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

Han Cheol Moon
Nanyang Technological University
Singapore
hancheol001@e.ntu.edu.sg

Shafiq Joty
Nanyang Technological University
Salesforce Research, Singapore
srjoty@ntu.edu.sg

Xu Chi
Nanyang Technological University
Singapore Institute of Manufacturing Technology, Singapore
cxu@simtech.a-star.edu.sg

ABSTRACT

We present GradMask, a simple adversarial example detection scheme for natural language processing (NLP) models. 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 improved detection performance and an interpretation of its decision with a only moderate computational cost. Its approximated inference cost is no more than a single forward- and back-propagation through the target model without requiring any additional detection module. Extensive evaluation on widely adopted NLP benchmark datasets demonstrates the efficiency and effectiveness of GradMask. Code and models are available at https://github.com/Han8931/grad_mask_detection.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**.

KEYWORDS

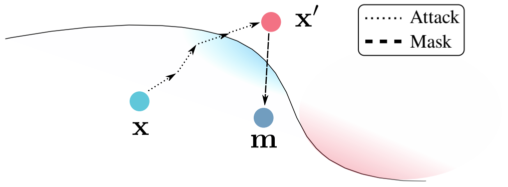
Adversarial attacks, Adversarial example detection

ACM Reference Format:
Han Cheol Moon, Shafiq Joty, and Xu Chi. 2022. GradMask: Gradient-Guided Token Masking for Textual Adversarial Example Detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539206

1 INTRODUCTION AND RELATED WORK

The advances in deep learning have 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* [50] at all stages of development and deployment [2, 8, 21, 51, 52, 63]. This is troubling as there is little to no change in the adversarially crafted test distributions compared to the training distribution [40].

In response to the adversarial attacks, various defense schemes have been proposed (see [62] for a survey). These approaches can be grouped into three main categories: (*i*) adversarial training [30, 31, 45, 68], (*ii*) synonym substitution based methods [7, 20, 56, 57, 67], and (*iii*) certified robustness [18, 55].



$\mathbf{x}$	... during the 2004 Republican convention in New York were not that compelling to make a ...
$\mathbf{x}'$	... during the 2004 Republican convention in New York were not that compel to make a documentary ...
$\mathbf{m}$	... during the 2004 Republican convention in New York were not that [MASK] to make a [MASK]...

Figure 1: An illustration of GradMask's adversarial example detection process on a binary classification task. Given an input $\mathbf{x}$, 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 perturbation(s) through the gradient signal and masks the top-K tokens to generate a masked sequence $\mathbf{m}$. The final decision is made by measuring the largest difference in model's confidence for $\mathbf{x}'$ and $\mathbf{m}$.

Another branch of defense strategy that is underexplored in NLP is the *adversarial example detection* based schemes. While the above defense schemes aim to improve the adversarial robustness of NLP systems, adversarial example detection methods are designed to reject suspicious inputs, thus they share the same goal of defeating the adversarial attacks [1]. Detection-based approaches provide several advantages over adversarial robustness improvement methods. 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. Additionally, the "discard-rather-than-correct" strategy could be a desirable feature for production systems where models are expected to retain certain information about the original data distribution (*e.g.,* customer specific information for product reviews), and any

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539206

change of the model through adversarial examples - whether correct or not - will be undesirable. Finally, the detection algorithms may provide a better strategy for developing defense methods by informing which parts of an input sequence are perturbed [66].

Unlike the other defense schemes, the textual adversarial detection problem has not been explored much in NLP. The very first work is the discriminate perturbations (DISP) framework [66], which consists of two BERT [6] based perturbation discriminator and embedding estimator. To provide supervising signals for the discriminator, DISP randomly samples adversarial examples and learns to discriminate clean examples from the adversarial examples. In contrast, a more recent adversarial detection work, the frequency-guided word substitutions or FGWS [33], 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, this approach is limited to detection of only word-level attacks and its effectiveness 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 (Appendix B.4). Unlike those synonym-based adversarial attack detection algorithms, Le et al. [22] proposed a detection framework for universal adversarial trigger or UAT [54], which is a sequence that is concatenated to input texts to misguide the model prediction. However, UAT tends to significantly violate grammars and semantics of the original inputs compared to synonym-based attacks.

Our work in this paper focuses on detecting synonym substitution-based adversarial attacks. Our proposed method attempts to reduce assumptions about characteristics of potential attacks. In practice, we have no access to a perturbation process of attackers. Thus, we first deviate from the word-frequency assumption by utilizing gradient signals as guidance. Specifically, we harness the gradient signal to detect (potential) adversarially perturbed tokens in an input sequence by investigating the *sensitivity* of the model prediction [3, 25, 49, 61], which 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 the model's confidence with respect to the original prediction. The masking process allows avoiding searching synonyms of potential perturbations generated from attackers that are unknown to the target systems in practice. Figure 1 illustrates our gradient-guided detection, GradMask.

