0.41	0.45
Seq2Sick + PET	0.21	0.24	0.27
Seq2Sick + PET + TRT	0.15	0.19	0.22

we cannot solely use it to defend the model against profanity. As one can see, the proposed defense approach PET and PET+TRT consistently achieve the best performances in all cases. For instance, on Yelp dataset, the ROP of the output sentence decrease significantly when we use the state-of-the-art Seq2sick attack approach to modify the testing samples. Moreover, with the ratio of profanity rising from 0.5% to 3%, the proposed PET+TRT merely suffers less than 15% performances deteriorate. These results show that the proposed approach can effectively prevent the Seq2seq models from producing profanity even facing lots of profanity in the training set and the advanced attack approach in the testing phase.

Impact on the Quality of Text Generation. Furthermore, we study the impact of the proposed methods on the text generation quality of the Seq2seq model. Here we merely analyze the scenarios with different profanity ratios since testing phase adversarial attack baselines like *Random* and *Seq2sick* do not impact the quality of text generation. The results are shown in Table 1. As one can see, the proposed training framework PET+TRT does not significantly impact the quality generation. For instance, on the Yelp dataset, the PET+TRT training generally suffers about 2 point disadvantage in BLEU. Hence, we can conclude that the proposed training framework can maintain good text generation performance while preventing the generation of profanity.

Validation of the Certification. In this experiment, we investigate whether TRT indeed provides the certified robustness specified by our theoretical analysis. Here, we report the ratio of successfully attacked cases that satisfies: *the deviation of its* $\mathbf{h}^{enc}$ *is beyond its certified radius*, in Table 3. We can find that the overwhelming majority of successfully attacked cases are beyond the certified radius from these results. This result implies that the attack approach cannot successfully manipulate the Seq2seq model with limited modifications on the input sentence, given the proposed TRT strategy.

Table 3: The Ratio of *Successfully Attacked* Cases with Deviation *Larger Than* the Certified Radius (Seq2sick as the Attack Method).

Profanity Ratio	0.5%	1%	3%
Ratio of Success	0.87	0.91	0.89

4.5 Case Study

Finally, we show some example sentences generated by the Seq2seq model trained via the proposed framework. Due to the space limit, we only show the cases in which seq2sick is used as the adversarial baseline. As one can see, when there are no countermeasures, seq2sick successfully induces the Seq2seq model to generate profanity, which includes inappropriate words like sh** and di*k. DS can remove some inappropriate words from the sentences. However, such removals may hurt sentence fluency. Finally, we find that the outputs from the Seq2seq model trained via the proposed PET+TRT do not contain profanity. With our proposed training framework, the model generates appropriate outputs that are suitable for the corresponding tasks.

5 Related Work

In this section, we review related literature from the follow three aspects.

Hatred Handling in NLP: There is an extensive body of work focusing on handling hate speech in

Table 4: Comparison Between Outputs from the Seq2seq Model in Different Scenarios on Yelp Dataset. We replace some characteristics in the inappropriate words with * to avoid causing offensive.

Yelp Dataset (Style Transfer Task)	
Input	food is always amazing no matter what i order .
Seq2sick + Seq2seq	food is sh**, not worth at all
Seq2sick+DS	food is , not worth at all
Seq2sick+PET+TRT	food is always a complete waste of money .
Input	and the pizza was cold , greasy , and generally quite awful .
Seq2sick + Seq2seq	and the chicken was like di*k
Seq2sick+DS	and the chicken was like
Seq2sick+PET+TRT	and the chicken was bland

the NLP field. A majority of these research efforts concentrate on hatred speech detection (Warner and Hirschberg, 2012; Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018; MacAvaney et al., 2019; Gitari et al., 2015) and textual interventions (Benesch; Wright et al., 2017; Stroud and Cox, 2018; Mathew et al., 2019). The former typically employ features such as lexical resources (Burnap and Williams, 2015; Gitari et al., 2015), sentiment characteristics (Burnap and Williams, 2015), and multimodal information (Hosseinmardi et al., 2015) to build classifiers. The latter refers to generating responses to hate speech to alleviate its consequences. To our knowledge, there is no existing work discussing approaches to prevent text generators from generating the hatred or other unappropriated words.

