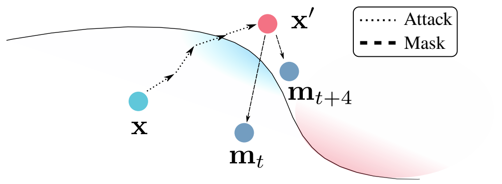
GRADMASK: Gradient-Guided Token Masking for Textual Adversarial Example Detection

Anonymous ACL submission

Abstract

We present a simple model-agnostic textual adversarial example detection scheme called GRADMASK. It uses gradient signals to detect adversarially perturbed tokens in an input sequence and occludes such tokens by a masking process. GRADMASK provides several advantages over existing methods including lower computational cost, improved detection performance, and a weak interpretation of its decision. Extensive evaluations on widely adopted natural language processing benchmark datasets demonstrate the efficiency and effectiveness of GRADMASK. Code and models are available at <redacted>.



$\mathbf{x}$	... The protests during the 2004 Republican convention in New York were not that compelling to make a ...
$\mathbf{x}'$	...The protests during the 2004 Republican convention in New York were not that compel to make a documentary ...
$\mathbf{m}_t$	...The protests during the 2004 Republican convention in New York were not that $\underbrace{\text{[MASK]}}_{m_t}$ to make a $\underbrace{\text{[MASK]}}_{m_{t+4}}$ ...

Figure 1: An illustration of the detection process of GRADMASK with a binary classification example. An attacker tries to find an adversarial example $\mathbf{x}'$ by searching for the best perturbation (*compel*) that flips the original model prediction (expressed as the dotted line). GRADMASK attempts to identify the candidate perturbations through the gradient signal and masks one token (m_t) at a time to generate a masked sequence $\mathbf{m}_t$. The final decision is made by measuring the largest difference in model's confidence for $\mathbf{x}'$ and $\mathbf{m}_t$.

1 Introduction and Related Work

The advances in deep learning has revolutionized natural language processing (NLP) with state-of-the-art performance in practically every task. However, it has been shown that such systems are significantly vulnerable to specifically crafted *adversarial attacks* (Szegedy et al., 2014) at all stages of development and deployment (Ebrahimi et al., 2018; Alzantot et al., 2018; Zhang et al., 2020; Krishna et al., 2020; Tan et al., 2020, 2021). This is quite troubling as there is little to no change in the adversarially chosen test distributions compared to the training distribution (Robin, 2020).

In response to the adversarial attacks, various defense schemes have been proposed. These approaches can be grouped into three categories: (*i*) adversarial training (Si et al., 2020; Maharana and Bansal, 2020; Miyato et al., 2017; Zhu et al., 2020), (*ii*) certified robustness (Jia et al., 2019; Wang et al., 2021), and (*iii*) synonym substitution based methods (Wang et al., 2019, 2020; Dong et al., 2021; Zhou et al., 2021; Jones et al., 2020).

Originally introduced by Goodfellow et al. (2015), the adversarial training methods aim to train a target model on adversarial examples (in additional to clean samples) until the model learns to classify them correctly. However, adversarial training not only increases the training time but also tends to hurt the standard task performance of the model (Tsipras et al., 2019). For NLP, this cost is even greater as many textual attack algorithms rely on an extensive iterative search for potential candidates with a large number of queries (Yuan et al., 2018; Li et al., 2021, 2020; Garg and Ramakrishnan, 2020). In addition, the defense performance largely depends on how well the crafted examples represent the potential attack.

Another branch of adversarial defense scheme is the certified robustness, which aims to provably characterize the output of a model within a restricted space around an input (Cohen et al., 2019). However, certified robustness often requires strong assumptions on the target model architecture. Typically, they have troubles in scaling to large networks such as Transformers (Vaswani et al., 2017). Thus, prior studies (Jia et al., 2019; Wang et al., 2021) mostly adopt recurrent architectures such as

LSTMs (Hochreiter and Schmidhuber, 1997) and convolutional neural networks.

With a growing interest in synonym substitution-based attacks (Garg and Ramakrishnan, 2020; Jin et al., 2020; Ren et al., 2019; Alzantot et al., 2018), there have been a number of studies on defense schemes against such attacks. The goal of these approaches is to encode input texts into a canonical form or robust representation so that the model predictions do not change by synonym substitutions. These methods have shown effectiveness against token-level attacks, but it is unclear how synonym-based defense approaches can protect the model from attacks that perturb tokens aggressively. For instance, synonym-based defense schemes are typically evaluated against token-level attacks such as genetic attack (Alzantot et al., 2018) and PWWS attack (Ren et al., 2019). These defense methods typically construct a synonym set through GloVe (Pennington et al., 2014) or WordNet (Fellbaum, 1998), which are also commonly adopted by the token-level attack algorithms as a synonym-search module (Dong et al., 2021; Zhou et al., 2021; Wang et al., 2019). Thus, it is natural to extend our concern towards a scenario in which these defense schemes can be brittle for defending low-resourced language NLP systems which have no synonym resources, or even for deletion or sub-token perturbation based attacks.

While the above defense schemes aim to improve the adversarial robustness of NLP systems, adversarial example detection methods are designed to reject suspicious inputs although they share the same goal of defeating the adversarial attacks (Aldahdooh et al., 2021). Detection-based approaches provide several advantages over defense schemes. The most obvious advantage is that they do not require to modify the target model architecture or the training procedure, because they typically work as a separate module. Consequently, they do not compromise the model performance on clean datasets. Secondly, they are able to identify the intention (adversarial or not) of adversarial attacks, so users can take actions (reject or revise) accordingly. Finally, the detection algorithms may provide a better strategy for developing defense methods by informing us which parts of an input sequence are perturbed (Zhou et al., 2019).

