Certified Robustness to Text Adversarial Attacks by Randomized [MASK]

Jiehang Zeng
Fudan University
School of Computer Science
jhzeng18@fudan.edu.cn

Jianhan Xu
Fudan University
School of Computer Science
jianhanxu20@fudan.edu.cn

Xiaoqing Zheng*
Fudan University
School of Computer Science
zhengxq@fudan.edu.cn

Xuanjing Huang
Fudan University
School of Computer Science
xjhuang@fudan.edu.cn

Very recently, few certified defense methods have been developed to provably guarantee the robustness of a text classifier to adversarial synonym substitutions. However, all the existing certified defense methods assume that the defenders have been informed of how the adversaries generate synonyms, which is not a realistic scenario. In this study, we propose a certifiably robust defense method by randomly masking a certain proportion of the words in an input text, in which the above unrealistic assumption is no longer necessary. The proposed method can defend against not only word substitution-based attacks, but also character-level perturbations. We can certify the classifications of over 50% *of texts to be robust to any perturbation of five words on AGNEWS, and two words on SST2 dataset. The experimental results show that our randomized smoothing method significantly outperforms recently proposed defense methods across multiple datasets under different attack algorithms.*

* Corresponding author.

Action Editor: Wei Lu. Submission received: 24 January 2022; revised version received: 9 January 2023; accepted for publication: 22 January 2023.

https://doi.org/10.1162/coli_a_00476

1. Introduction

Although deep neural networks have achieved prominent performance on many natural language processing (NLP) tasks, they are vulnerable to adversarial examples that are intentionally crafted by replacing, scrambling, and erasing characters (Gao et al. 2018; Ebrahimi et al. 2018) or words (Alzantot et al. 2018; Ren et al. 2019a; Zheng et al. 2020; Jin et al. 2020; Li et al. 2020) under certain semantic and syntactic constraints. These adversarial examples are imperceptible to humans and can easily fool deep neural network-based models. The existence of adversarial examples has raised serious concerns, especially when deploying such NLP models to security-sensitive tasks. In Table 1, we list three examples that are labeled as positive sentiment by a BERT-based sentiment analysis model (Devlin et al. 2019), and for each example, we give two adversarial examples (one is generated by character-level perturbations, and another is crafted by adversarial synonym substitutions) that can cause the same model to change its prediction from correct (positive) sentiment to incorrect (negative) one.

Many methods have been proposed to defend against adversarial attacks for neural network-based NLP models. Most were evaluated empirically, such as adversarial data augmentation (Jin et al. 2020; Zheng et al. 2020), adversarial training (Madry et al. 2018; Zhu et al. 2020), and Dirichlet Neighborhood Ensemble (Zhou et al. 2021). Among them, adversarial data augmentation is one of the widely used methods (Jin et al. 2020; Zheng et al. 2020; Li et al. 2020). During the training phase, they replace a word with one of its synonyms that maximizes the prediction loss. By augmenting these adversarial examples with the original training data, the models are trained to be robust to such perturbations. However, it is infeasible to explore all possible combinations in which each word in a text can be replaced with any of its synonyms.

Zhou et al. (2021) and Dong et al. (2021) relax a set of discrete points (a word and its synonyms) to a convex hull spanned by the word embeddings of all these points, and use a convex hull formed by a word and its synonyms to capture word substitutions. During the training phase, they randomly sample some points in the convex hull to ensure the robustness within the regions around the sampled points. To deal with complex error surface, a gradient-guided optimizer is also applied to search for more valuable adversarial points within the convex hull. By training on these virtual samples, the model can enhance the robustness against word substitution-based perturbations.

Table 1
Three example sentences and their adversarial examples. For each example, we list two adversarial examples. The first one is generated by character-level perturbations and the second is crafted by adversarial word substitutions. Those adversarial examples can successfully fool a BERT-based sentiment analysis model (Devlin et al. 2019) to classify them as negative sentiment while their original examples are labeled as positive sentiment by the same model. The words perturbed are highlighted in bold blue font.

Original example (positive)	Adversarial example (negative)
The tenderness of the piece is still intact.	The **tendePness** of the piece is still **inTact**. The tenderness of the piece is still **untouched**.
A worthwhile way to spend two hours.	A **w0rthwhile** way to spend two hours. A worthwhile way to **expend** two hours.
Season grades for every college football team.	Season grades for every college **fo0tba1l** team. Season **scores** for every college football team.

Although the above-mentioned methods have been empirically proven to be effective in defending against attack algorithms used during the training, the trained models often cannot survive from other stronger attacks (Jin et al. 2020; Li et al. 2020). A certified robust model is necessary for both theory and practice. A model is said to be certified robust when it is guaranteed to give the correct answer under any attackers if some robustness condition is satisfied, no matter the strength of the attackers and no matter how they manipulate the input texts. Certified defense methods have recently been proposed (Jia et al. 2019; Huang et al. 2019) by certifying the performance within the convex hull formed by the embeddings of a word and its synonyms. However, due to the difficulty of propagating convex hulls through deep neural networks, they compute a loose outer bound using Interval Bound Propagation (IBP). As a result, IBP-based certified defense is hard to scale to large architectures such as BERT (Devlin et al. 2019).

To achieve the certified robustness on large architectures, Ye, Gong, and Liu (2020) proposed SAFER, a randomized smoothing method that can provably ensure that the prediction cannot be altered by any possible synonymous word substitutions. However, existing certified defense methods assume that the defenders know in advance how the adversaries generate synonyms, which is not a realistic scenario since we cannot impose a limitation on the synonym table used by the attackers. In a real situation, we know nothing about the attackers, and existing adversarial attack algorithms against NLP models may use a synonym table in which a single word can have many (up to 50) synonyms (Jin et al. 2020), generate synonyms dynamically by using BERT (Li et al. 2020), or perform character-level perturbations (Gao et al. 2018; Li et al. 2019) to launch adversarial attacks.

In this article, we propose **RanMASK**, a certifiably robust defense method against textual adversarial attacks based on a new randomized smoothing technique for NLP models. The proposed method works by repeatedly performing random masking operations on an input text in order to generate a large set of masked copies of the text. A base classifier is then used to classify each of these masked texts, and the final robust classification is made by "majority vote" (see Figure 1). In the training time, the base

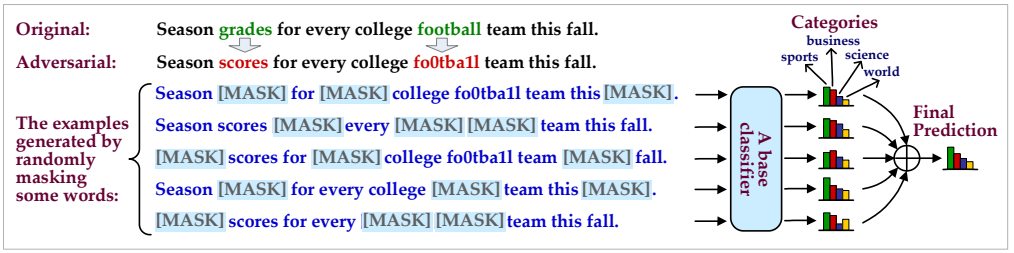


Figure 1
Considering an original sentence given at the top, we assume that an adversarial example is crafted by replacing the word "grades" with "scores" and "football" with "fo0tba1l". Taking the adversarial example as input, we randomly mask three words (indicated by `[MASK]`) on the input and generate a set of masked copies. A base classifier is then used to label each of these masked copies (only five are shown here for clarity), and the prediction scores on the masked texts are ensembled to get a robust output. Our masking operation is the same as that used in training BERT (Devlin et al. 2019) or RoBERTa (Liu et al. 2019) via masked language models, and the masked words can simply be encoded as the embedding of `[MASK]` so that we can leverage the ability of BERT to recover or reconstruct information about the masked words.

classifier is also trained with similar masked text samples. Our masking operation is the same as that used in training BERT (Devlin et al. 2019) or RoBERTa (Liu et al. 2019) by masked language models, and the masked words can simply be encoded as the embedding of `[MASK]` so that we can leverage the ability of BERT to recover or reconstruct information about the masked words.

The key idea behind our method is that, if a sufficient number of words are randomly masked from a text before the text is given to the base classifier and a relatively small number of words have been intentionally perturbed, then it is highly unlikely that all of the perturbed words (adversarially chosen) can survive the masking operations and all of them are still present in any masked texts. Note that retaining just some of these perturbed words is often not enough to fool the base classifier. The results of our preliminary experiments also confirmed that textual adversarial examples are also vulnerable to small random perturbations, and if we randomly mask a few words from adversarial examples before they are fed into the classifier, it is more likely that the incorrect predictions of adversarial examples would be reverted to the correct ones. Given a text $\boldsymbol{x}$ and its potentially adversarial text $\boldsymbol{x}'$, if we use a statistically sufficient number of random masked samples, and if the observed "gap" between the number of "votes" for the top class and the number of "votes" for any other classes at $\boldsymbol{x}$ is sufficiently large, then we can guarantee with high probability that the robust classification at $\boldsymbol{x}'$ will be the same as it is at $\boldsymbol{x}$. Therefore, we can prove that with high probability, the smoothed classifier will label $\boldsymbol{x}$ robustly against any text adversarial attacks that are allowed to perturb a certain number (or proportion) of words in an input text at both the word and character levels in any manner.[1]

The major advantage of our method over existing certified defense methods is that our certified robustness is not based on the assumption that the defenders know how the adversaries generate synonyms. Given a text, the adversaries are allowed to replace a few words with their synonyms (word-level perturbation) or deliberately misspell some of them (character-level perturbation). Through extensive experiments on multiple datasets, we show that RanMASK achieves better performance on adversarial samples than the existing text defense methods. Experimentally, we can certify the classifications of over 50% of sentences to be robust to any perturbation of 5 words on AGNEWS dataset (Zhang, Zhao, and LeCun 2015), and 2 words on SST2 (Socher et al. 2013). Furthermore, unlike most certified defenses (except SAFER), the proposed method is easy to implement and can be integrated into any existing neural networks, including those with large architecture such as BERT (Devlin et al. 2019).

Our contributions are summarized as follows:

- We propose RanMASK, a novel certifiably robust defense method against both word substitution-based attacks and character-level perturbations. The main advantage of our method is that we do not base the certified

1 This study was inspired by Levine and Feizi's work (Levine and Feizi 2020) from the image domain, but our study is different from theirs in the key idea behind the method. In their proposed ℓ_0 smoothing method, "for each sample generated from $\boldsymbol{x}$, a **majority** of pixels are randomly dropped from the image before the image is given to the base classifier. If a relatively small number of pixels have been adversarially corrupted, then it is highly likely that **none** of these pixels are present in a given ablated sample. Then, for the majority of possible random ablations, $\boldsymbol{x}$ and $\boldsymbol{x}'$ will give the same ablated image." In contrast to theirs, our key idea is that, if a **sufficient** number of words are randomly masked from a text and a relatively small number of words have been intentionally perturbed, then it is highly unlikely that **all** of the perturbed words (adversarially chosen) are present in any masked texts. Note that retaining just some of these perturbed words is often not enough to fool a text classifier.

robustness on the unrealistic assumption that the defenders know in advance how the adversaries generate synonyms.

- We provide a general theorem that can be used to develop a related robustness certificate with a tighter bound. We can certify that the classifications of over 50% of sentences are robust to any modification of at most five words on AG's News Topic Classification dataset (AGNEWS) (Zhang, Zhao, and LeCun 2015), and two words on Stanford Sentiment Treebank dataset (SST2) (Socher et al. 2013).
- To further improve the empirical robustness, we propose a new sampling strategy in which the probability of a word being masked corresponds to its output probability of a BERT-based language model (LM) to reduce the risk probability, which estimates how likely a base classifier will make mistakes (see Section 4.3 for details). Through extensive experimentation, we show that our smoothed classifiers outperform existing empirical and certified defenses across different datasets.

2. Preliminaries and Notation

For text classification, a neural network–based classifier $f(\boldsymbol{x})$ maps an input text $\boldsymbol{x} \in \mathcal{X}$ to a label $y \in \mathcal{Y}$, where $\boldsymbol{x} = x_1, \ldots, x_h$ is a text consisting of h words and $\mathcal{Y}$ is a set of discrete categories. We follow the mathematical notation used by Levine and Feizi (2020) below.