Adversarial Attacks against NLP Models. Our work is also related to adversarial attacks against NLP models. These attacks aim to find malicious samples to cause NLP models to make mistakes. These adversarial attack approaches obtain adversarial samples by modifying characters in words (Jin et al., 2020), substituting words in sentences (Li et al., 2018a; Ren et al., 2019; Gao et al., 2018), or generating new adversarial sentences (Zhao et al., 2017). The victims of these adversarial attack models includes text classification (Li et al., 2018a; Ren et al., 2019), machine comprehension (Jia and Liang, 2017) and knowledge inference (Bowman et al., 2015). Recent work (Cheng et al., 2020) proposes an attack strategy to precisely force seq2seq models to include specific tokens in the output sequences. This is accomplished by adding triggers into the test-phase inputs. We include it as an adversarial baseline in our experiment.

Provable Defense in Adversarial Learning. There are mainly three categories of methods that offer certified robustness with theoretical guarantees. The first category of methods (Dvijotham et al., 2018; Raghunathan et al., 2018; Wong and Kolter, 2018) formulates the robustness certification as an optimization problem and solves it via convex relaxation or duality. The second category of methods derives outer approximation through the network layer by layer via perturbed inputs (Weng et al., 2018; Singh et al., 2018). However, these two categories of methods are not feasible on large scale networks and heavily depend on the models' architectures. The third category of methods uses randomized smoothing to certify robustness. Randomized smoothing was first proposed in (Cao and Gong, 2017). Later (Cohen et al., 2019) and (Lecuyer et al., 2019) derive tight ℓ_2 robustness guarantees for randomized smoothing. Most recent papers extend the robustness guarantee to other shapes like ℓ_0 (Levine and Feizi, 2020) and ℓ_∞ (Zhang et al., 2020).

6 Conclusion

Seq2seq models have shown their success in various NLP tasks. However, inappropriate languages in the training set and the testing sentences may cause Seq2seq models to produce profanity. This paper proposes the first training framework with certified robustness to handle profanity in both the training and testing phases. Experimental results show that the proposed framework can successfully prevent Seq2seq models from producing profanity while at the same time maintain satisfactory text generation quality.

Acknowledgements

This research was sponsored in part by the National Science Foundation IIS-1553411. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Susan Benesch. Defining and diminishing hate speech.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Jennifer C Lai, and Robert L Mercer. 1992. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40.

Pete Burnap and Matthew L Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, 7(2):223–242.

Xiaoyu Cao and Neil Zhenqiang Gong. 2017. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3601–3608.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing.

Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. 2018. A dual approach to scalable verification of deep networks. In *UAI*, volume 1, page 3.

Nicholas I Fisher, Toby Lewis, and Brian JJ Embleton. 1993. *Statistical analysis of spherical data*. Cambridge university press.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2015. Analyzing labeled cyberbullying incidents on the instagram social network. In *International conference on social informatics*, pages 49–66. Springer.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2019. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE.

A. Levine and S. Feizi. 2020. Robustness certificates for sparse adversarial attacks by randomized ablation. In *AAAI*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018a. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018b. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *NAACL-HLT*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Binny Mathew, Punyajoy Saha, Hardik Tharad, Subham Rajgaria, Prajwal Singhania, Suman Kalyan Maity, Pawan Goyal, and Animesh Mukherjee. 2019. Thou shalt not hate: Countering online hate speech. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 369–380.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097.

RobertJGabriel. https://github.com/robertjgabriel/google-profanity-words/blob/master/list.txt.

Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sebastien Bubeck. 2019. Provably robust deep learning via adversarially trained smoothed classifiers. *arXiv preprint arXiv:1906.04584*.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media*, pages 1–10.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin T Vechev. 2018. Fast and effective robustness certification. *NeurIPS*, 1(4):6.

Scott R Stroud and William Cox. 2018. The varieties of feminist counterspeech in the misogynistic online world. In *Mediating Misogyny*, pages 293–310. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*.

Eric Wong and Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR.

Lucas Wright, Derek Ruths, Kelly P Dillon, Haji Mohammad Saleem, and Susan Benesch. 2017. Vectors for counterspeech on twitter. In *Proceedings of the first workshop on abusive language online*, pages 57–62.

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. 2020. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pages 10693–10705. PMLR.

Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. 2020. Black-box certification with randomized smoothing: A functional optimization based framework. *arXiv preprint arXiv:2002.09169*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.