Unlike the other defense schemes, the textual adversarial detection has not been explored much. To our best knowledge, there are two prior studies trying to detect token-level adversarial attacks. The very first work is the discriminate perturbations (DISP) framework proposed by Zhou et al. (2019). DISP consists of two BERT-BASE (Devlin et al., 2019) based perturbation discriminator and embedding estimator. To provide supervising signals for the discriminator, DISP randomly samples adversarial examples and learns to discriminate clean samples from the adversarial examples. In contrast, a more recent textual adversarial example detection work, the frequency-guided word substitutions (FGWS) approach proposed by Mozes et al. (2021), does not need an additional training process. The key assumption of FGWS is that adversarial attack algorithms tend to exploit words that are rarely exposed during a target model's training. However, as Mozes et al. (2021) mentioned, their approach is limited to detection of only word-level attacks and the effectiveness of FGWS against attacks that do not rely on infrequent words is unclear. Especially, our experiments with a constrained high-frequency vocabulary show that attackers can still find successful attacks by using frequent tokens (§4.1).

Our work in this paper, instead, deviates from the word-frequency assumption by utilizing gradient signals as guidance. We harness the gradient signal to detect adversarially perturbed tokens in an input sequence by investigating the *adversary response*, which, analogous to impulse response or step response (Oppenheim et al., 1996), indicates the network's response to an adversarial input. The identified tokens are subsequently occluded by a mask token and fed to the model to measure the change in model's confidence with respect to the original prediction. Fig. 1 provides an illustration of our gradient-guided detection, GRADMASK.

The gradient-based attribution of neural system's prediction has been studied widely in deep learning (Sundararajan et al., 2017; Simonyan et al., 2014; Li et al., 2016). Some prior work in NLP uses the gradient to identify important words (Li et al., 2017; Murdoch et al., 2018). To the best of our knowledge, this is the first work on detecting textual adversarial attacks by attributing the model prediction via gradient signal analysis.

GRADMASK has several advantages over the previous methods. Firstly, it does not require any additional modules for synonym search or frequent word count. Secondly, our detection algorithm works entirely without any prior knowledge about potential attacks, which is a more practical setup.

Thirdly, it works without any pre-training. Finally, it provides a weak interpretation of decision by identifying adversarially perturbed tokens. The main contributions of this work are:

- We propose GRADMASK, a novel gradient-guided adversarial example detection method.
- We demonstrate that NLP systems can still be significantly brittle to synynym-based adversaries in a high-frequency constrained vocabulary setup, a finding that deviates from the frequency-based assumption of Mozes et al. (2021).
- We demonstrate the advantage of GRADMASK over state-of-the-art adversarial example detection algorithm through extensive experiments.

2 Method

In this section, we present our proposed method. We first establish the notations in §2.1.

2.1 Notations

We consider a standard text classification task for a model $f_{\boldsymbol{\theta}}(\cdot)$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$. The model $f_{\boldsymbol{\theta}}(\cdot)$ is trained to fit a data distribution $\mathcal{D}$ over pairs of an input sequence $\mathbf{x} = [x_1, \cdots, x_T]$ of T tokens and its corresponding label $y \in \{1, \ldots, C\}$ with C being the number of classes. We also assume a loss function $\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)$ such as a cross-entropy loss. The output of the model is a probability distribution that satisfies: $0 \leq f_{\boldsymbol{\theta}}(\mathbf{x})_i \leq 1$ and $\sum_{i=1}^{C} f_{\boldsymbol{\theta}}(\mathbf{x})_i = 1$, where i is the class index. We denote the final prediction as $c(\mathbf{x}) = \arg\max_i f_{\boldsymbol{\theta}}(\mathbf{x})_i$ and true label as $c^*(\mathbf{x}) = y^*$.

Given a sequence $\mathbf{x}$, a textual adversarial example $\mathbf{x}'$ can be defined as follows: for some semantic dissimilarity measure $\delta(\mathbf{x}, \mathbf{x}')$, it has to be small and $c(\mathbf{x}') \neq c^*(\mathbf{x})$. These two conditions denote that an adversarial example has to maintain semantic meaning of the original input $\mathbf{x}$ but misguide the model prediction (Athalye et al., 2018).

2.2 Gradient-guided Token Masking for Adversarial Example Detection

GRADMASK first finds salient tokens that significantly attribute to the model prediction, $c(\mathbf{x})$; see Fig. 1 for an illustration. A simple and widely employed approach is the gradient-based attribution analysis (Ancona et al., 2018; Sundararajan et al., 2017; Li et al., 2016). However, due to the discrete nature of texts, we cannot directly exploit the

Algorithm 1 Gradient-based Masking for Adversarial Example Detection.

Require: Input sequence $\mathbf{x}$, target model $f_{\boldsymbol{\theta}}$

```
Initialize M = {} and K = ⌊T × p⌋.
Compute f_θ(x)_i, where i = c(x).          ▷ pred. for x
L := {||g_1||, ··· , ||g_T||} via Eq. 1.
Sort L in descending order.
while k ≤ K do
    ||g||_t ← L[k]
    m_t = [x_1, ··· , m_t, ··· , x_T]
    M[k] = f_θ(m_t)_i                        ▷ prediction for m_t
end while
w = (f_θ(x)_i − min_k M[k])^2
```

gradient-based approach. In order to deviate the issue, we compute a gradient of the word embedding $\mathbf{e}_t$ with regard to the loss function $\mathcal{L}$, where $\mathbf{e}_t$ is a simple linear projection of a (subword) token x_t. The gradient can be expressed as follows:

$$\mathbf{g}_t = \nabla_{\mathbf{e}_t} \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, c(\mathbf{x})) \tag{1}$$

Note that the above loss is computed with respect to the model's final prediction $c(\mathbf{x})$ and not the ground truth y^*.