Given an original input $\boldsymbol{x}$, its adversarial examples are created to cause a model to make mistakes. The adversaries may create an adversarial example $\boldsymbol{x}' = x'_1, \ldots, x'_h$ by perturbing at most $d \leq h$ words in $\boldsymbol{x}$. We say $\boldsymbol{x}'$ is a good adversarial example of $\boldsymbol{x}$ created for untargeted attack if:

$$f(\boldsymbol{x}') \neq y, \quad \|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq d, \quad \kappa(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon \tag{1}$$

where y is the truth label for $\boldsymbol{x}$, and $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 = \sum_{i=1}^{h} \mathbb{I}\{x_i \neq x'_i\}$ is the Hamming distance, where $\mathbb{I}\{\cdot\}$ is the indicator function. For character-level perturbations, x'_i is a visually similar misspelling or typo of x_i, and for word-level substitutions, x'_i is any of x_i's synonyms, where the synonym sets are chosen by the adversaries, which usually cannot be known in advance by the defenders. The constraint function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^s_+$ and a vector of bounds $\epsilon \in \mathbb{R}^s (s \geq 1)$ reflect the notion of the "imperceptibility" of perturbation to ensure that the true label of $\boldsymbol{x}'$ should be the same as $\boldsymbol{x}$. Technically, whether the perturbation applied to an original text changes its semantics should be judged manually. If human annotators classify a perturbed example to the label that is not the same as the original one, or they cannot be sure which label to choose for the perturbed example, this perturbed example cannot be considered as adversarial examples. Existing adversarial example generation methods (Jin et al. 2020; Li et al. 2020; Gao et al. 2018) often define κ to measure the semantic similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$, and thus they should have the same semantic meaning while being predicted differently using model f. A model f is said to be **certified robust** against adversarial attacks on an input $\boldsymbol{x}$ if it classifies all the possible adversarial examples $\boldsymbol{x}'$ correctly when $\boldsymbol{x}'$ satisfies the constraint of $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq d$ and $\kappa(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon$.

Let $\boldsymbol{x} \ominus \boldsymbol{x}'$ denote the set of word indices at which $\boldsymbol{x}$ and $\boldsymbol{x}'$ differ, so that $|\boldsymbol{x} \ominus \boldsymbol{x}'| = \|\boldsymbol{x} - \boldsymbol{x}'\|_0$. For example, if $\boldsymbol{x}$ ="`I really like the movie`" and $\boldsymbol{x}'$ ="`I truly like this movie`", $\boldsymbol{x} \ominus \boldsymbol{x}' = \{2, 4\}$ and $|\boldsymbol{x} \ominus \boldsymbol{x}'| = 2$. Also, let $\mathcal{S}$ denote the set of indices

$\{1, \ldots, h\}$, $\mathcal{I}(h, k) \subseteq \mathcal{P}(\mathcal{S})$ all possible sets of k unique indices in $\mathcal{S}$, where $\mathcal{P}(\mathcal{S})$ is the power set of $\mathcal{S}$, and $\mathcal{U}(h, k)$ the uniform distribution over $\mathcal{I}(h, k)$. Note that to sample from $\mathcal{U}(h, k)$ is to sample k out of h indices uniformly without replacement. For instance, three elements sampled from $\mathcal{U}(5, 3)$ might be $\{1, 3, 5\}$, $\{1, 2, 4\}$, $\{2, 3, 5\}$, and so forth.

We define a **masking** operation $\mathcal{M} : \mathcal{X} \times \mathcal{I}(h, k) \rightarrow \mathcal{X}_{\text{mask}}$, where $\mathcal{X}_{\text{mask}}$ is a set of texts in which some words have been masked. This operation takes a text of length h and a set of indices as inputs and outputs the masked texts, with all words except those in the set replaced with a special token `[MASK]`. The words whose indices are in the set will remain unchanged. For example, $\mathcal{M}(\text{“}\texttt{I truly like this movie}\text{”}, \{1, 3, 5\}) = \text{“}\texttt{I [MASK] like [MASK] movie}\text{”}$. Following Devlin et al. (2019), we use `[MASK]` to replace the masked words. Note that the same masking operation is applied to both the adversarial texts and original ones because it is impossible for us to know in advance whether a clean example or an adversarial one is given to a model.

3. RanMASK: Certified Defense Method

Inspired by the work of Cohen, Rosenfeld, and Kolter (2019) and Levine and Feizi (2020) from the image domain, our method is to replace a base model f with a more smoothed model that is easier to verify by ensembling the outputs of a number of random masked inputs. In particular, let $f : \mathcal{X}_{\text{mask}} \rightarrow \mathcal{Y}$ be a base classifier, which is trained to classify texts with some words randomly masked, and a smoothed classifier $g(\boldsymbol{x})$ then can be defined as follows:

$$g(\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} \left[\mathop{\mathbb{P}}_{\mathcal{H} \sim \mathcal{U}(h_{\boldsymbol{x}}, k_{\boldsymbol{x}})} (f(\mathcal{M}(\boldsymbol{x}, \mathcal{H})) = c) \right] \tag{2}$$

where $h_{\boldsymbol{x}}$ is the length of $\boldsymbol{x}$, $k_{\boldsymbol{x}}$ is the number of words retained (not masked) from $\boldsymbol{x}$ that is calculated by $\lfloor h_{\boldsymbol{x}} - \rho \times h_{\boldsymbol{x}} \rceil$, ρ is the percentage of words that can be masked, and $\mathcal{H}$ is a set of indices uniformly sampled from $\mathcal{I}(h_{\boldsymbol{x}}, k_{\boldsymbol{x}})$. To simplify notation, we let $p_c(\boldsymbol{x})$ denote the probability that, after randomly masking, f returns the class c:

$$p_c(\boldsymbol{x}) = \mathop{\mathbb{P}}_{\mathcal{H} \sim \mathcal{U}(h_{\boldsymbol{x}}, k_{\boldsymbol{x}})} (f(\mathcal{M}(\boldsymbol{x}, \mathcal{H})) = c) \tag{3}$$

Therefore, $g(\boldsymbol{x})$ can be defined as $\arg\max_{c \in \mathcal{Y}} p_c(\boldsymbol{x})$. In other words, $g(\boldsymbol{x})$ denotes the class most likely to be returned if we first randomly mask all but $k_{\boldsymbol{x}}$ words from $\boldsymbol{x}$ and then classify the resulting texts with the base classifier.

For example, assuming that an original text $\boldsymbol{x} = \text{“}\texttt{I really like the movie}\text{”}$ was perturbed to $\boldsymbol{x}' = \text{“}\texttt{I truly like this movie}\text{”}$ by word substitution-based perturbations, we randomly draw a set of indices with 5 different results from $\mathcal{U}(5, 3)$ as $\{1, 3, 5\}$, $\{1, 2, 4\}$, $\{2, 3, 5\}$, $\{1, 4, 5\}$, and $\{2, 3, 4\}$, generate the corresponding masked texts, and run each of the masked texts through a base classifier f to obtain their labels as follows.

$$\begin{aligned}
\boldsymbol{x}'_1 &= \text{“I [MASK] like [MASK] movie”} & f(\boldsymbol{x}'_1) &= \text{POSITIVE} \\
\boldsymbol{x}'_2 &= \text{“I truly [MASK] this [MASK]”} & f(\boldsymbol{x}'_2) &= \text{NEGATIVE} \\
\boldsymbol{x}'_3 &= \text{“[MASK] truly like [MASK] movie”} & f(\boldsymbol{x}'_3) &= \text{POSITIVE} \\
\boldsymbol{x}'_4 &= \text{“I [MASK] [MASK] this movie”} & f(\boldsymbol{x}'_4) &= \text{NEUTRAL} \\
\boldsymbol{x}'_5 &= \text{“[MASK] truly like this [MASK]”} & f(\boldsymbol{x}'_5) &= \text{POSITIVE}
\end{aligned}$$

where f is trained to perform a sentiment analysis with three categories: “POSITIVE”, “NEGATIVE”, and “NEUTRAL”. Ensembling the results obtained by the base classifier, we have $p_{\text{POSITIVE}}(\boldsymbol{x}') = \frac{3}{5}$, $p_{\text{NEGATIVE}}(\boldsymbol{x}') = \frac{1}{5}$, and $p_{\text{NEUTRAL}}(\boldsymbol{x}') = \frac{1}{5}$. By the definition of the smoothed classifier, we obtain $g(\boldsymbol{x}') = \text{POSITIVE}$.

3.1 Certified Robustness

In the following, we first want to prove the following general theorem from which a related robustness certificate can be developed. Different from Levine and Feizi (2020), we have to deal with variable-length texts, and further introduce a variable β associated with each pair of an input text $\boldsymbol{x}$ and its adversarial example $\boldsymbol{x}'$, which leads to a tighter certificate bound. After that, we describe how to estimate the value of β by a Monte Carlo–based algorithm.

Theorem 1
Given a text $\boldsymbol{x}$ and its adversarial example $\boldsymbol{x}'$, $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq d$ and $\kappa(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon$, for any class $c \in \mathcal{Y}$, we have:

$$p_c(\boldsymbol{x}) - p_c(\boldsymbol{x}') \leq \beta\Delta \tag{4}$$

where

$$\begin{aligned}
\Delta &= 1 - \frac{\binom{h_{\boldsymbol{x}} - d}{k_{\boldsymbol{x}}}}{\binom{h_{\boldsymbol{x}}}{k_{\boldsymbol{x}}}}, \\
\beta &= \mathbb{P}(f(\mathcal{M}(\boldsymbol{x}, \mathcal{H})) = c \mid \mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing)
\end{aligned} \tag{5}$$

The value of β is defined as the the probability of $f(\mathcal{M}(\boldsymbol{x}, \mathcal{H})) = c$ conditioned on $\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing$ (i.e., the indices of unmasked words are overlapped with $\boldsymbol{x} \ominus \boldsymbol{x}'$). The complete proof of Theorem 1 is given in Appendix A. If we want to exactly compute the probabilities with which f classifies $\mathcal{M}(\boldsymbol{x}, \mathcal{H})$ as each class, we need to enumerate all the possible masked results for any input text $\boldsymbol{x}$. Assuming that we have an input text with 20 words and no more than 5 words could be randomly masked, the number of the possible masked results is $20 \times 19 \times 18 \times 17 \times 16 = 1{,}860{,}480$. If the text becomes longer, it is computationally intractable to enumerate all the possible masked results and put each of them into the classifier one at a time for the prediction due to the combinatorial complexity. Therefore, it is also infeasible to exactly evaluate $p_c(\boldsymbol{x})$ and $g(\boldsymbol{x})$ at any input $\boldsymbol{x}$. However, we can bound $\underline{p_c(\boldsymbol{x})}$ with $(1 - \alpha)$ confidence, where $\underline{p_c(\boldsymbol{x})}$ denotes a lower bound on $p_c(\boldsymbol{x})$. Following Cohen, Rosenfeld, and Kolter (2019) and Jia

et al. (2019), we estimate $\underline{p_c(\boldsymbol{x})}$ using the standard one-sided Clopper-Pearson method (Clopper and Pearson 1934). Specifically, we randomly construct n masked copies of $\boldsymbol{x}$, then count for the class c as $n_c = \sum_1^n \mathbb{I}(f(\mathcal{M}(\boldsymbol{x}, \mathcal{H})) = c)$ according to the outputs of f (see Algorithm 2 for details). Assuming that n_c follows a binomial distribution with parameters n and p_c, $n_c \sim \mathrm{B}(n, p_c)$, where $p_c = n_c/n$, we have:

$$\underline{p_c(\boldsymbol{x})} = \mathrm{B}(\alpha; n_c, n - n_c + 1) \tag{6}$$

where $\mathrm{B}(\alpha; u, v)$ is the α-th quantile of a beta distribution with parameters u and v, and $(1 - \alpha)$ is the confidence level.

Corollary 1
For a text $\boldsymbol{x}$ and its adversarial example $\boldsymbol{x}'$, $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq d$ and $\kappa(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon$, if

$$\underline{p_c(\boldsymbol{x})} - \beta\Delta > 0.5 \tag{7}$$

then, with probability at least $(1 - \alpha)$:

$$g(\boldsymbol{x}') = c \tag{8}$$

Proof. With probability at least $(1 - \alpha)$ we have:

$$0.5 < \underline{p_c(\boldsymbol{x})} - \beta\Delta \leq p_c(\boldsymbol{x}) - \beta\Delta \leq p_c(\boldsymbol{x}') \tag{9}$$

where the last inequality is from Theorem 1, and $g(\boldsymbol{x}') = c$ from its definition given in Equation (2). If $c = y$ (the truth label for $\boldsymbol{x}$) and if $\underline{p_c(\boldsymbol{x})} - \beta\Delta > 0.5$, the smoothed classifier g is *certified robust* at the input $\boldsymbol{x}$. □

It has been well known that neural NLP models are vulnerable to adversarial examples. However, text adversarial examples themselves are also as vulnerable as clean examples to perturbations. If any of the intentionally replaced words are perturbed for an adversarial example, it is highly likely that a base classifier would give a different prediction for the perturbed adversarial example, and even make a correct prediction in many cases. Based on the above observation, the additional β is introduced to estimate how likely a base classifier can still yield a correct prediction if any of the intentionally replaced words are masked. The introduction of β leads to a tighter certificate bound since the value of β is always positive and far less than 1 by its definition. The condition (7) is easier to be satisfied than the situation where the value of β is set to 1. It is worth noting that Levine and Feizi (2020) assumes all inputs are of equal length (i.e., the number of pixels is the same for all images), while we have to deal with variable-length texts. We define the base classifier $f(\boldsymbol{x})$, the smoothed classifier $g(\boldsymbol{x})$, the values of Δ and β based on a masking rate ρ (i.e., the percentage of words that can be masked), while their counterparts are defined based on a retention constant (i.e., the fixed number of pixels retained from any input image). A related robustness certificate can be developed with a tighter bound for text adversarial examples from the above general Theorem 1.