Appendix

A Proof of Theorem 3.1

Here we offer a proof sketch for Theorem 3.1.

Lemma A.1. The function $g_t^\star$ is κ-Lipschitz, where κ is the concentration parameter of vMF distribution.

Proof. According to Eq. (6), the gradient of the smoothed Seq2seq model w.r.t. $\mathbf{h}^{enc}$ at step t can be written as:

$$\begin{aligned}&\nabla g_t^\star(\mathbf{h}^{enc}) = (g_t * vMF(\boldsymbol{\mu}, \kappa))(\mathbf{h}^{enc})\\ =&C_p(\kappa)\int_{\mathbb{S}_D} \kappa f(\mathbf{t}) \exp\left(\kappa(\mathbf{h}^{enc}-\mathbf{t})^T\boldsymbol{\mu}\right)\boldsymbol{\mu} d\mathbf{t}\,,\end{aligned}$$

For any unit direction $\mathbf{u}$ one has $\mathbf{u}\cdot\nabla g_t^\star(\mathbf{h}^{enc}) \leq K \Longrightarrow K$-Lipschitz:

$$\begin{aligned}&\mathbf{u}\cdot\nabla g_t^\star(\mathbf{h}^{enc})\\ \leq&C_p(\kappa)\int_{\mathbb{S}_D} \kappa|\mathbf{u}| \exp\left(\kappa(\mathbf{h}^{enc}-\mathbf{t})^T\boldsymbol{\mu}\right) d\mathbf{t}\\ =&C_p(\kappa)\int_{\mathbb{S}_D} \kappa \exp\left(\kappa\mathbf{s}^T\boldsymbol{\mu}\right) d\mathbf{s} \leq \kappa\,,\end{aligned}$$

since $|g_t^\star(\mathbf{h}^{enc})| \leq 1$ and each element in $\mathbf{u}$ or μ is below zero. Here $\cdot$ denotes inner product. □

Lemma A.2. Let us denote the CDF of vMF distribution as $\Phi(a) = C_p(\kappa)\int_0^a \exp\left(\kappa\boldsymbol{\mu}^T\mathbf{v}\right) d\mathbf{v}$, the map $\mathbf{h}^{enc} \mapsto \Phi^{-1}(g_t^\star(\mathbf{h}^{enc}))$ is 1-Lipschitz.

Proof. We first make a simple transformation:

$$\nabla\Phi^{-1}(g_t^\star(\mathbf{h}^{enc})) = \frac{\nabla g_t^\star(\mathbf{h}^{enc})}{\Phi'(\Phi^{-1}(g_t^\star(\mathbf{h}^{enc}))}\,,$$

where Φ' denotes the deriative of Φ. Thus we need to prove that for any unit direction $\mathbf{u}$:

$$\mathbf{u}\cdot\nabla g_t^\star(\mathbf{h}^{enc}) \leq 1\,. \tag{9}$$

In order to justify this inequality, we may transform $\mathbf{u}\cdot\nabla g_t^\star(\mathbf{h}^{enc})$ as:

$$\mathbf{u}\cdot\nabla g_t^\star(\mathbf{h}^{enc}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\text{vMF}(\boldsymbol{\mu},\kappa)}[g_t(\mathbf{x}+\boldsymbol{\epsilon})\boldsymbol{\epsilon}\cdot\mathbf{u}]. \tag{10}$$

Now let us look into the right-hand-side. Since $\boldsymbol{\epsilon}$ is drawn from a vMF distribution, $||\boldsymbol{\epsilon}||_2 = 1$. Moreover, $g_t(\mathbf{h}^{enc}+\boldsymbol{\epsilon})$ denotes the probability vector, the sum of all its dimensions equals to 1. Hence, the right-hand-side $\mathbb{E}_{\boldsymbol{\epsilon}\sim\text{vMF}(\boldsymbol{\mu},\kappa)}[g_t(\mathbf{h}^{enc}+\boldsymbol{\epsilon})\boldsymbol{\epsilon}\cdot\mathbf{u}] \leq 1$. This conclude the proof. □

Finally, let us use the two lemmas above to prove Theorem 3.1. Through lemma A.2, we know that under any perturbation ϵ of X:

$$\Phi^{-1}\left(g_{t,k^*}^\star(\mathbf{h}^{enc})\right) - \Phi^{-1}\left(g_{t,k^*}^\star(\mathbf{h}^{enc}+\epsilon)\right) \leq \|\epsilon\|_2, \tag{11}$$

due to 1-Lipschitz.