Subsequently, we measure the amount of stimulus of the input tokens toward the model prediction by computing the L_2-norm of $\mathbf{g}_t$. The stimulus is considered as a saliency score of the tokens and it is determined in descending order of the magnitude of $||\mathbf{g}_t||_2$ following Li et al. (2016). GRADMASK only considers the top-p portion of the input tokens in $\mathbf{x}$. Specifically, the number of chosen K salient tokens is $\lfloor T \times p \rfloor$, where the brackets denote the floor operation. The sampled K salient tokens are masked individually one at a time to generate a masked input sequence $\mathbf{m}_t = [x_1, \ldots, m_t, \ldots, x_T]$ with t being the token position of a salient token, and m_t is the mask token, [MASK].[1]

The rationale behind the masking approach is based on two assumptions. The first assumption is that *adversarial examples are the result of sophisticated optimization algorithms rather than the result of random perturbations* (Goodfellow et al., 2015; Galloway et al., 2018). Thus, we conjecture that masking the suspicious tokens which are carefully crafted can significantly drop the model confidence. The second assumption is that *NLP systems are generally robust to weak-level of noise*.

[1]In case of non-masked language model-based classifiers, we adopted an unknown token.

The partial information loss in clean samples due to masking can be offset by the overall context of the input text (supported by our experiments in §4.1).

Each masked sequence $\mathbf{m}_t$ is then fed into the target model to get a prediction $f_{\boldsymbol{\theta}}(\mathbf{m}_t)_i$, where $i = c(\mathbf{x})$. This process gives K such confidence scores which are stored in $\mathcal{M}$. We then compare the minimum confidence value in $\mathcal{M}$ to the original confidence score $f(\mathbf{x})_i$, and the confidence change is squared to assign a stronger penalty to the higher changes. More formally,

$$w = \left(f_{\boldsymbol{\theta}}(\mathbf{x})_i - \min_k \mathcal{M}[k]\right)^2 \tag{2}$$

The final decision is determined by an indicator function $\mathcal{I}(w, \tau)$ defined as follows:

$$\mathcal{I}(w, \tau) = \begin{cases} 0 & \text{if } w \leq \tau \\ 1 & \text{else} \end{cases} \tag{3}$$

where τ is a pre-defined threshold. Alg. 1 presents the overall process of GRADMASK.

3 Experiment Settings

In this section, we present our experiment settings: the datasets, target models, adversarial example generation and evaluation metrics.

3.1 Datasets

We evaluate the methods on three classification tasks. We use the IMDB (Maas et al., 2011), AG-NEWS (Zhang et al., 2015), and Stanford Sentiment Treebank (SST) (Socher et al., 2013) datasets that are widely adopted for benchmarking adversarial robustness of NLP systems. The IMDB dataset contains movie reviews labeled with positive or negative sentiment labels. The AGNEWS dataset contains news articles from more than 2,000 news sources and the samples are categorized into the four largest classes. The SST dataset provides movie reviews with fine-grained sentiment labels. We turn the labels into binary (SST-2) to follow the setting of FGWS (Mozes et al., 2021). Table 1 gives an overview of the datasets.

Dataset	Train / Test	Avg. Len
IMDb	25k/25k	215
AG	120k/7.6k	43
SST-2	67k/1.8k	20

Table 1: A summary of the datasets used in our work.

3.2 Target Models

We evaluate GRADMASK on three different sequence modeling architectures, which have been widely employed in NLP. We first consider a large-scaled pre-trained Transformer-based language model, ROBERTA-BASE (Liu et al., 2019), which contains 124 million parameters. Subsequently, we also evaluate on a relatively smaller Transformer-based model called DISTILBERT-BASE (Sanh et al., 2020), which has approximately 40% fewer parameters than ROBERTA-BASE. Finally, we consider the LSTM, which used to be the dominant architecture before the arrival of Transformers.

MODEL	DATASET	ACC (%)
ROBERTA	IMDB SST-2 AG	93.36 91.98 95.3
ROBERTA-LONG	IMDB SST-2	93.71 88.69
DISTILBERT	IMDB SST-2 AG	90.57 91.21 94.37
LSTM	IMDB SST-2	87.27 83.53

Table 2: A summary of the target models and their clean testset performance.

Table 2 shows the standard task performance of the models on the three datasets. To train the models, we followed the hyperparameter settings provided by Mozes et al. (2021). The TRANSFORMER based models are optimized by AdamW (Loshchilov and Hutter, 2019) with a linear adaptive learning rate scheduler. For LSTM, the initial word embeddings are initialized with GloVe (Pennington et al., 2014). The texts in IMDB are comparatively longer than those in AGNEWS and SST-2. For the IMDB classification task, the maximum sequence lengths for ROBERTA, DISTILBERT and LSTM are set to 256, 256, and 200, respectively, and ROBERTA-LONG is trained with a longer sequence (400 tokens) than the standard one. The details of model architectures are provided in the supplementary material. All of the experiments are conducted on an Intel Xeon Gold 5218R CPU-2.10GHz processor with a single Quadro RTX 6000 GPU.