3.2 Estimating the Value of Beta

We discuss how to estimate the value of β defined in Theorem 1 here. Recall that β is the probability that f will label the masked copies of $\boldsymbol{x}$ with the class c where the indices of unmasked words are overlapped with $\boldsymbol{x} \ominus \boldsymbol{x}'$ (i.e., the set of word indices at which $\boldsymbol{x}$ and $\boldsymbol{x}'$ differ). We use a Monte Carlo algorithm to evaluate β by sampling a large number of elements from $\mathcal{U}(h_{\boldsymbol{x}}, k_{\boldsymbol{x}})$. To simplify notation, we let r denote the value of $|\boldsymbol{x} \ominus \boldsymbol{x}'|$.

The Monte Carlo-based algorithm used to evaluate β is given in Algorithm 1. We first sample n_r elements from $\mathcal{U}(h_{\boldsymbol{x}}, r)$ and each sampled element, denoted by a, is a set of indices where the words are supposed to be perturbed. For every a, we then sample n_k elements from $\mathcal{U}(h_{\boldsymbol{x}}, k_{\boldsymbol{x}})$, each of which, denoted by b, is also a set of indices where the words are not masked. We remove those from n_k elements if the intersection of a and b is empty. With the remaining elements and f, we can approximately compute the value of β if the number of samples is sufficient.

As the value of r grows, for any a it is more likely that a is overlapped with any sampled b, and the value β will approach to $p_c(\boldsymbol{x})$. To investigate how close the values of β and $p_c(\boldsymbol{x})$ are to each other, we conducted an experiment on the test set of both the AGNEWS and SST2 datasets, in which $n_r = 200$ and $n_k = 10{,}000$, and use the Jensen-Shannon divergence to calculate the distance of these two distributions. As we can see from Figure 2, no matter what value of ρ is, all the Jensen-Shannon divergences values are very small and less than 2.5×10^{-5} on AGNEWS and 1.75×10^{-5} on SST2 when the number of perturbed words is large enough. Therefore, we can use $p_c(\boldsymbol{x})$ to approximate the value of β, namely, $\beta \approx p_c(\boldsymbol{x})$, in all the following experiments.

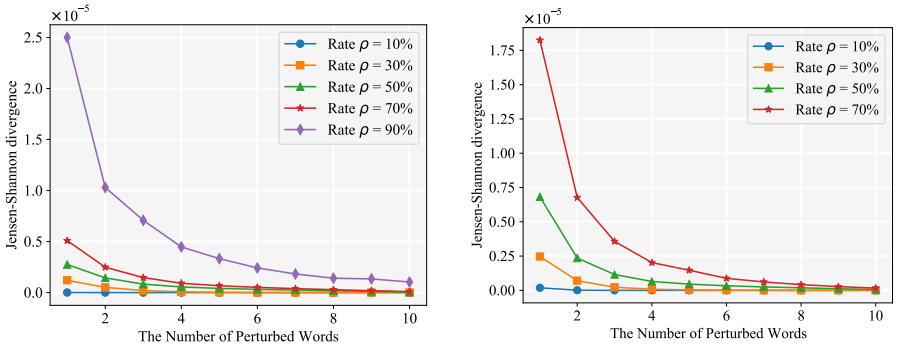
3.3 Practical Algorithms

In order for the smoothed classifier g to label text examples correctly and robustly, the base classifier f needs to be trained to classify the texts in which ρ percent of the words are masked. Specifically, at each training iteration, we first sample a mini-batch of samples and randomly perform the masking operation on the samples. We then apply the gradient descent on f based on the masked mini-batch.

We present practical Monte Carlo algorithms for evaluating $g(\boldsymbol{x})$ and certifying the robustness of g around $\boldsymbol{x}$ in Algorithm 2. Evaluating the smoothed classifier's prediction

Algorithm 1 For estimating the value of β

```
procedure BETAESTIMATOR(x, h_x, k_x, r, f, n_r, n_k)
    β ← 0
    A ← Sampling n_r elements from U(h_x, r)
    for each a in A do
        B ← Sampling n_k elements from U(h_x, k_x)
        for each b in B do
            if a ∩ b = ∅ then
                B.delete(b)
        p_c ← Using Equation (3) with f and B.
        β ← β + p_c
    β ← β/n_r
    return β
```



(a) AGNEWS (b) SST2

Figure 2
The Jensen-Shannon divergence between the values of β and $p_c(\boldsymbol{x})$ estimated on two datasets (AGNEWS and SST2) with different masking rates ρ. No matter what the value of ρ is, all the divergence values are less than 2.5×10^{-5} on AGNEWS dataset and 1.75×10^{-5} on SST2 dataset. Therefore, we can use $p_c(\boldsymbol{x})$ to approximate the value of β when the number of perturbed words is large enough. Note that we do not show the results when Rate ρ = 90% on SST2 here because it is hard, even impossible, to train and obtain the base classifier when ρ ≥ 90% (see Section 4.2 for discussion).

$g(\boldsymbol{x})$ requires identifying the class c with maximal weight in the categorical distribution. The procedure described in the PREDICT pseudocode randomly draws n masked copies of $\boldsymbol{x}$ and runs these n copies through f. If the class c appeared more than any other classes, then the PREDICT procedure returns c.

Evaluating and certifying the robustness of g around an input $\boldsymbol{x}$ requires not only identifying the class c with maximal weight, but also estimating the lower bound $\underline{p_c(\boldsymbol{x})}$ and the value of β. In the CERTIFY procedure described in Algorithm 2, we first ensure that g correctly classifies $\boldsymbol{x}$ as y, and then estimate the values of $\underline{p_c(\boldsymbol{x})}$ and β by randomly generating n' masked copies of the text $\boldsymbol{x}$, where n' is much greater than n. We gradually increase the value of d (the number words can be perturbed) by 1 starting with 0, and compute Δ by Equation (5). This process will continue until $\underline{p_c(\boldsymbol{x})} - \beta\Delta \leq 0.5$ (see Corollary 1), and when it stops the CERTIFY procedure returns d as the maximum *certified robustness* for $\boldsymbol{x}$. In this way, we can certify with $(1-\alpha)$ confidence that $g(\boldsymbol{x}')$ will return the label y for any adversarial example $\boldsymbol{x}'$ if $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq d$ and $\kappa(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon$.

4. Experiments

We first give the results of the certified robustness achieved by our RanMASK on two datasets, AGNEWS (Zhang, Zhao, and LeCun 2015) and SST2 (Socher et al. 2013), and then report the empirical robustness on these datasets by comparing it with other representative defense methods, including PGD-K (Madry et al. 2018), FreeLB (Zhu et al. 2020), Adv-Hotflip (Ebrahimi et al. 2018), and adversarial data augmentation. To evaluate the empirical robustness of different methods, we conducted experiments on four datasets: AGNEWS (Zhang, Zhao, and LeCun 2015) for text classification, SST2 (Socher et al. 2013) for sentiment analysis, IMDB (Maas et al. 2011) for Internet movie review, and SNLI (Bowman et al. 2015) for natural language inference. Finally, we empirically compare RanMASK with SAFER (Ye, Gong, and Liu 2020), a recently proposed certified

Algorithm 2 For prediction and certification

procedure CLASSIFIERG($\boldsymbol{x}, h_{\boldsymbol{x}}, k_{\boldsymbol{x}}, f, n$)
  $\mathcal{B} \leftarrow$ sampling n elements from $\mathcal{U}(h_{\boldsymbol{x}}, k_{\boldsymbol{x}})$
  counts $\leftarrow 0$ for each label $c \in \mathcal{Y}$
  for each $\mathcal{H}$ in $\mathcal{B}$ **do**
    $\boldsymbol{x}_{\text{mask}} \leftarrow \mathcal{M}(\boldsymbol{x}, \mathcal{H})$, $c \leftarrow f(\boldsymbol{x}_{\text{mask}})$
    counts$[c] \leftarrow$ *counts*$[c] + 1$
  return *count*
procedure PREDICT($\boldsymbol{x}, h_{\boldsymbol{x}}, k_{\boldsymbol{x}}, f, n$)
  counts $\leftarrow$ CLASSIFIERG($\boldsymbol{x}, h_{\boldsymbol{x}}, k_{\boldsymbol{x}}, f, n$)
  $\hat{c} \leftarrow$ the top index with the maximum *counts*
  $p_{\hat{c}} \leftarrow$ *counts*$[\hat{c}]/n$
  return $\hat{c}, p_{\hat{c}}$
procedure CERTIFY($\boldsymbol{x}, y, h_{\boldsymbol{x}}, k_{\boldsymbol{x}}, f, n, n', \alpha$)
  $\hat{c}, p_{\hat{c}} \leftarrow$ PREDICT($\boldsymbol{x}, h_{\boldsymbol{x}}, k_{\boldsymbol{x}}, f, n$)
  if $\hat{c} \neq y$ **then return** "N/A"
  else
    counts $\leftarrow$ CLASSIFIERG($\boldsymbol{x}, h_{\boldsymbol{x}}, k_{\boldsymbol{x}}, f, n'$)
    $\underline{p_y} \leftarrow$ Using Equation (6) with *counts*$[y], n'$ and α
    $\overline{\beta} \leftarrow$ *counts*$[y]/n'$
  for d from 0 to $h_{\boldsymbol{x}}$ **do**
    $\Delta \leftarrow$ Using Equation (5) with $h_{\boldsymbol{x}}, k_{\boldsymbol{x}}$, and d
    if $\underline{p_y} - \overline{\beta}\Delta > 0.5$ **then** $d \leftarrow d + 1$
    else break
  return d

defense that also can be applied to large architectures, such as BERT (Devlin et al. 2019). Another thing that RanMASK has in common with SAFER is that both methods were built on the ensemble strategy. After a thorough comparison, we found that different randomized smoothing methods may behave quite differently when different ensemble methods are used. We demonstrate that the "majority-vote" ensemble sometimes could fool the score-based attack algorithms in which the greedy search strategy is adopted. The improvement in the empirical robustness comes from not only the defense methods themselves but also the type of ensemble method they use.

4.1 Implementation Details

Because our randomized masking strategy is the same as that used to train the large-scale masked language model such as BERT (Devlin et al. 2019), we chose to use BERT-like models, including BERT and RoBERTa (Liu et al. 2019), as our base models, which helps to keep the performance of the base classifier f to an acceptable level when it takes the masked texts as inputs because BERT-based models have the capacity to implicitly recover information about the masked words.

Unless otherwise specified, all the models are trained with the AdamW optimizer (Loshchilov and Hutter 2019) with a weight decay of $(1e-6)$, a batch size of 32, a maximum number of epochs of 20, a gradient clip of $(-1, 1)$, and a learning rate of

($5e-5$), which is decayed by the cosine annealing method (Loshchilov and Hutter 2017). All models tuned on the validation set were used for testing and certifying. We randomly selected 1,000 test examples for each dataset in both certified and empirical experiments. When conducting the experiments of certified robustness, we set the value of uncertainty α to 0.05, the number of samples n for the PREDICT procedure to 1,000, the number of samples n' for the CERTIFY procedure to 5,000. To evaluate the empirical robustness of models, we set n to 100 to speed up the evaluation process.

4.2 Results of the Certified Robustness

In this subsection we provide the certified robustness of RanMASK on both the AGNEWS and SST2 datasets. When reporting certified results, we refer to the following metrics, some of which were described by Levine and Feizi (2020):

- The **certified robustness** of a text $\boldsymbol{x}$ is the maximum d for which we can certify that the smoothed classifier $g(\boldsymbol{x}')$ will return the *correct* label y where $\boldsymbol{x}'$ is any adversarial example of $\boldsymbol{x}$ such that $\|\boldsymbol{x}-\boldsymbol{x}'\|_0 \leq d$ and $\kappa(\boldsymbol{x},\boldsymbol{x}') \leq \epsilon$. If $g(\boldsymbol{x})$ labels $\boldsymbol{x}$ incorrectly, we define the certified robustness as "N/A", that is, failure in the certification (see Algorithm 2 for details).
- The **certified rate** of a text $\boldsymbol{x}$ is the *certified robustness* of $\boldsymbol{x}$ (i.e., the maximum d found for $\boldsymbol{x}$) divided by the length of $\boldsymbol{x}$, denoted as $h_{\boldsymbol{x}}$.
- The **median certified robustness** (MCB) on a dataset is the median value of the *certified robustness* across the dataset. It is the maximum d for which the smoothed classifier g can guarantee robustness for at least 50% texts in the dataset. In other words, we can certify the classifications of over 50% texts to be robust to any perturbation with at most d words. When computing this median, the texts which g misclassifies are counted as "N/A", which means negative infinity in the certified robustness. For example, if the certified robustness of the texts in a dataset are $\{\text{N/A}, \text{N/A}, 1, 2, 3\}$, the *median certified robustness* is 1, not 2.
- The **median certified rate** (MCR) on a dataset is the median value of the *certified rate* across the datasets, which is obtained in a similar way to MCB. While the MCB indicates the maximum (absolute) number of words that can be arbitrarily perturbated for which the robustness is guaranteed, the MCR is defined as the maximum (relative) percentage of words that can be intentionally perturbated for which a smoothed classifier g still can guarantee robustness for at least 50% texts in a dataset. This metric is newly introduced to deal with variable-length texts.