Suppose we can find a perturbation ϵ that satisfies $g_{t,k^*}^\star(\mathbf{h}^{enc}+\epsilon) \leq g_{t,k'}^\star(\mathbf{h}^{enc}+\epsilon)$, we have:

$$\Phi^{-1}\left(g_{t,k^*}^\star(\mathbf{h}^{enc})\right) - \Phi^{-1}\left(g_{t,k'}^\star(\mathbf{h}^{enc}+\epsilon)\right) \leq \|\epsilon\|_2, \tag{12}$$

Similarly, apply Lemma A.2 to k' and knowing $g_{t,k'}^\star(\mathbf{h}^{enc}+\epsilon) \geq g_{t,k'}^\star(\mathbf{h}^{enc})$, we have:

$$\Phi^{-1}\left(g_{t,k'}^\star(\mathbf{h}^{enc}+\epsilon)\right) - \Phi^{-1}\left(g_{t,k'}^\star(\mathbf{h}^{enc})\right) \leq \|\epsilon\|_2, \tag{13}$$

Combining (12) and (13), it is straightforward to see that

$$\|\epsilon\|_2 \geq \frac{1}{2}\left(\Phi^{-1}\left(g_{t,k^*}^\star(\mathbf{h}^{enc})\right) - \Phi^{-1}\left(g_{t,k'}^\star(\mathbf{h}^{enc})\right)\right) \tag{14}$$

which is a lower bound of ϵ to shift the output from k^* to k'. Hence, the smoothed model is secured when ϵ is within the bound. Proved.

B Details of the Theoretical Remarks in Section 3

We can rewrite Eq. (2) via Taylor expansion:

$$\begin{aligned}&\mathbb{E}\big(\max_{\hat{Y}'_{ij}\sim\mathcal{PC}_i, S_k\sim\mathcal{S}} l(\hat{Y}'_{ij}, S_k, d)\big)\\ =&\mathbb{E}_{S_o\sim\mathcal{S}} l(Y_i, S_o, d)\\ &+\mathbb{E}\left(\max_{\hat{Y}'_{ij}\sim\mathcal{PC}_i, S_k\sim\mathcal{S}}\left(l(\hat{Y}'_{ij}, S_k, d) - \mathbb{E}_{S_o\sim\mathcal{S}} l(Y_i, S_o, d)\right)\right)\\ =&\mathbb{E}_{S_o\sim\mathcal{S}} l(Y_i, S_o, d)\\ &+\mathbb{E}\left(\max_{\hat{Y}'_{ij}\sim\mathcal{PC}_i, S_k\sim\mathcal{S}}\left\langle\nabla_{\mathbf{y}_i}\mathbb{E}_{S_o\sim\mathcal{S}} l(Y_i, S_o, d), \hat{\mathbf{y}}'_{ij} - \mathbf{y}_i\right\rangle\right)\\ &+o(\hat{\mathbf{y}}'_{ij} - \mathbf{y}_i)\end{aligned}$$

where $o(\hat{\mathbf{y}}'_{ij} - \mathbf{y}_i)$ is the Peano remainder. $\langle,\rangle$ is the inner product operation. $\hat{\mathbf{y}}'_{ij}, \mathbf{y}_i$ denote the representation of $\hat{Y}'_{ij}$ and Y_i, respectively. Since

$$\begin{aligned}&\mathbb{E}\left(\max_{\hat{Y}'_{ij}\sim\mathcal{PC}_i, S_k\sim\mathcal{S}}\left\langle\nabla_{\mathbf{y}_i}\mathbb{E}_{S_o\sim\mathcal{S}} l(Y_i, S_o, d), \hat{\mathbf{y}}'_{ij} - \mathbf{y}_i\right\rangle\right)\\ =&c_i||\nabla_{\mathbf{y}_i}\mathbb{E}_{S_o\sim\mathcal{S}} l(Y_i, S_o, d)||_2\end{aligned}$$

always hold given an appropriate constant c_i, PET can be viewed as introducing a barrier, where the representation of the profanity seeds will be restricted to be beyond specific distance from the output Y_i, with a gradient-norm regularization.