3.3 Adversarial Example Generation

We generated adversarial examples against the selected target models via four different attack algorithms. They include two baseline attacks and two widely adopted synonym substitution-based token-level attacks, as used in previous work

- **Random** is a simple word replacement-based baseline attack algorithm. It randomly selects a synonym of a token in the original input text. Synonyms are identified via WordNet.

- **Prioritized** attack is also based on word replacement, but it puts a higher priority on a synonym that maximizes the target model's prediction confidence change.

- **Genetic** attack (GA) was proposed by Alzantot et al. (2018). It adopts the crossover and mutation operations in genetic algorithms to generate adversarial examples. GA searches synonyms based on the GloVe word embedding space with a language model (Radford et al., 2019).[2]

- **PWWS** or Probability weighted word saliency (Ren et al., 2019) is a greedy word substitution-based attack algorithm. The word replacement order is determined by a word saliency score computed through the model's confidence change. The word synonym is searched via WordNet.

3.4 Evaluation Metrics

The main interest of this work lies in an evaluation of the detection performance of our proposed method GRADMASK. FGWS (Mozes et al., 2021) was mainly evaluated via F1 score, but we follow the standards from the out-of-distribution (OOD) sample detection literature (Zheng et al., 2020; Hendrycks et al., 2019; Ouyang et al., 2021) for better understanding of the methods.

The adversarial example detection can be considered as a binary classification problem of verifying *positive (adversarial)* vs. *negative (clean)* class. We evaluate a ratio of true positive samples so-called true positive rate (TPR or recall) against false positive rate (FPR) defined as:

$$TPR = \frac{1}{n^+}\sum_i \mathcal{I}(w^+, \tau) \tag{4}$$

$$FPR = \frac{1}{n^-}\sum_i \mathcal{I}(w^-, \tau), \tag{5}$$

where the superscripts $+$ and $-$ denote the positive and the negative classes, respectively. Based on these two rates, we evaluate the methods with the following evaluation metrics:

[2] We adopted the modified implementation provided by Mozes et al. (2021) for a fair comparison. The details are provided in the supplementary material.

- **AUROC** stands for the area under receiver operating characteristic curve. For each operational setting of τ from 0 to 1, TPR and FPR can be plotted. This curve is called receiver operating characteristic curve (ROC curve).

- **FPR95** refers to a FPR at 95 TPR. FPR95 quantifies how many clean samples have to be rejected to detect 95% of the adversarial examples. FPR is a very important metric for evaluating detection algorithms (Aldahdooh et al., 2021). A lower FPR95 score is often required for systems that require a high level of system safety or security.

- **AUPR** denotes area under precision-recall (PR) curves. There exists an imbalance of data distribution between positive class and negative class. To deal with the data distribution skew, we evaluate AUPR scores for each class.

4 Results & Analysis

We first investigate the relationship between the adversarial robustness of NLP classification models and the word frequency in the adversarial examples (§4.1). We then analyze the adversarially perturbed token detection performance of GRADMASK (§4.2). In §4.3, we evaluate GRADMASK on widely employed NLP benchmarks. Finally, we investigate GRADMASK's potential against a non-synonym based (character-level) attack §4.4.

4.1 Word Frequency and Adversarial Robustness of NLP Systems

According to Mozes et al. (2021), the brittleness of NLP systems against adversarial examples would be attributed to the distribution of word frequency in a training set. However, one of the widely accepted explanations about the existence of adversarial examples insists that adversarial examples are a result of the standard optimization rather than data distribution (Ilyas et al., 2019). We investigated how the word frequency can affect the model's robustness via a series of experiments. Consequently, we find that *deep NLP systems can still be fooled by adversarial examples with words that are frequently exposed during their training stage*.

To validate this claim, we trained the victim models with a word frequency constraint. Specifically, we built a new vocabulary set V' to be comprised of only the top-10% frequently used words from the original vocabulary set V. The vocabulary-constrained models are designed to block all infrequent words that are out of V' in an input sequence

Model	Dataset	Acc-V	Acc-V'	$x' \in V'$	AAcc
DISTILBERT	IMDb	92.98	92.17	71.73	10.4
	AG	94.37	90.78	68.92	15.6
ROBERTA	IMDb	95.33	95.15	67.38	7.6
	AG	95.22	94.87	44.26	30.8

Table 3: Word frequency and adversarial robustness. Acc-V and Acc-V' refer to accuracies of the model with the original vocabulary V and constrainted vocabulary V', respectively. $x' \in V'$ denotes a ratio of perturbed tokens that are part of V'. AAcc denotes an under attack accuracy of the model with V'.

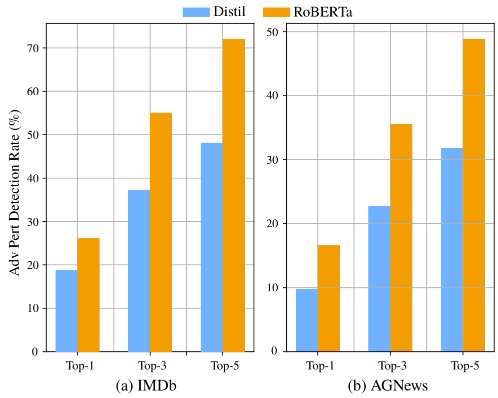


Figure 2: Adversarially perturbed token detection rates at top-1, top-2 and top-5 for GRADMASK.

by masking those tokens. We first evaluated the model performance to observe how the vocabulary constraint affects the model performance. As shown in Table 3, the standard task performance of the victim models under the constraint (**Acc**-V') only marginally decreases (about 1 - 4%) compared to the original accuracy (**Acc**-V). These results show that masking infrequent tokens does not hurt the model performance significantly.