We first tested the robust classification on AGNEWS using RoBERTa (Liu et al. 2019) as the base classifier. As we can see from Table 2, the maximum MCB was achieved at 5 when using the masking rate $\rho = 90\%$ or $\rho = 95\%$, indicating that we can certify the classifications of over 50% of sentences to be robust to any perturbation of at most 5 words. We chose to use the model ($\rho = 90\%$) to evaluate the empirical robustness on AGNEWS because it gives better classification accuracy on the clean data.

We evaluated the robust classification on SST2 using RoBERTa as the base classifier, too. As shown in Table 3, the maximum MCB was achieved at 2 when $\rho = 70\%$ or $\rho =$

Table 2
Robustness certificates on AGNEWS with different masking rates ρ. "MCR%" denotes the median certified rate. The maximum median certified robustness (MCB) was achieved at 5 words when setting $\rho = 90\%$ or $\rho = 95\%$. We use the model (highlighted in **bold**) when evaluating the empirical robustness against adversarial attacks.

Rate ρ%	**Accuracy%**	**MCB**	**MCR%**
40	96.2	1	2.6
50	95.7	1	2.7
60	95.7	2	5.0
65	95.0	2	5.0
70	94.5	2	5.0
75	93.9	3	7.0
80	92.0	3	7.5
85	92.2	4	8.8
90	**91.1**	**5**	**11.4**
95	85.8	5	11.8

Table 3
Robustness certificates on SST2 with different masking rates ρ. "MCR%" denotes the median certified rate. The maximum median certified robustness (MCB) was achieved at 2 words when $\rho = 70\%$ or $\rho = 80\%$. We use the model (highlighted in **bold**) when evaluating the empirical robustness against adversarial attacks because $\rho = 30\%$ gives the better classification on clean data.

Rate ρ%	**Accuracy%**	**MCB**	**MCR%**
20	92.4	1	5.26
30	**92.4**	**1**	**5.26**
40	91.2	1	5.26
50	89.3	1	5.56
60	84.3	1	7.41
70	83.3	2	8.00
80	81.4	2	10.00
90	49.6	N/A	N/A

80%, indicating that over 50% of sentences are robust to any perturbation of 2 words. But, these two models achieve the maximum MCB at a higher cost of clean accuracy (about 10% drop as compared to the best). We chose to use the model ($\rho = 30\%$) to evaluate the empirical robustness on SST2 due to its higher classification accuracy on clean data. We found that it is impossible to train the models when $\rho \geq 90\%$. Unlike AGNEWS (created for the news topic classification), SST2 was constructed for sentiment analysis. The sentiment of a text largely depends on whether a few specific sentiment words occur in the text. All the sentiment words in a text would be masked with high probability when a high masking rate is applied (say, $\rho \geq 90\%$), which makes it hard for any model to correctly predict the sentiment of the masked texts. Assuming that a text consists of neutral words and a single positive word "love", a smoothed classifier g still can correctly label the masked examples as the original text with a high probability if the masking rate ρ is less than 50% because there is more than a 50% chance that the word "love" is not masked and the ensemble method is used to get a final prediction.

However, when a higher masking rate is used, the word "love" would be masked with the probability higher than 50%, which reduces the probability that the masked texts have the same label as the original text.

4.3 Results of the Empirical Robustness

As mentioned in the introduction, if a relatively small proportion of words are perturbed (relative to the masking rate ρ), then it is highly unlikely that all of the perturbed words can survive from the random masking operation. As the value of ρ decreases, we have to make sure that the following *risk probability* is greater than 0.5 for any text $\boldsymbol{x}$; otherwise more than 50% of masked copies of $\boldsymbol{x}$ will contain all the perturbed words, which tends to cause the base classifier f to make mistakes, and so does g.

$$\mathbb{P}(\boldsymbol{x}, \rho, \gamma) = \frac{\binom{(1-\gamma)h_{\boldsymbol{x}}}{(1-\gamma-\rho)h_{\boldsymbol{x}}}}{\binom{h_{\boldsymbol{x}}}{\rho h_{\boldsymbol{x}}}} \tag{10}$$

where γ is the maximum percentage of words that can be perturbed. For the AGNEWS dataset, this risk probability is very close to zero when setting $\rho = 90\%$ (chosen by the certified robustness experiment) no matter what the value of γ is applied. We use the average length of texts in a dataset (instead of each text) to estimate this risk probability. For the SST2 dataset, the risk probability is approximately equal to 0.5 when setting $\rho = 30\%$ (selected by the certified robustness experiment). To reduce this risk, we designed a new sampling strategy in which the probability of a word being masked corresponds to its output probability of a BERT-based LM. Generally, the higher the output probability of a word is, the lower the probability that this word has been perturbed. Intuitively, we want to retain the words that have not been perturbed as much as possible. Such an LM-based sampling was also used to evaluate the empirical robustness on SST2 instead of that based on the uniform distribution. Note that LM-based sampling is unnecessary when evaluating on AGNEWS, and the experimental results also show that there is no significant difference whether or not the LM-based sampling is used on this dataset.

In the following experiments, we consider two ensemble methods (Cheng et al. 2020a): *logits-summed* ensemble (logit) and *majority-vote* ensemble (vote). In the "logit" method, we take the average of the logits produced by the base classifier f over all the individual random samples as the final prediction. In the "vote" strategy, we simply count the votes for each class label. The following metrics (Li et al. 2021) are used to report the results of empirical robustness:

- The **clean accuracy** (Cln) is the classification accuracy achieved by a classifier on the clean texts.
- The **robust accuracy** (Boa) is the accuracy of a classifier achieved under a certain attack.
- The **success rate** (Succ) is the number of texts successfully perturbed by an attack algorithm (causing the model to make errors) divided by all the number of texts to be attempted.

We evaluated the empirical robustness under test-time attacks by using the TextAttack[2] framework (Morris et al. 2020) with three black-box, score-based attack algorithms: TextFooler (Jin et al. 2020), BERT-Attack (Li et al. 2020), and DeepWordBug (Gao et al. 2018). TextFooler and BERT-Attack adversarially perturb the text inputs by the word-level substitutions, whereas DeepWordBug performs the character-level perturbations to the input texts. TextFooler generates synonyms using 50 nearest neighbors of GloVe vectors (Pennington, Socher, and Manning 2014), while BERT-Attack uses the BERT to generate synonyms dynamically, meaning that no defenders can know in advance the synonyms used by BERT-Attack. DeepWordBug generates text adversarial examples by replacing, scrambling, and erasing a few characters of some words in the input texts.

We compared RanMASK with the following defense methods proposed recently:

- PGD-K (Madry et al. 2018): applies gradient-guided adversarial perturbations to word embeddings and minimizes the resultant adversarial loss inside different regions around input samples.
- FreeLB (Zhu et al. 2020): adds norm-bounded adversarial perturbations to the input's word embeddings using a gradient-based method, and enlarges the batch size with diversified adversarial samples under such norm constraints.
- Adv-Hotflip (Ebrahimi et al. 2018): first generates textual adversarial examples by using Hotflip (Ebrahimi et al. 2018) and then augments the generated examples with the original training data to train a robust model. Unlike PGD-K and FreeLB, Adv-Hotflip will generate real adversarial examples by replacing the original words with their synonyms rather than performing adversarial perturbations in the word embedding space.
- Adversarial Data Augmentation: it still is one of the most successful defense methods for NLP models (Miyato, Dai, and Goodfellow 2017; Sato et al. 2018). During the training phase, they replace a word with one of its synonyms that maximizes the prediction loss. By augmenting these adversarial examples with the original training data, the model is robust to such perturbations.

The results of the empirical robustness on AGNEWS dataset are reported in Table 4. From these reported numbers, we see that RanMASK-90% consistently performs better than the competitors under all the three attack algorithms on the robust accuracy while suffering little performance drop on the clean data. The empirical results on SST2 are reported in Table 5, and we found similar trends as those on AGNEWS, especially for those when the LM-based sampling strategy was used. On the IMDB dataset, we even observed that RanMASK achieved the highest accuracy on the clean data with 1.5% improvement compared to the baseline built upon RoBERTa, where the masking rate (i.e., 30%) was tuned on the validation set when the maximum MCB was achieved by the method introduced in Section 4.2. The results on three text classification datasets

2 https://github.com/QData/TextAttack.

Table 4
Empirical results on AGNEWS. RanMASK-90% with the "vote" ensemble method achieved the best results on the robust accuracy under all three different attack algorithms, indicating that RanMASK performs better in defense against both word substitution-based attacks and character-level perturbations.

Method	TextFooler			BERT-Attack			DeepWordBug		
	Cln%	Boa%	Succ%	Cln%	Boa%	Succ%	Cln%	Boa%	Succ%
Baseline (RoBERTa)	93.9	15.8	83.2	94.7	26.7	71.8	94.2	33.0	65.0
PGD-10 (Madry et al. 2018)	**95.0**	22.3	76.5	**95.3**	30.0	68.5	**94.9**	38.8	59.1
FreeLB (Zhu et al. 2020)	93.9	24.6	73.8	**95.3**	28.3	70.3	93.7	44.0	53.0
Adv-Hotflip (Ebrahimi et al. 2018)	93.4	21.3	77.2	93.9	26.8	71.5	94.6	37.6	60.3
Data Augmentation	93.3	23.7	74.6	92.3	39.1	57.6	93.8	49.7	47.0
RanMASK-90% (logit)	89.1	42.7	52.1	88.5	30.0	66.1	89.8	45.4	45.4
RanMASK-90% (vote)	91.2	**55.1**	**39.6**	89.1	**41.1**	**53.9**	90.3	**57.5**	**36.0**

Table 5
Empirical results on SST2. RanMASK-30% with both the LM-based sampling strategy and the "vote" ensemble method achieved the best results on the robust accuracy under all three different attacks.

Method	TextFooler			BERT-Attack			DeepWordBug		
	Cln%	Boa%	Succ%	Cln%	Boa%	Succ%	Cln%	Boa%	Succ%
Baseline (RoBERTa)	**94.3**	5.4	94.3	93.9	6.2	93.4	**94.7**	17.0	82.1
PGD-10 (Madry et al. 2018)	94.0	5.6	94.0	**94.4**	5.6	94.1	92.9	18.3	80.3
FreeLB (Zhu et al. 2020)	93.7	13.9	85.2	93.8	10.4	89.0	93.0	23.7	74.5
Adv-Hotflip (Ebrahimi et al. 2018)	**94.3**	12.3	87.0	93.8	11.4	87.9	93.3	23.4	74.9
Data Augmentation	91.0	9.6	89.5	88.2	16.9	80.8	91.8	23.5	74.4
RanMASK-30% (logit)	92.9	8.9	90.4	92.9	9.5	89.8	93.0	21.1	77.3
RanMASK-30% (vote)	92.7	12.9	86.1	93.0	11.4	87.7	92.7	27.5	70.3
RanMASK-30% (vote) + LM	90.6	**23.4**	**74.2**	90.4	**22.8**	**74.8**	91.0	**41.7**	**53.1**

show that our RanMASK consistently achieved better robust accuracy while suffering little loss on the original clean data.

Comparing to the baseline (RoBERTa), RanMASK can improve the accuracy under attack or the robust accuracy (Boa) by 21.06%, and lower the attack success rate (Succ) by 23.71% on average at the cost of a 2.07% decrease in the clean accuracy across three datasets and under three attack algorithms. When comparing RanMASK to a strong competitor, FreeLB (Zhu et al. 2020), which was proposed very recently, RanMASK still can further increase the accuracy under attack by 15.47%, and reduce the attack success rate by 17.71% on average at the cost of a 1.98% decrease in the clean accuracy under three different attacks. Generally, RanMASK with the "vote" ensemble performs better than that with the "logit" ensemble, except on the IMDB dataset under BERT-Attack and TextFooler attacks. We will thoroughly discuss the properties and behaviors of those two ensemble methods in the following sections.

As shown in Tables 4 and 6, any model applied to IMDB shows to be more vulnerable to adversarial attacks than the same one on AGNEWS. For example, BERT-Attack achieved a 100% attack success rate against the baseline model on IMDB

Table 6
Empirical results on IMDB. RanMASK-30% with the "vote" ensemble method achieved the best results on the robust accuracy under all three attack algorithms.

Method	TextFooler			BERT-Attack			DeepWordBug		
	Cln%	Boa%	Succ%	Cln%	Boa%	Succ%	Cln%	Boa%	Succ%
Baseline (RoBERTa)	91.5	0.5	99.4	91.5	0.0	100.0	91.5	48.5	47.0
PGD-10 (Madry et al. 2018)	92.0	1.0	98.9	92.0	0.5	99.4	92.0	44.5	51.6
FreeLB (Zhu et al. 2020)	92.0	3.5	96.2	92.0	2.5	97.3	92.0	52.5	42.9
Adv-Hotflip (Ebrahimi et al. 2018)	91.5	6.5	92.9	91.5	11.5	87.4	91.5	42.5	53.5
Data Augmentation	90.5	2.5	97.2	91.0	5.5	94.0	91.0	50.5	44.5
RanMASK-30% (logit)	**93.0**	**23.5**	**74.7**	93.0	**22.0**	**76.3**	**93.5**	62.0	33.7
RanMASK-30% (vote)	**93.0**	18.0	80.7	**93.5**	17.0	81.8	92.5	**66.0**	**28.7**

while its attack success rates are far below 100% on the other datasets. It is probably because the average length of the sentences in IMDB (255 words on average) is much longer than that in AGNEWS (43 words on average). Longer sentences allow the adversaries to apply more synonym substitution-based or character-level perturbations to the original examples. While RanMASK shows to be more resistant to adversarial attacks, it also can improve the clean accuracy on IMDB. One reasonable explanation is that the models rely too heavily on the non-robust features that are less relevant to the categories to be classified, and our random masking strategy disproportionately affects non-robust features, which thus hinders the model's reliance on them. Note that the sentences in IMDB are relatively long, and many words in any sentence might be irrelevant for the classification but would be inappropriately used by the models for the prediction.

We also conducted the experiments of natural language inference on the Stanford Natural Language Inference (SNLI) corpus (Bowman et al. 2015), which is a collection of 570,000 English sentence pairs (a premise and a hypothesis) manually labeled for balanced classification with three labels: entailment, contradiction, and neutral. What makes the natural language inference different from the text classification is that it needs to determine whether the directional relation holds whenever the truth of one text (i.e., hypothesis) follows from another text (i.e., premise). We implemented a baseline model based on RoBERTa for this task. The premise and hypothesis are encoded by running RoBERTa on the word embeddings to generate the sentence representations, which uses attention between the premise and hypothesis to compute richer representations of each word in both sentences, and then the concatenation of these encodings is fed to a two-layer feedforward network for the prediction. The baseline model was trained with cross-entropy loss, and their hyperparameters were tuned on the validation set.

The results of the empirical robustness on SNLI are reported in Table 7. The masking rate (i.e., 15%) was tuned for RanMASK on the validation set when the maximum MCB was achieved. From these numbers, a handful of trends are readily apparent. RanMASK using the "vote" ensemble achieved better empirical robustness than that using the "logit" ensemble again. Comparing to the baseline, RanMASK can improve the accuracy under attack or the robust accuracy (Boa) by 14.05%, and lower the attack success rate (Succ) by 16.30% on average at the cost of 3.43% decrease in the clean accuracy under three different attack algorithms. When FreeLB is used for comparison, RanMASK can further improve the robust accuracy (Boa) by 13.43%, and reduce the attack success rate (Succ) by 15.83% on average.

In conclusion, RanMASK can improve the robust accuracy under different attacks much further than existing defense methods on various tasks, including text

Table 7
Empirical results on SNLI. RanMASK-15% with the "vote" ensemble method achieved the best results on the robust accuracy under all three attack algorithms.

Method	TextFooler			BERT-Attack			DeepWordBug		
	Cln%	Boa%	Succ%	Cln%	Boa%	Succ%	Cln%	Boa%	Succ%
Baseline (RoBERTa)	91.0	3.9	95.7	91.0	0.6	99.3	91.0	4.2	95.4
PGD-10 (Madry et al. 2018)	**91.9**	4.7	95.0	**91.9**	1.0	98.9	**91.9**	4.8	94.8
FreeLB (Zhu et al. 2020)	91.2	4.3	95.3	91.2	0.4	99.6	91.2	5.3	94.1
Adv-Hotflip (Ebrahimi et al. 2018)	88.8	6.0	93.2	88.8	1.5	98.3	88.5	8.9	90.0
Data Augmentation	89.5	14.2	84.1	89.7	1.8	98.0	91.0	20.3	77.7
RanMASK-15% (logit)	89.4	10.8	87.9	89.8	1.2	98.7	89.7	6.8	92.4
RanMASK-15% (vote)	87.0	**21.5**	**74.7**	89.5	**5.8**	**93.5**	86.2	**23.0**	**73.3**

classification, sentiment analysis, and natural language inference. However, it is well-known that there is a tradeoff between clean accuracy and adversarial robustness. This tradeoff also has been observed on these tasks with all the considered models, including those trained with RanMASK. Specifically, there is a tradeoff between clean accuracy and maximum median certified robustness (MCB) for RanMASK. Generally, the higher the masking rate ρ, the greater the MCB and the lower the clean accuracy. In our experiments, the masking rate was chosen to achieve the highest MCB while maintaining the clean accuracy as much as possible for each dataset. In practice, the masking rate should be chosen to meet the requirements of specific applications and the preferences of their developers. It would be interesting to seek an efficient search method for finding a proper masking rate to balance clean accuracy and MCB. We leave this as future work.

4.4 Comparison with SAFER

Unlike other baselines, SAFER (Ye, Gong, and Liu 2020) is a certified defense method against adversarial attacks proposed for NLP models. Although the same evaluation metrics and attack algorithms are used as when comparing to other baselines, we thoroughly compare SAFER to our RanMASK in this separate subsection because both methods provide the certified robustness of neural text models. What else our method has in common with SAFER is that both methods were built on the ensemble strategy. We report in Table 8 the empirical robustness of RanMASK on AGNEWS compared with SAFER (Ye, Gong, and Liu 2020), a very recently proposed certified defense. From these reported numbers, we found that RanMASK outperforms SAFER under the setting where the "logit" ensemble is used for the predictions, while SAFER slightly performs better than RanMASK when the "vote" ensemble is used under the attack of TextFooler. However, this comparison is not direct and fair. First, SAFER makes use of the same synonym table used by TextFooler (i.e, they also assume that the defenders know in advance how the adversaries generate synonyms to launch adversarial attacks). Second, we found that different smoothing defense methods behave quite differently as the ensemble method is changed from the "vote" to the "logit."

Typical score-based attack algorithms, such as TextFooler and DeepWordBug, usually use two steps to craft adversarial examples: greedily identify the most vulnerable position to change; modify it slightly to maximize the model's prediction error. This two-step would be repeated iteratively until the model's prediction changes. If the

Table 8
Empirical results of RanMASK on AGNEWS compared with SAFER under three attack algorithms: TextFooler, BERT-Attack, and DeepWordBug.

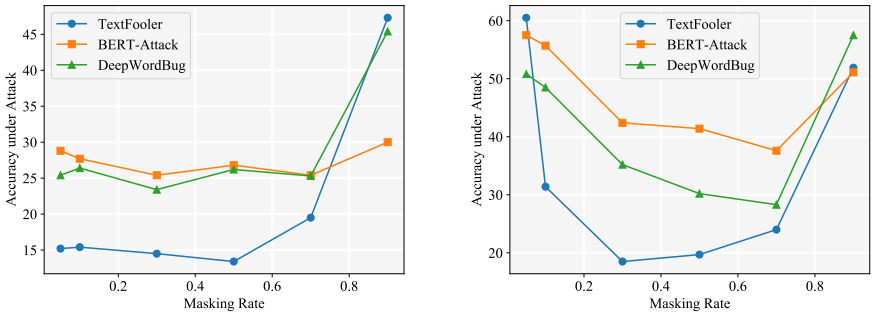
Method	TextFooler			BERT-Attack			DeepWordBug		
	Cln%	**Boa%**	**Succ%**	**Cln%**	**Boa%**	**Succ%**	**Cln%**	**Boa%**	**Succ%**
Baseline (BERT)	93.0	5.6	94.0	95.1	16.3	82.9	94.3	16.6	82.4
SAFER (logit)	94.6	26.1	72.4	94.8	29.0	69.4	95.1	31.9	66.5
SAFER (vote)	**95.4**	**78.6**	**17.6**	94.3	63.4	32.8	95.2	78.4	17.7
RanMASK-90% (logit)	91.3	47.3	48.2	91.6	38.3	58.2	89.2	39.6	55.6
RanMASK-90% (vote)	90.3	51.9	42.5	92.7	46.3	50.1	90.4	51.8	42.7
RanMASK-5% (logit)	94.4	13.2	86.0	**96.0**	25.6	73.3	94.8	23.4	75.3
RanMASK-5% (vote)	93.9	68.6	26.9	95.2	63.0	33.8	**95.3**	77.1	19.1
RanMASK-5% (vote) + LM	94.8	71.4	24.7	95.7	**65.3**	**31.8**	93.8	**80.3**	**14.4**

"vote" ensemble method is used, the class distributions produced by the models trained with SAFER would be quite sharp, even very close to one-hot categorical distribution,[3] which hinders the adversaries to peek into the changes in the model's predictions by a small perturbation on the input, ending up trapped in local minima. This forces the adversaries to launch the decision-based attacks (Maheshwary, Maheshwary, and Pudi 2020) instead of the score-based ones, which can dramatically affect the resultant attack success rates. If the "logit" ensemble method is used or the attack algorithm is designed to perturb more than one word at a time, the empirical robustness of SAFER will drop significantly. Therefore, it is unfair to compare "vote"-based ensemble defense methods with others when conducting empirical experiments. We believe these methods using the "vote" ensemble will greatly improve the model's defense performance when the models are deployed for real-world applications, but we recommend using the "logit" ensemble method if someone really wants to analyze and prove the effectiveness of the proposed defense methods against textual adversarial attacks in future research.

4.5 Impact of Masking Rate on Robust Accuracy

We want to understand the impact of different masking rates on the accuracy of RanMASK under adversarial attacks by varying the masking rates from 5% to 90%. We show in Figure 3 the average robust accuracy of RanMASK on the test set of AGNEWS versus several masking rates with two ensemble methods under three different attacks: TextFooler, BERT-Attack, and DeepWordBug. For each setting, the average accuracy is obtained over 5 runs with different random initialization. Taking the results obtained with TextFooler as an example (similar trends can be observed for other attack algorithms), when the "logit" ensemble method is used, the accuracy under attack generally increases until the best performance is achieved at the masking rate of 90% ($\rho = 90\%$)

3 In contrast to SAFER, the class distributions produced by the models trained with RanMASK are relatively smoother than those with SAFER. We estimated the average entropy of the distributions predicted by SAFER and RanMASK on 1,000 test samples selected randomly from the AGNEWS dataset. When TextFooler starts to attack, the average entropy of SAFER's predictions is 0.006, while those of RanMASK's are 0.025, 0.036, 0.102, and 0.587 when $\rho = 5\%$, 10%, 50%, and 90%, respectively. Note that the greater the entropy is, the smoother the distribution will be.



(a) Logits-summed Ensemble (b) Majority-vote Ensemble

Figure 3
Accuracy under attacks versus masking rates.

and there is a big jump from 70% to 90%. However, if the "vote" ensemble method is applied, we observed a dramatic decrease in the robust accuracy when the masking rates vary from 5% to 30%, and then the robust accuracy climbs up smoothly until getting to its peak when setting $\rho = 90\%$. Although it seems counterintuitive that the robust accuracy falls greatly at first and rises later, this phenomenon can be explained with the same reason we used to explain the difference between SAFER and RanMASK in their predictive behaviors in Section 4.4.

Note that the lower the masking rates, the more similar the class distributions RanMASK will produce for different masked copies of a given input text. Those distributions will be extremely similar when the "vote" ensemble method is used, which prevents the attack algorithms from effectively identifying the weakness of victim models by applying small perturbations to the input and tracking the resulting impacts. This misleading phenomenon deserves particular attention since it might hinder us to search for the optimal setting. For example, as we can see from Figure 3b, RanMASK-5% with the "vote" ensemble method achieved the remarkable robust accuracy on AGNEWS under three different attacks, but when the method is switched to the "logit" ensemble, RanMASK-5% performs much worse. No matter which ensemble method is used, the optimal masking rate should be set to around 90%, which can provide the certified robustness to any perturbation on the inputs by no more than 5% of words. As shown in Table 2, we can certify the classifications of over 50% of texts to be robust to any perturbation of 5 words on AGNEWS when $\rho = 90\%$, while the number of words allowed to be perturbed is very close to 0 when $\rho = 5\%$. Therefore, if the "vote" ensemble method was used, we may come to wrong conclusions.