Next, we generated 1,000 pairs of samples via the PWWS attack algorithm (Ren et al., 2019) against the word frequency constrained models.[3] Each sample pair consists of a clean example and its corresponding adversarial example that successfully fools the target model.

According to the infrequent word assumption (Mozes et al., 2021), the models trained on V' are expected to be robust against adversarial attacks. However, from the results in Table 3, we notice that they showed significant brittleness against adversarial attacks. The attack algorithms deviate from the masking strategy by using frequent words that are within V' ($x' \in V'$). For instance, 71.7% adversarially perturbed tokens in the adversarial examples against DISTILBERT model are in the constrained vocabulary set V'. DISTILBERT models show approximately 10% accuracies for both datasets when under attack (**AAcc**). Similarly, ROBERTA models show under attack accuracies of 7.6% and 30.8% for AGNEWS and IMDB, respectively. Thus, we claim that *the vulnerabilities of NLP systems cannot only be attributed to the infrequent words*.

4.2 Adversarial Token Detection

We now analyze how our gradient-based approach GRADMASK attributes the model prediction on adversarial examples. Fig. 2 shows perturbed token detection rates of two Transformer-based models, DISTILBERT and ROBERTA, on two datasets, IMDB and AGNEWS. We report detection rates at top-1, top-3, and top-5, which refers to the total number of adversarially perturbed tokens identified within the top-N values of w in Eq. (2). In case of DISTILBERT, it shows 48.17% and 31.82% detection rates for IMDB and AGNEWS within the top-5 predictions, respectively. On the other hand, ROBERTA shows 72.04% and 48.85% detection rates for IMDB and AGNEWS within the top-5 predictions. Another notable observation is that for the IMDB classification task, top-1 predictions detect the adversarial tokens with 49% and 78% probability for DISTILBERT and ROBERTA, respectively. For AGNEWS, their top-1 predictions show 45% and 67% detection probability, respectively.

4.3 Adversarial Example Detection

For adversarial example detection, we compare the performance of GRADMASK with that of FGWS (Mozes et al., 2021). The hyperparameter settings of FGWS is tuned as provided by Mozes et al. (2021).[4] The overall experimental results are presented in Table 4. Note that AUPR-C and AUPR-A represent the AUPR score of clean samples (negative class) and that of adversarial samples (positive class), respectively.

As shown in Table 4, GRADMASK tends to show better AUROC, FPR95, and AUPR-C scores in most of the evaluation measures. Particularly, it significantly outperforms FGWS for all Transformer-based systems (ROBERTA, ROBERTA-LONG, and DISTILBERT) in terms of the FPR95 score,

[3] We adopted TextAttack framework (Morris et al., 2020) to attack the victim models. Their implementation difference is provided in the supplementary material.

[4] https://github.com/maximilianmozes/fgws

Model	Dataset	# Samples		Attack	AUROC (%)		FPR95 (%)		AUPR-C (%)		AUPR-A (%)		K
		TN	TP		FGWS	GM	FGWS	GM	FGWS	GM	FGWS	GM	
RoBERTa	IMDb	2000	147	Random	86.06	**94.97**	84.98	**14.25**	98.46	**99.62**	**51.55**	43.5	1
		2000	995	Prioritized	92.67	**95.55**	68.31	**11.1**	95.06	**98.12**	**89.2**	84.89	1
		2000	1042	Genetic	89.88	**95.69**	78.53	**11.4**	92.89	**98.17**	**86.72**	85.04	1
		2000	1016	PWWS	85.85	**95.38**	85.17	**13.15**	90.47	**98**	83	**84.92**	1
	SST-2	1821	148	Random	75.4	**81.43**	90.54	**52.39**	97.17	**98.18**	**37.62**	20.37	1
		1821	479	Prioritized	**83.57**	82.09	84.69	**54.26**	94.23	**94.65**	**65.35**	46.95	1
		1821	968	Genetic	74.6	**79.19**	90.82	**56.89**	84.22	**90.97**	**66.55**	61.33	1
		1821	736	PWWS	77.72	**82.73**	65.06	**51.29**	88.66	**92.44**	**66.05**	58.51	1
RoBERTa-Long	IMDb	2000	190	Random	81.05	**94.50**	89.77	**16.70**	97.26	**99.46**	**58.84**	52.65	1
		2000	1037	Prioritized	93.08	**94.75**	68.20	**16.00**	95.02	**97.60**	**90.70**	85.41	1
		2000	888	Genetic	89.05	**95.51**	80.96	**13.60**	93.24	**98.25**	**85.38**	85.34	1
		2000	1129	PWWS	87.10	**95.01**	84.38	**15.70**	90.26	**97.44**	86.38	**88.35**	1
	SST-2	1821	176	Random	**76.42**	75.72	89.34	**60.35**	96.94	**96.97**	**35.15**	18.24	1
		1821	527	Prioritized	**79.80**	77.73	87.06	**60.08**	92.71	**92.78**	**62.95**	43.31	1
		1821	960	Genetic	68.18	**73.55**	92.15	**69.80**	82.55	**84.89**	**61.46**	53.11	1
		1821	772	PWWS	75.54	**78.57**	90.05	**57.50**	87.83	**90.41**	**66.44**	54.38	1
DistilBERT	IMDb	2000	212	Random	83.36	**87.66**	86.98	**37.30**	97.46	**98.56**	**59.59**	33.33	1
		2000	1182	Prioritized	**93.20**	89.66	62.85	**31.70**	**94.79**	94.50	**91.88**	76.09	1
		2000	1202	Genetic	**90.28**	90.23	75.59	**22.80**	**92.50**	95.27	**89.25**	74.41	1
		2000	1335	PWWS	86.56	**88.74**	83.06	**36.64**	88.9	**92.93**	**86.95**	79.10	1
	SST-2	1821	171	Random	**83.17**	77.78	84.42	**59.69**	87.77	**97.32**	**37.23**	18.40	1
		1821	614	Prioritized	**84.29**	78.87	84.36	**58.70**	**92.97**	92.34	**70.36**	46.86	1
		1821	1105	Genetic	74.74	**78.06**	90.97	**49.81**	82.27	**88.18**	**69.36**	57.32	1
		1821	860	PWWS	**80.30**	78.87	71.56	**54.31**	88.25	**89.93**	**71.56**	54.41	1
LSTM	IMDb	2000	198	Random	77.82	**84.22**	89.64	**37.55**	96.90	**98.31**	**44.47**	24.87	20
		2000	1451	Prioritized	**88.34**	86.64	78.68	**30.50**	89.66	**92.41**	**88.66**	73.90	20
		2000	1548	Genetic	77.47	**86.59**	89.73	**30.50**	81.04	**92.00**	**78.92**	74.50	20
		2000	1735	PWWS	80.53	**86.99**	88.85	**30.90**	81.47	**91.45**	**83.85**	78.43	20
	SST-2	1821	238	Random	**79.14**	58.45	**86.35**	98.13	**96.36**	90.22	**36.37**	13.35	20
		1821	669	Prioritized	**74.97**	68.45	**89.89**	95.18	**88.73**	84.33	**57.21**	36.24	20
		1821	1186	Genetic	**71.37**	66.74	**91.28**	96.00	**80.08**	72.67	**66.55**	51.55	20
		1821	1013	PWWS	**74.68**	69.59	**90.28**	95.51	**83.96**	78.51	**66.46**	48.26	20