4.6 Impact of Sentence Length on Accuracy

To determine whether the smoothed classifier g is certified robust at a given $\boldsymbol{x}$, the condition, $\underline{p_c}(\boldsymbol{x}) - \beta\Delta > 0.5$ (see Corollary 1), needs to be evaluated to check if it is satisfied. It would be hard or even impossible to make this condition satisfied when the given sentence is very short. For an extreme case of two-word sentences, the maximum number of words that can be masked is 1, and such a condition cannot be satisfied no matter what value of d since the accuracy is always equal to or less than 100% (d can only take the value 1 or 2). However, when the length of sentence $\boldsymbol{x}$ is 3, the condition

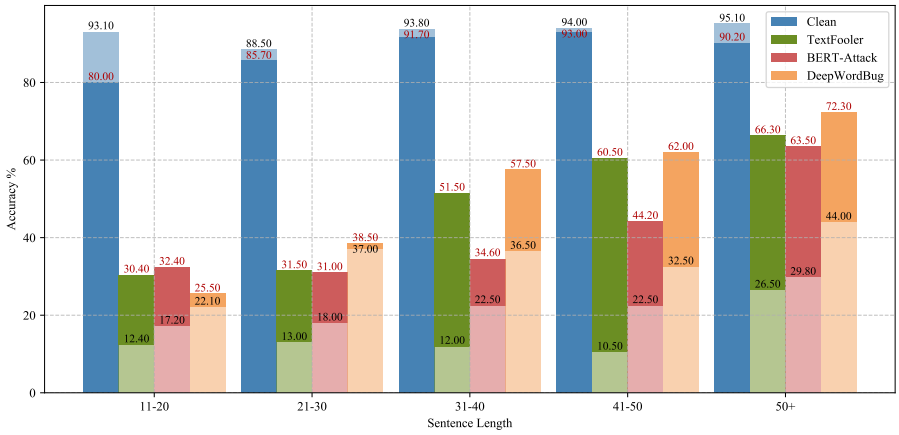


Figure 4
The impact of sentence length on clean accuracy and robust accuracy under three different attack algorithms (TextFoolor, BERT-Attack, and DeepWordBug) on the test set of AGNEWS. The RanMASK with the masking rate $\rho = 90\%$ was used to evaluate the performance as the model reported in Table 4. The accuracies achieved by RanMASK are shown in red font while those yielded by baseline model (RoBERTa) are reported in black font.

could be satisfied if $d = 1$ (33% words are allowed to be perturbed), $k_{\boldsymbol{x}} = 1$ (only one word is not masked), and $\underline{p_c(\boldsymbol{x})} > 0.83$. As discussed in Section 3.2, $p_c(\boldsymbol{x})$ is used to approximate the value of β. When the length of sentence $\boldsymbol{x}$ is equal to 5, the condition has a higher chance of being satisfied. Assuming that $d = 1$ (20% of words can be perturbed by adversaries), the certified condition would be satisfied if $k_{\boldsymbol{x}} = 1$ and $\underline{p_c(\boldsymbol{x})} > 0.62$ or if $k_{\boldsymbol{x}} = 2$ and $\underline{p_c(\boldsymbol{x})} > 0.83$. When $d = 2$ (40% of words can be perturbed), the condition still could be satisfied if $k_{\boldsymbol{x}} = 1$ and $\underline{p_c(\boldsymbol{x})} > 0.83$. In real-world applications, most sentences are longer than 5 words. According to our statistics, there are no sentences shorter than 5 words in the test sets of AGNEWS (Zhang, Zhao, and LeCun 2015) and SST2 (Socher et al. 2013), and all the sentences in AGNEWS test set have more than 10 words.

In theory, the length of a sentence does affect how likely the condition of Corollary 1 can be satisfied. This effect decreases rapidly as the length of sentence increases, and the effect would be negligible when the sentence length is greater than 10 words because the differences in the value of Δ are less than 0.04 if $d \geq 3$ and $k_{\boldsymbol{x}} \geq 3$ for any pairs of these sentences with different lengths. We also want to know how sentence length empirically impacts clean and robust accuracies by varying the length of the sentences in the test sets of AGNEWS and SST2. In Figure 4, we show the clean and robust accuracies achieved by RanMASK-90% and the RoBERTa baseline for the different subsets of AGNEWS test dataset. The sentences in the test set were grouped into five different subsets (i.e., 11–20, 21–30, 31–40, 41–50, and 50+ words) according to their length. The numbers of the sentences in different subsets are 199 (2.6%), 1,072 (14.0%), 3,142 (41.1%), 2,420 (31.6%), and 821 (10.7%), respectively. As the setting described in Section 4.3, three different attack algorithms of TextFoolor, BERT-Attack, and DeepWordBug were used to evaluate the robustness of the two models. The accuracies achieved by RanMASK-90% are shown in red font while those produced by the baseline model (RoBERTa) are reported in black font. In Figure 5, we show the results on the test set of SST2 where the sentences were divided into five different subsets of 1–10, 11–20, 21–30, 31–40, and 41+ words (note that the average length of the sentences in AGNEWS is longer than that in SST2). The sizes

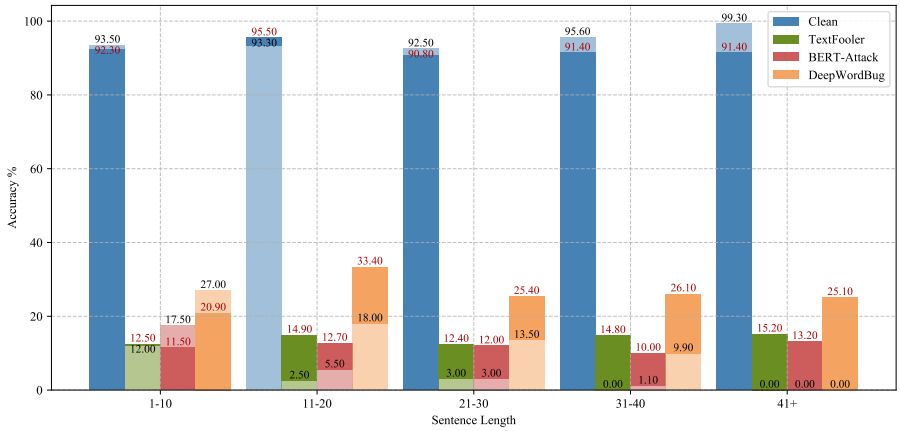


Figure 5
The impact of sentence length on clean accuracy and robust accuracy under three different attack algorithms (TextFoolor, BERT-Attack, and DeepWordBug) on the test set of SST2. The RanMASK with the masking rate $\rho = 30\%$ was used to evaluate the performance as the model reported in Table 5. The accuracies achieved by RanMASK are shown in red font. In comparison, the accuracies obtained by baseline (RoBERTa) are reported in black font.

of different SST2 subsets are 351 (19.3%), 702 (38.5%), 560 (30.8%), 181 (9.9%), and 27 (1.5%), respectively.

As we can see from Figures 4 and 5, RanMASK achieved comparable performance to the RoBERTa baseline in clean accuracy, and the changing trend of clean accuracy across different sentence lengths is generally consistent with that produced by the baseline with very few exceptions. RanMASK clearly performs substantially better than the baseline built upon RoBERTa in robust accuracy across different sentence lengths on both AGNEWS and SST2 datasets. The baseline only outperforms RanMASK-30% on one subset of SST2 with 1–10 lengths under the attacks of BERT-Attack and DeepWordBug. As mentioned in Section 4.3, the risk probability is approximately equal to 0.5 when setting $\rho = 30\%$, and the LM-based sampling should be used to reduce the risk when evaluating on SST2. If the LM-based sampling strategy was used, RanMASK-30% can outperform the baseline by 2% and 16% increases in accuracy on the same subset of SST2 under the attack algorithms of BERT-Attack and DeepWordBug, respectively. The numbers presented in Figure 4 show that the longer the sentence, the better the robust accuracy of RanMASK, while such a trend is not observed on SST2 dataset (see Figure 5). The robust accuracy achieved by RanMASK fluctuates within a relatively small range when the length of the sentence increases from 1-10 to 41+ on SST2 test set. The reason for this difference between the two datasets is that SST2 was constructed for sentiment analysis, and the sentiment of a sentence largely depends on whether a few specific sentiment words occur in the sentence, and the number of sentiment words does not increase significantly with the length of the sentence. Unlike SST2, AGNEWS was created for news topic classification and the number of topic words generally increases with sentence length, which makes it easier for RanMASK to correctly predict the category of a longer sentence because there is a great chance for some topic words to survive from the masking operations even though a large portion of words have been masked. The adversarial robustness of RanMASK could be affected by sentence length, but this effect exerts in a more complex way and depends on the type of task.

5. Related Work

Even though achieving prominent performance on many important tasks, it has been reported that neural networks are vulnerable to adversarial examples—inputs generated by applying imperceptible but intentionally perturbations to them, such that the perturbed inputs can cause the model to make mistakes. Adversarial examples were first discovered in the image domain (Szegedy et al. 2014), and then their existence and pervasiveness were also observed in the text domain. Despite the fact that generating adversarial examples for texts has proven to be a more challenging task than for images due to their discrete nature, many methods have been proposed to generate textual adversarial examples and reveal the vulnerability of deep neural networks for NLP tasks, including reading comprehension (Jia and Liang 2017), text classification (Samanta and Mehta 2017; Wong 2017; Liang et al. 2018; Alzantot et al. 2018), machine translation (Zhao, Dua, and Singh 2018; Ebrahimi et al. 2018; Cheng et al. 2020b), dialogue systems (Cheng, Wei, and Hsieh 2019), and syntactic parsing (Zheng et al. 2020). The existence and pervasiveness of adversarial examples pose serious threats to neural networks-based models, especially when applying them to security-critical applications, such as face recognition systems (Zhu, Lu, and Chiang 2019), autonomous driving systems (Eykholt et al. 2018), and toxic content detection (Li et al. 2019). The introduction of adversarial examples and training ushered in a new era to understand and improve the machine learning models and has received significant attention recently (Goodfellow, Shlens, and Szegedy 2015a; Moosavi-Dezfooli, Fawzi, and Frossard 2016; Madry et al. 2018; Ilyas et al. 2019; Cohen, Rosenfeld, and Kolter 2019; Lécuyer et al. 2019; Yuan et al. 2021). Adversarial examples yield broader insights into the targeted models by exposing them to such intentionally crafted examples. In the following, we briefly review related work in both text adversarial attacks and defenses.

5.1 Text Adversarial Attacks

Text adversarial attack methods generate adversarial examples by perturbing original texts to maximize the model's prediction errors while maintaining the adversarial examples' fluency and naturalness. The recently proposed methods attack text examples mainly by replacing, scrambling, and erasing characters (Gao et al. 2018; Ebrahimi et al. 2018) or words (Alzantot et al. 2018; Ren et al. 2019a; Zheng et al. 2020; Jin et al. 2020; Li et al. 2020) under semantics- or/and syntax-preserving constraints based on the cosine similarity (Li et al. 2019; Jin et al. 2020), edit distance (Gao et al. 2018), or syntactic structural similarity (Zheng et al. 2020; Han et al. 2020).

Depending on the degree of access to the target (or victim) model, adversarial examples can be crafted in two different settings: *white-box* and *black-box* settings (Xu et al. 2019; Wang et al. 2019). In the white-box setting, an adversary can access the model's architecture, parameters, and input feature representations that are not accessible in the black-box setting. The white-box attacks normally yield a higher success rate because the knowledge of target models can be leveraged to guide the generation of adversarial examples. However, the black-box attacks do not require access to target models, making them more practicable for many real-world attacks. Adversarial attacks also can be divided into *targeted* and *non-targeted* ones, depending on the purpose of the adversary. Taking the classification task as an example, the output category of a generated example is intentionally controlled to a specific category in a targeted attack while a non-targeted attack does not care about the category of misclassified results.

For text data, input sentences can be manipulated at character (Ebrahimi et al. 2018), sememe (the minimum semantic units) (Zang et al. 2020), or word (Samanta and Mehta 2017; Alzantot et al. 2018) levels by replacement, alteration (e.g., deliberately introducing typos or misspellings), swap, insertion, erasure, or directly making small perturbations to their feature embeddings. Generally, they want to ensure that the crafted adversarial examples are sufficiently similar to their original ones, and these modifications should be made within semantics-preserving constraints. Such semantic similarity constraints were usually defined based on the cosine similarity (Wong 2017; Barham and Feizi 2019; Jin et al. 2020; Ribeiro, Singh, and Guestrin 2018) or edit distance (Gao et al. 2018). Zheng et al. (2020) showed that adversarial examples also exist in syntactic parsing, and they crafted the adversarial examples that preserve the same syntactic structures as the original sentences by imposing the constraints based on syntactic structural similarity.

Text adversarial example generation usually involves two steps: determine an important position (or token) to change; and modify it slightly to maximize the model's prediction error. This two-step recipe can be repeated iteratively until the model's prediction changes or certain stopping criteria are reached. Many methods have been proposed to determine the important positions by random selection (Alzantot et al. 2018), trial-and-error testing at each possible point (Kuleshov et al. 2018), analyzing the effects on the model of masking various parts of an input text (Samanta and Mehta 2017; Gao et al. 2018; Jin et al. 2020; Yang et al. 2018), comparing their attention scores (Hsieh et al. 2019), or gradient-guided optimization methods (Ebrahimi et al. 2018; Lei et al. 2019; Wallace et al. 2019; Barham and Feizi 2019).

After the important positions are identified, the most popular way to alter text examples is to replace the characters or words at selected positions with similar substitutes. Such substitutes can be chosen from the nearest neighbors in an embedding space (Alzantot et al. 2018; Kuleshov et al. 2018; Jin et al. 2020; Barham and Feizi 2019), synonyms in a prepared dictionary (Samanta and Mehta 2017; Hsieh et al. 2019), visually similar alternatives like typos (Samanta and Mehta 2017; Ebrahimi et al. 2018; Liang et al. 2018), Internet slang and trademark logos (Eger et al. 2019), paraphrases (Lei et al. 2019), or even randomly selected ones (Gao et al. 2018). Given an input text, Zhao, Dua, and Singh (2018) proposed to search for adversaries in the neighborhood of its corresponding representation in latent space by sampling within a range that is recursively tightened. In order to mislead a reading comprehension system, Jia and Liang (2017) tried to insert a few distraction sentences generated by a simple set of rules into text examples. It is worth noting that the certified robustness still can be achieved in theory by our RanMASK under insertion attacks if the constraint of $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq d$ is satisfied because we do not assume that the defenders know how the adversaries choose the words to replace original ones. When calculating this 0-norm of the change, a special token (say, `[BLANK]`) should be inserted into $\boldsymbol{x}$ for each position where a word is inserted in $\boldsymbol{x}'$. Unfortunately, RanMASK cannot be used to provide the certified robustness against deletion attacks because we cannot know which words and how many of them were deleted from the original sentences.

5.2 Text Adversarial Defenses

The goal of adversarial defenses is to learn a model capable of achieving high test-time accuracy on both clean and adversarial examples. Recently, many defense methods have been proposed to defend against text adversarial attacks, which can roughly be divided

into two categories: *empirical* (Miyato, Dai, and Goodfellow 2017; Sato et al. 2018; Zhou et al. 2021; Dong et al. 2021) and *certified* (Jia et al. 2019; Huang et al. 2019; Ye, Gong, and Liu 2020) methods.

Adversarial data augmentation is one of the most effective empirical defenses (Ren et al. 2019b; Jin et al. 2020; Li et al. 2020) for NLP models. During the training time, they replace a word with one of its synonyms to create adversarial examples. By augmenting these adversarial examples with the original training data, the model is robust to such perturbations. However, the number of possible perturbations scales exponentially with the length of texts, so data augmentation cannot cover all the perturbations of any input text. Zhou et al. (2021) use a convex hull formed by a word and its synonyms to capture word substitution-based perturbations, and they guarantee with high probability that the model is robust at any point within the convex hull. The similar technique also has been used by Dong et al. (2021). During the training phase, they allow the models to search for the worse-case over the convex hull (i.e., a set of synonyms) and minimize the error with the worst-case. Zhou et al. (2021) also showed that their framework can be extended to higher-order neighbors (synonyms) to boost the model's robustness further.

Adversarial training (Miyato, Dai, and Goodfellow 2017; Zhu et al. 2020) is another one of the most successful empirical defense methods by adding norm-bounded adversarial perturbations to word embeddings and minimizing the resultant adversarial loss. A family of fast-gradient sign methods was introduced by Goodfellow, Shlens, and Szegedy (2015b) to generate adversarial examples in the image domain. They showed that the robustness and generalization of machine learning models could be improved by including high-quality adversaries in the training data. Miyato, Dai, and Goodfellow (2017) applied a fast-gradient sign method–like adversarial training method to the text domain by adding perturbations to word embeddings rather than to discrete text inputs. In order to improve the interpretability of adversarial training, Sato et al. (2018) extended the work of Miyato, Dai, and Goodfellow (2017) by constraining the directions of perturbations toward the existing words in the word embedding space. Zhang and Yang (2018) applied several types of noise to perturb the input word embeddings, including Gaussian, Bernoulli, and adversarial noises, to mitigate the overfitting problem of NLP models. Recently, Zhu et al. (2020) proposed a novel adversarial training algorithm, called FreeLB, which adds adversarial perturbations to word embeddings and then minimizes the resultant adversarial loss inside different regions around input samples. They add norm-bounded adversarial perturbations to the input sentences' embeddings using a gradient-based method and enlarge the batch size with diversified adversarial samples under such norm constraints. The studies in this line of research focus on the generalization of models rather than their robustness against adversarial attacks.

Although the above empirical methods can successfully defend against the adversarial examples generated by the algorithms used during the training phase, the downside of such methods is that failure to discover an adversarial example does not mean that another more sophisticated attack could not find one. Recently, a set of certified defenses has been introduced, which guarantees the robustness to some specific types of attacks. For example, Jia et al. (2019) and Huang et al. (2019) use a bounding technique, Interval Bound Propagation (Gowal et al. 2018; Dvijotham et al. 2018), to formally verify a model's robustness against word substitution-based perturbations. Shi et al. (2020) and Xu et al. (2020) proposed the robustness verification and training method for transformers (Vaswani et al. 2017) based on linear relaxation-based perturbation analysis. However, these defenses often lead to loose upper bounds for arbitrary networks and result in a higher cost of clean accuracy. Furthermore, due to the difficulty of verification,

existing certified defense methods are usually not to scale and remain hard to scale to large networks, such as BERT and RoBERTa. To achieve certified robustness on large architectures, Ye, Gong, and Liu (2020) proposed a certified robust method, called SAFER, which is structure-free and can be applied to arbitrary large architectures. However, the base classifier of SAFER needs to be trained by adversarial data augmentation, and randomly perturbing a word to its synonyms performs poorly in practice. Mathias et al. (2019) also proposed a certified defense, called PixelDP, which is based on a novel connection between adversarial robustness and differential privacy. They achieve the certified robustness by introducing a noise layer in networks and making the expected output of those networks to have bounded sensitivity to p-norm changes in the input. However, they require that $p > 0$ and, therefore, their robustness condition cannot be generalized to 0-norm bounded adversarial examples (e.g., word substitution–based threat). Their certified defense was mainly proposed to defend against the adversarial perturbations measured by the p-norm of the change, when $p = 1$ or $p = 2$ (theoretically works for $p = \infty$).

The major problem is that all the existing certified defense methods make an unrealistic assumption that the defenders can access the synonyms used by the adversaries. They could be broken by more sophisticated attacks by using synonym sets with large size (Jin et al. 2020), generating synonyms dynamically with BERT (Li et al. 2020), or perturbing the inputs at the character level (Gao et al. 2018; Li et al. 2019). In this study, we show that random smoothing can be integrated with random masking strategy to boost the robust accuracy and such an integration leads to a certified defense method. In contrast to existing certified robust methods, the above unrealistic assumption is no longer required. Furthermore, the NLP models trained by our defense method can defend against both the word substitution-based attacks and character-level perturbations.

This study is most related to the work of Levine and Feizi (2020) that was developed to defend against sparse adversarial attacks in the image domain. The theoretical contribution of our study beyond theirs (Levine and Feizi 2020) is that we introduce a key variable β that is associated with each pair of an input text $\boldsymbol{x}$ and its adversarial example $\boldsymbol{x}'$, and the introduction of β yields a tighter certificate bound. The value of β is defined to be the conditional probability that the base classifier f will label the masked copies of $\boldsymbol{x}$ with the class c where the indices of unmasked words are overlapped with $\boldsymbol{x} \ominus \boldsymbol{x}'$ (i.e., the set of word indices at which $\boldsymbol{x}$ and $\boldsymbol{x}'$ differ). For estimating the value of β, we also present a Monte Carlo–based algorithm to evaluate β. On the MNIST dataset (Deng 2012), Levine and Feizi (2020) can certify the classifications of over 50% of images to be robust to any perturbations of at most only 8 pixels. The number of pixels in each image of the MNIST dataset is 784 (28×28), which means that to provide the certified robustness just 1.02% of pixels can be perturbed. By contrast, we can certify the classifications of over 50% of texts to be robust to any perturbations of 5 words on AGNEWS (Zhang, Zhao, and LeCun 2015). The average length of sentences in the AGNEWS dataset is about 43, which means that 11.62% of words can be maliciously manipulated while the certified robustness is still guaranteed due to the tighter certificate bound provided by Equation (7). Note that the changes by a very few pixels (say, 8 pixels, 1.02%) could be negligible and might even be left unnoticed but replacing with some words (say, 5 words, 11.62%) would significantly change the way to express the meaning. In addition to this theoretical contribution, we proposed a new sampling strategy in which the probability of a word being masked corresponds to its output probability of a BERT-based language model to reduce the risk probability, defined as Equation (10). The experimental results show that this LM-based sampling achieved better robust accuracy while suffering little loss on the clean data.

6. Conclusions

In this study, we propose a smoothing-based certified defense method for NLP models to substantially improve the robust accuracy against different threat models, including synonym substitution-based transformations and character-level perturbations. The main advantage of our method is that we do not base the certified robustness on the unrealistic assumption that the defenders know how the adversaries generate synonyms. This method is broadly applicable, generic, and scalable, and it can be incorporated with little effort in any neural networks, and scales to large architectures, such as BERT. We demonstrated through extensive experimentation that our smoothed classifiers perform better than existing empirical and certified defenses across different datasets.

It would be interesting to see the results of combining RanMASK with other defense methods such as FreeLB (Zhu et al. 2020) and Dirichlet Neighborhood Ensemble (Zhou et al. 2021) because they are orthogonal to each other. To the best of our knowledge, there is no method that can boost both clean and robust accuracy, and the trade-off has been proved empirically. We suggest a defense framework that first uses a pre-trained detector to determine whether an input text is an adversarial example. If it is classified as an adversarial example, it should be fed to a text classifier trained with a certain defense method; otherwise, it will be input to a normally trained classifier for the prediction. Although the masking rate used in RanMASK should be chosen to meet the requirements of specific applications and the preferences of their developers, it would be useful to seek an efficient search method for finding a proper masking rate to balance the clean accuracy and adversarial robustness. We leave these three possible improvements and extensions as future work.

Appendix A. Proof of Theorem 1

Theorem 1
Given a text $\boldsymbol{x}$ and it adversarial example $\boldsymbol{x}'$, $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq d$ and $\kappa(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon$, for any class $c \in \mathcal{Y}$, we have:

$$p_c(\boldsymbol{x}) - p_c(\boldsymbol{x}') \leq \beta\Delta \tag{A.11}$$

where

$$\begin{aligned} \Delta &= 1 - \frac{\binom{h_{\boldsymbol{x}}-d}{k_{\boldsymbol{x}}}}{\binom{h_{\boldsymbol{x}}}{k_{\boldsymbol{x}}}}, \\ \beta &= \mathbb{P}(f(\mathcal{M}(\boldsymbol{x}, \mathcal{H})) = c \mid \mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing). \end{aligned} \tag{A.12}$$

Proof. Recall that $\mathcal{H}$ is the set of indices uniformly sampled from $\mathcal{I}(h_{\boldsymbol{x}}, k_{\boldsymbol{x}})$, i.e., $\mathcal{H} \sim \mathcal{U}(h_{\boldsymbol{x}}, k_{\boldsymbol{x}})$, we have:

$$\begin{aligned} p_c(\boldsymbol{x}) &= \mathbb{P}(f(\mathcal{M}(\boldsymbol{x}, \mathcal{H})) = c), \\ p_c(\boldsymbol{x}') &= \mathbb{P}(f(\mathcal{M}(\boldsymbol{x}', \mathcal{H})) = c). \end{aligned} \tag{A.13}$$

By the law of total probability, we obtain:

$$\begin{aligned} p_c(\boldsymbol{x}) = & \\ \mathbb{P}([f(\mathcal{M}(\boldsymbol{x},\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing]) & \\ + \mathbb{P}([f(\mathcal{M}(\boldsymbol{x},\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing]) & \\ p_c(\boldsymbol{x}') = & \\ \mathbb{P}([f(\mathcal{M}(\boldsymbol{x}',\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing]) & \\ + \mathbb{P}([f(\mathcal{M}(\boldsymbol{x}',\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing]). & \end{aligned} \tag{A.14}$$