Table 4: Adversarial example detection results of FGWS and GradMask (GM). AUPR-C and AUPR-A denote AUPR of clean example and adversarial example classes, respectively.

which is an important metric for systems with high security requirements. In addition, GradMask achieves notably better AUPR-C scores in most of the experiment scenarios. This tendency is well presented in Fig. 3, which shows ROC curves of FGWS and GradMask for RoBERTa model. The ROC curves of FGWS tend to increase steeply and remain stable. However, as TPR increases, FGWS significantly compromises FPR score. Especially, at some point, TPR and FPR show a linear trend. In contrast, GradMask tends to reach 95% TPR at lower FPR scores and shows larger AUROC scores.

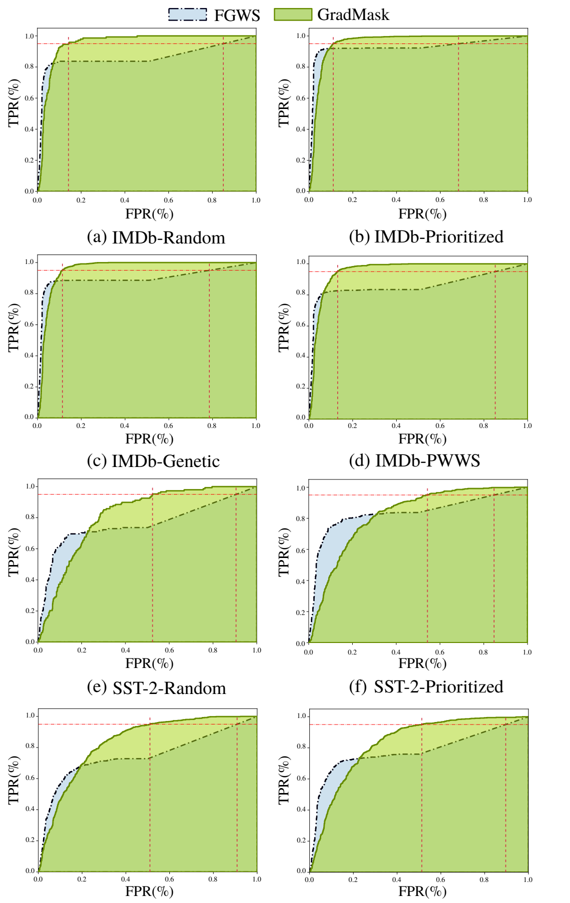
On the other hand, GradMask shows lower performance scores in all metrics on SST-2 with the LSTM model as shown in Table 4. Nevertheless, the overall detection performance of GradMask tends to improve proportionally to the model size and the standard performance. Another notable observation is that GradMask achieves these results with a single token masking except for the LSTM model (K in Table 4). These results may imply that NLP systems are largely robust to a partial loss of information resulting from the masking strategy on clean samples, but there is a significant change in the adversary response caused by a salient token masking. Also, our gradient-based masking strategy occasionally detects adversarial examples through masking a clean token as presented in §4.2 and Fig. 2. This result implies that the hidden representation of adversarial tokens significantly affects that of clean tokens.

Moreover, GradMask shows consistently better performance in detecting strong attacks such as genetic attack and PWWS attack which are more aggressive than the others. We conjecture that stronger attacks select and engineer the crucial tokens more carefully, so masking these tokens would hugely reduce the effectiveness of these attacks.