Note that if $\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing$, then $\boldsymbol{x}$ and $\boldsymbol{x}'$ are identical at all the indices in $\mathcal{H}$ after random masking. In this case, we have $\mathcal{M}(\boldsymbol{x},\mathcal{H}) = \mathcal{M}(\boldsymbol{x}',\mathcal{H})$, which implies:

$$\begin{aligned} \mathbb{P}(f(\mathcal{M}(\boldsymbol{x},\mathcal{H})) = c \mid \mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing) = & \\ \mathbb{P}(f(\mathcal{M}(\boldsymbol{x}',\mathcal{H})) = c \mid \mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing) & \end{aligned} \tag{A.15}$$

Multiplying both sides of Equation (A.15) by $\mathbb{P}(\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing)$, we have:

$$\begin{aligned} \mathbb{P}([f(\mathcal{M}(\boldsymbol{x},\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing]) = & \\ \mathbb{P}([f(\mathcal{M}(\boldsymbol{x}',\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing]) & \end{aligned} \tag{A.16}$$

By Equations (A.14) and (A.16), and the non-negativity of probability, subtracting $p_c(\boldsymbol{x}')$ from $p_c(\boldsymbol{x})$ yields:

$$\begin{aligned} p_c(\boldsymbol{x}) - p_c(\boldsymbol{x}') = & \\ \mathbb{P}([f(\mathcal{M}(\boldsymbol{x},\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing]) - & \\ \mathbb{P}([f(\mathcal{M}(\boldsymbol{x}',\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing]) & \\ \leq \mathbb{P}([f(\mathcal{M}(\boldsymbol{x},\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing]) & \end{aligned} \tag{A.17}$$

By the definition of β, we have

$$\begin{aligned} \mathbb{P}([f(\mathcal{M}(\boldsymbol{x},\mathcal{H})) = c] \wedge [\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing]) = & \\ \beta \times \mathbb{P}(\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing) & \end{aligned} \tag{A.18}$$

Substituting Equation (A.18) into Equation (A.17) gives:

$$p_c(\boldsymbol{x}) - p_c(\boldsymbol{x}') \leq \beta \times \mathbb{P}(\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing) \tag{A.19}$$

Note that

$$\begin{aligned}\mathbb{P}(\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing) &= \frac{\binom{h_{\boldsymbol{x}} - |\boldsymbol{x} \ominus \boldsymbol{x}'|}{k_{\boldsymbol{x}}}}{\binom{h_{\boldsymbol{x}}}{k_{\boldsymbol{x}}}} \\ &= \frac{\binom{h_{\boldsymbol{x}} - \|\boldsymbol{x} - \boldsymbol{x}'\|_0}{k_{\boldsymbol{x}}}}{\binom{h_{\boldsymbol{x}}}{k_{\boldsymbol{x}}}}\end{aligned} \tag{A.20}$$

where the last equality follows since $\mathcal{H}$ is a uniform choice of $k_{\boldsymbol{x}}$ elements from $h_{\boldsymbol{x}}$: there are $\binom{h_{\boldsymbol{x}}}{k_{\boldsymbol{x}}}$ total ways to make this selection. Among all these selections, there are $\binom{h_{\boldsymbol{x}} - |\boldsymbol{x} \ominus \boldsymbol{x}'|}{k_{\boldsymbol{x}}}$ of which do not contain any element from $\boldsymbol{x} \ominus \boldsymbol{x}'$.

By the constraint of $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq d$, we obtain:

$$\begin{aligned}\mathbb{P}(\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') \neq \varnothing) &= 1 - \mathbb{P}(\mathcal{H} \cap (\boldsymbol{x} \ominus \boldsymbol{x}') = \varnothing) \\ &= 1 - \frac{\binom{h_{\boldsymbol{x}} - \|\boldsymbol{x} - \boldsymbol{x}'\|_0}{k_{\boldsymbol{x}}}}{\binom{h_{\boldsymbol{x}}}{k_{\boldsymbol{x}}}} \\ &\leq 1 - \frac{\binom{h_{\boldsymbol{x}} - d}{k_{\boldsymbol{x}}}}{\binom{h_{\boldsymbol{x}}}{k_{\boldsymbol{x}}}} = \Delta\end{aligned} \tag{A.21}$$

Combining inequalities (A.19) and (A.21) gives the statement of Theorem 1. □

Note that it is unnecessary to consider the case of $p_c(\boldsymbol{x}') > p_c(\boldsymbol{x})$ because if $p_c(\boldsymbol{x}') \geq p_c(\boldsymbol{x})$, the inequality of $p_c(\boldsymbol{x}) - p_c(\boldsymbol{x}') \leq \beta\Delta$ must hold since β and Δ are always positive by their definitions.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by National Natural Science Foundation of China (No. 62076068), Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0103), and Laboratory of Pinghu (Beijing Institute of Infinite Electric Measurement), Pinghu, China (No. 20220521).

References

Alzantot, Moustafa, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2890–2896. https://doi.org/10.18653/v1/D18-1316

Barham, Samuel and Soheil Feizi. 2019. Interpretable adversarial training for text. *arXiv preprint arXiv:1905.12864*.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642. https://doi.org/10.18653/v1/D15-1075

Cheng, Minhao, Cho-Jui Hsieh, Inderjit Dhillon, et al. 2020a. Voting based ensemble improves robustness of defensive models. *arXiv preprint arXiv:2011.14031*.

Cheng, Minhao, Wei Wei, and Cho-Jui Hsieh. 2019. Evaluating and enhancing the robustness of dialogue systems: A case study on a negotiation agent. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 3325–3335. https://doi.org/10.18653/v1/N19-1336

Cheng, Minhao, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020b.

Seq2Sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 3601–3608. https://doi.org/10.1609/aaai.v34i04.5767

Clopper, Charles J. and Egon S. Pearson. 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413. https://doi.org/10.1093/biomet/26.4.404

Cohen, Jeremy M., Elan Rosenfeld, and J. Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320.

Deng, Li. 2012. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142. https://doi.org/10.1109/MSP.2012.2211477

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

Dong, Xinshuai, Hong Liu, Rongrong Ji, and Anh Tuan Luu. 2021. Towards robustness against natural language word substitutions. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dvijotham, Krishnamurthy, Sven Gowal, Robert Stanforth, Relja Arandjelović, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. 2018. Training verified learners with learned verifiers. *arXiv preprint arXiv:* 1805.10265.

Ebrahimi, Javid, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 31–36. https://doi.org/10.18653/v1/P18-2006

Eger, Steffen, Gozde Gul Sahin, Andreas Rucklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1634–1647. https://doi.org/10.18653/v1/N19-1165

Eykholt, Kevin, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1625–1634. https://doi.org/10.1109/CVPR.2018.00175

Gao, Ji, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *Proceedings of the IEEE Security and Privacy Workshops (SPW)*, pages 50–56. https://doi.org/10.1109/SPW.2018.00016

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. 2015a. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. 2015b. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Gowal, Sven, Krishnamurthy (Dj) Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelović Timothy, Timothy Mann, and Pushmeet Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:* 1810.12715.

Han, Wenjuan, Liwen Zhang, Yong Jiang, and Kewei Tu. 2020. Adversarial attack and defense of structured prediction models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2327–2338. https://doi.org/10.18653/v1/2020.emnlp-main.182

Hsieh, Yu Lun, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1520–1529. https://doi.org/10.18653/v1/P19-1147

Huang, Po Sen, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven

Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4083–4093. https://doi.org/10.18653/v1/D19-1419

Ilyas, Andrew, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *Proceedings of Advances in Neural Information Processing Systems (NuerIPS)*, pages 125–136.

Jia, Robin and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2021–2031. https://doi.org/10.18653/v1/D17-1215

Jia, Robin, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142. https://doi.org/10.18653/v1/D19-1423, PubMed: 31773974

Jin, Di, 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 (AAAI)*, 34(05):8018–8025. https://doi.org/10.1609/aaai.v34i05.6311

Kuleshov, Volodymyr, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems. https://openreview.net/forum?id=r1QZ3zbAZ

Lécuyer, Mathias, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2019. Certified robustness to adversarial examples with differential privacy. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pages 656–672. https://doi.org/10.1109/SP.2019.00044

Lei, Qi, Lingfei Wu, Pin-Yu Chen, Alex Dimakis, Inderjit S. Dhillon, and Michael J. Witbrock. 2019. Discrete adversarial attacks and submodular optimization with applications to text classification. In *Proceedings of Machine Learning and Systems (MLSys)*, mlsys.org/Conferences/2019/

Levine, Alexander and Soheil Feizi. 2020. Robustness certificates for sparse adversarial attacks by randomized ablation. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 4585–4593. https://doi.org/10.1609/aaai.v34i04.5888

Li, Jinfeng, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating adversarial text against real-world applications. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2019.23138

Li, Linyang, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202. https://doi.org/10.18653/v1/2020.emnlp-main.500

Li, Zongyi, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3137–3147. https://doi.org/10.18653/v1/2021.emnlp-main.251

Liang, Bin, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4208–4215. https://doi.org/10.24963/ijcai.2018/585

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

Loshchilov, Ilya and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Loshchilov, Ilya and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 142–150.

Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Maheshwary, Rishabh, Saket Maheshwary, and Vikram Pudi. 2020. Generating natural language attacks in a hard label black box setting. *arXiv preprint arXiv:2012.14956*. https://doi.org/10.1609/aaai.v35i15.17595

Mathias, Lecuyer, Atlidakis Vaggelis, Geambasu Roxana, Hsu Daniel, and Jana Suman. 2019. Certified robustness to adversarial examples with differential privacy. In *Proceedings of 2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. https://doi.org/10.1109/SP.2019.00044

Miyato, Takeru, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582. https://doi.org/10.1109/CVPR.2016.282

Morris, John, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation and adversarial training in NLP. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 119–126. https://doi.org/10.18653/v1/2020.emnlp-demos.16

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162

Ren, Shuhuai, Yihe Deng, Kun He, and Wanxiang Che. 2019a. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1085–1097. https://doi.org/10.18653/v1/P19-1103

Ren, Shuhuai, Yihe Deng, Kun He, and Wanxiang Che. 2019b. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1085–1097. https://doi.org/10.18653/v1/P19-1103

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 856–865. https://doi.org/10.18653/v1/P18-1079

Samanta, Suranjana and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*

Sato, Motoki, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4323–4330. https://doi.org/10.24963/ijcai.2018/601

Shi, Zhouxing, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2020. Robustness verification for transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Wallace, Eric, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for NLP. *arXiv preprint arXiv:1908.07125*. https://doi.org/10.18653/v1/D19-1221

Wang, Wenqi, Lina Wang, Benxiao Tang, Run Wang, and Aoshuang Ye. 2019. A survey: Towards a robust deep neural network in text domain. *arXiv preprint arXiv:* 1902.07285.

Wong, Catherine. 2017. DANCin SEQ2SEQ: Fooling text classifiers with adversarial text example generation. *arXiv preprint arXiv:* 1712.05419.

Xu, Han, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil K. Jain. 2019. Adversarial attacks and defenses in images, graphs and text: A review. *arXiv preprint arXiv:* 1909.08072.

Xu, Kaidi, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. 2020. Automatic perturbation analysis for scalable certified robustness and beyond. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Yang, Puyudi, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. 2018. Greedy attack and Gumbel attack: Generating adversarial examples for discrete data. *arXiv preprint arXiv* 1805.12316.

Ye, Mao, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3465–3475. https://doi.org/10.18653/v1/2020.acl-main.317

Yuan, Liping, Xiaoqing Zheng, Yi Zhou, Cho-Jui Hsieh, and Kai-Wei Chang. 2021. On the transferability of adversarial attacks against neural text classifier. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1612–1625. https://doi.org/10.18653/v1/2021.emnlp-main.121

Zang, Yuan, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6066–6080. https://doi.org/10.18653/v1/2020.acl-main.540

Zhang, Dongxu and Zhichao Yang. 2018. Word embedding perturbation for sentence classification. *arXiv preprint arXiv:* 1804.08166.

Zhang, Xiang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 649–657.

Zhao, Zhengli, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zheng, Xiaoqing, Jiehang Zeng, Yi Zhou, Cho-Jui Hsieh, Minhao Cheng, and Xuanjing Huang. 2020. Evaluating and enhancing the robustness of neural network-based dependency parsing models with adversarial examples. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6600–6610. https://doi.org/10.18653/v1/2020.acl-main.590

Zhou, Yi, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5482–5492. https://doi.org/10.18653/v1/2021.acl-long.426

Zhu, Chen, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. FreeLB: Enhanced adversarial training for natural language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zhu, Zheng-An, Yun-Zhong Lu, and Chen-Kuo Chiang. 2019. Generating adversarial examples by makeup attacks on face recognition. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 2516–2520. https://doi.org/10.1109/ICIP.2019.8803269