We also observe that GradMask underperforms FGWS in terms of AUPR-A. A possible explanation may be related to the nature of the synonym substitution strategy. We hypothesize that FGWS tends to transform an input sequence aggressively. This view can be supported by their FPR95 scores and precision-recall (PR) curves. Firstly, the ROC curves of FGWS typically show high FPRs at high TPRs (Fig. 3). Secondly, from the PR curves of FGWS shown in Fig. 4, the precision scores drop significantly as the recall scores increase. We provide PR curves for 6 other scenarios in the supplementary material.

Model	Dataset	# Samples TN	# Samples TP	Attack	AUROC Mask	FPR95 Mask	AUPR-C Mask	AUPR-A Mask
RoBERTa	IMDb	691	691	Character	79.68	67.44	78.75	75.8
Distil	IMDb	897	897	Character	80.42	63.76	81.02	75.07

Table 5: Adversarial example detection results against a character-level attack.



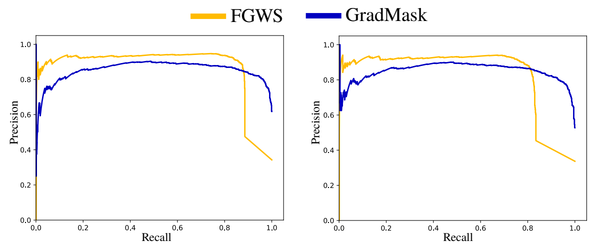
(a) IMDb-Random (b) IMDb-Prioritized (c) IMDb-Genetic (d) IMDb-PWWS (e) SST-2-Random (f) SST-2-Prioritized (g) SST-2-Genetic (h) SST-2-PWWS

Figure 3: ROC curves of FGWS and GradMask with the RoBERTa model. The horizontal red line is at the 95% TPR and the vertical lines at the FPRs of two algorithms, respectively (best viewed in color).

4.4 Character-Level Attack Detection

To investigate the potential of GradMask against non-synonym based attacks, we conduct an additional experiment with a character-level attack (Pruthi et al., 2019) from the TextAttack library (Morris et al., 2020). Even though character-level attacks are known to be relatively simple to defend at a preprocessing stage with a spell or a grammar checker (Pruthi et al., 2019), our motivation for this experiment is to demonstrate the potential of GradMask against non-synonym based attacks.

We generated adversarial examples against RoBERTa-base and Distil-base without any maximum text length limitation. From the results in Table 5, we see that our method shows promising results with AUROC scores of 79.68% and 80.42% for RoBERTa-base and Distil-base, respectively. It would be interesting to see how GradMask performs for other kinds of non-synonym attacks such as syntactically controlled paraphrase networks (SCPNs) (Iyyer et al., 2018) or universal adversarial attack (Song et al., 2021) which we leave as future work.



(a) IMDb-Genetic (b) IMDb-PWWS

Figure 4: Precision-Recall curves of FGWS and GradMask on IMDb with the RoBERTa model against the PWWS and genetic attacks.

5 Conclusion

We have proposed a simple model-agnostic adversarial example detection scheme, GradMask, which is designed to utilize gradient signals as a guidance to detect adversarially perturbed tokens. This guidance additionally provides a weak interpretation about its decision. The experimental results show that GradMask is a promising approach as a textual adversarial attack detection algorithm for NLP classification systems. Particularly, it shows significantly low FPR95 scores, which is a highly desirable property for NLP systems with high-security requirements. In addition, GradMask does not require an additional module or a strong assumption about potential attacks which are more realistic in practice. Finally, we have shown that adversarial perturbations with frequent words can successfully fool the NLP classification systems. In conclusion, our detection strategy can serve as a useful tool for identifying adversarial attacks for protecting the text classification systems.

References

Ahmed Aldahdooh, Wassim Hamidouche, Sid Ahmed Fezza, and Olivier Déforges. 2021. Adversarial example detection for DNN models: A review.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*.

Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.

Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR.

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

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

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

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Angus Galloway, Graham W. Taylor, and Medhat Moussa. 2018. Attacking binarized neural networks. In *International Conference on Learning Representations*.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. 2019. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142, Hong Kong, China. Association for Computational Linguistics.

D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 8018–8025. AAAI Press.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. 2020. Thieves on sesame street! model extraction of BERT-based APIs. In *International Conference on Learning Representations*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, M. Sun, and B. Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Understanding neural networks through representation erasure.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrew L. Maas, 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 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Adyasha Maharana and Mohit Bansal. 2020. Adversarial augmentation policy search for domain and cross-lingual generalization in reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *International Conference on Learning Representations*.

John X. Morris, 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.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186, Online. Association for Computational Linguistics.

W. James Murdoch, Peter J. Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *International Conference on Learning Representations*.

Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. 1996. *Signals Systems (2nd Ed.)*. Prentice-Hall, Inc., USA.

Yawen Ouyang, Jiasheng Ye, Yu Chen, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2021. Energy-based unknown intent detection with data manipulation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language ProcessingEMNLP*, volume 14, pages 1532–1543.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, volume abs/1905.11268.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Jia Robin. 2020. *Building robust natural language processing systems*. Ph.D. thesis, Stanford University, Stanford, California.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Better robustness by more coverage: Adversarial training with mixup augmentation for robust fine-tuning. *CoRR*, abs/2012.15699.

K. Simonyan, A. Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

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

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal adversarial attacks with natural triggers for text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3724–3733, Online. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

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

Samson Tan, Shafiq Joty, Kathy Baxter, Araz Taeihagh, Gregory A. Bennett, and Min-Yen Kan. 2021. Reliability testing for natural language processing systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, ACL'21, page 4153–4169, Bangkok, Thailand. ACL.

Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. It's morphin' time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, Online. Association for Computational Linguistics.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Wenjie Wang, Pengfei Tang, Jian Lou, and Li Xiong. 2021. Certified robustness to word substitution attack with differential privacy. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1102–1112, Online. Association for Computational Linguistics.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *CoRR*.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2020. Adversarial training with fast gradient projection method against synonym substitution based text attacks. *CoRR*, abs/2008.03709.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2018. Adversarial examples: Attacks and defenses for deep learning.

Wei Emma Zhang, Quan Z. Sheng, and Ahoud Abdulrahmn F. Alhazmi. 2020. Generating textual adversarial examples for deep learning models: A survey. *ACM Trans. Intell. Syst. Technol.*

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pages 649–657. Curran Associates, Inc.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain detection for natural language understanding in dialog systems.

Yi Zhou, 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, Online. Association for Computational Linguistics.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4904–4913, Hong Kong, China. Association for Computational Linguistics.

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

Table 6: Parameter settings of target models. AL and MAXLEN denote the adaptive linear learning rate scheduler and maximum sequence length, respectively.

MODEL	PARAMETERS	
ROBERTA	OPTIMIZER	ADAMW
	BATCH SIZE (IMDB/SST-2)	16/32
	EPOCHS	10
	LEARNINGRATE	10^{-5}
	LEARNINGRATE SCHEDULER	AL
	MAXLEN (IMDB/SST-2)	256/128
ROBERTA-LONG	OPTIMIZER	ADAMW
	BATCH SIZE (IMDB/SST-2)	16/32
	EPOCHS	10
	LEARNINGRATE	10^{-5}
	LEARNINGRATE SCHEDULER	AL
	MAXLEN (IMDB/SST-2)	400/256
DISTILBERT	OPTIMIZER	ADAMW
	BATCH SIZE (IMDB/SST-2)	16/32
	EPOCHS	10
	LEARNINGRATE	10^{-5}
	LEARNINGRATE SCHEDULER	AL
	MAXLEN (IMDB/SST-2)	256/128
LSTM	OPTIMIZER	ADAM
	BATCH SIZE (IMDB/SST-2)	100/100
	HIDDEN SIZE	128
	DROPOUT	0.1
	EMBEDDING	GLOVE
	EPOCHS	20
	LEARNINGRATE	10^{-3}
	MAXLEN (IMDB/SST-2)	200/50

A Model Parameters

Table 6 summarizes the parameter settings of the target models used for adversarial example detection experiments. We follow the model settings of (Mozes et al., 2021) except ROBERTA-LONG which is trained on a longer maximum sequence length setting.

B Adversarial Attack Implementation

For adversarial example detection experiments (§4.3), we adopted the implementation provided by Mozes et al. (2021). According to Mozes et al. (2021), they replaced Google language model (Chelba et al., 2013) in genetic attack with GPT-2 language model (Radford et al., 2019) for computational efficiency.

Note that for word-frequency analysis (§4.1) and adversarial token detection (§4.2) experiments we employed the publicly available TextAttack library (Morris et al., 2020) for PWWS attack (Ren et al., 2019). The main difference from the original implementation is PWWS attack in TextAttack does not include the named entity (NE) adversarial swap, because it requires NE labels of input sequences that are not available in practice (Morris et al., 2020).

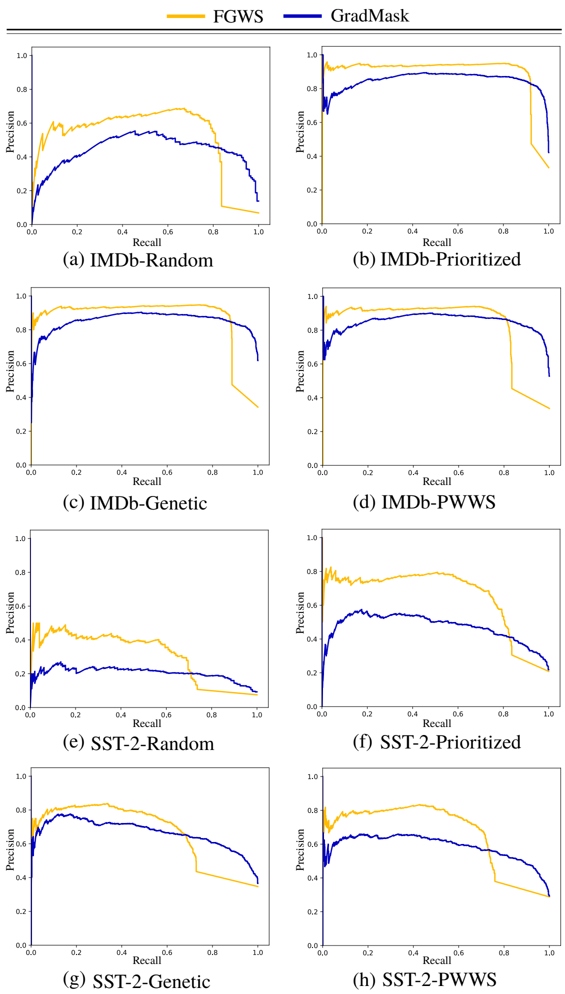


Figure 5: PR curves of FGWS and GRADMASK on IMDB and SST-2 ROBERTA models against four different attacks.

C Precision-Recall Curve of ROBERTA Model

Fig. 5 presents PR curves of FGWS and GRADMASK ROBERTA models trained on IMDB and SST-2 against four different attacks. As mentioned in §4.3, we observe the tendency that the overall precision scores of the FGWS algorithm drop at high recall scores. However, our method maintains high precision scores at high recall scores.