The gradient-based attribution of neural model's prediction has been studied widely in deep learning [25, 46, 49]. Some prior work in NLP uses the gradient to identify important words in a sequence [26, 35]. However, 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. First, it does not require any additional modules for synonym search or frequent word count that are essential in the previous methods [33, 66]. Second, it works entirely without any prior knowledge about potential attacks, which is a more practical setup. Third, 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:

- A novel gradient-guided adversarial example detection method, GradMask, which makes minimal assumptions about potential attacks.
- Extensive experiments with Transformer-based models [53] for textual classification tasks and a natural language inference task against various textual adversarial attacks demonstrating GradMask's advantage over state-of-the-art detection algorithms.
- Results showing GradMask outperforms baseline algorithms (including the traditional anomaly detection methods) in terms of AUROC, EER, and FPR95 measures.
- Further analysis showing GradMask also achieves promising results in detecting character-level attacks [38].
- Ablation studies showing the effectiveness of the gradient-based search and the operation components of GradMask.

2 METHOD

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

2.1 Notations

We consider a standard text classification task for a model $f_{\boldsymbol{\theta}}(\cdot)$ with parameters $\boldsymbol{\theta} \in R^p$. 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 (ground truth) label as $c^*(\mathbf{x}) = y^*$.

Given an input sequence $\mathbf{x}$, a successful adversarial example $\mathbf{x}'$ can be defined as follows: the semantic dissimilarity between $\mathbf{x}$ and $\mathbf{x}'$ has to be small according to some dissimilarity measure $\delta(\mathbf{x}, \mathbf{x}')$, 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 but misguide the model prediction [4].

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 Figure 1 for an illustration. A simple and widely employed approach is the gradient-based attribution analysis [3, 25, 49]. However, due to the discrete nature of texts, we cannot directly exploit the gradient-based approach. In order to deviate the issue, we compute a gradient of the loss function $\mathcal{L}$ with respect to the word embedding $\mathbf{e}_t$, where $\mathbf{e}_t$ is a simple linear projection of a 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 label y^*.

Subsequently, we measure the amount of stimulus of the input tokens toward the model prediction by computing the L_2-norm of $\mathbf{g}_t$, *i.e.,* $||\mathbf{g}_t||_2$. The stimulus is considered as a saliency score of the tokens and they are considered in descending order of the magnitude of $||\mathbf{g}_t||_2$ following [25]. GradMask only considers the top-p portion of the input tokens in $\mathbf{x}$. Specifically, the number

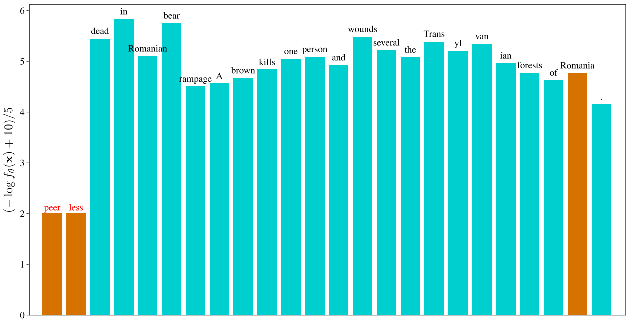


Figure 2: Changes in model confidence (in log scale) on an adversarial example from AGNews dataset [64] with regard to token masking. The orange colored bars are the top-3 tokens detected by the gradient-based attribution. In the log scale, a shorter bar height indicates a larger change in model confidence.

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 to generate a masked input sequence $\mathbf{m} = [x_1, \ldots, m_t, \ldots, x_T]$ with t being the position of a salient token, and m_t is the mask token, [MASK].

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* [10, 14]. Thus, we conjecture that masking the suspicious tokens which are carefully crafted can significantly drop the model confidence. Figure 2 presents a real example showing how model confidence changes in a log scale (shorter bar indicates larger change) with token masking on an adversarial sample. The sample text was adversarially perturbed at the first and second positions (*peer* and *less*). As shown in the chart, the model confidences are significantly dropped by masking those perturbed tokens.

The second assumption is that *NLP models are generally robust to a weak-level of noise as they typically employ regularization methods like dropout [48] and layer normalization [5] during training*. The partial information loss in clean examples due to masking can be offset by the overall context of the input text. As shown in Figure 2, masking clean tokens marginally decreases the model confidence and this observation is further supported by our experiments in Table 6.

The masked sequence $\mathbf{m}$ is then fed into the target model to get a prediction $f_\theta(\mathbf{m})_i$, where $i = c(\mathbf{x})$. We then square the confidence change to assign a stronger penalty to the higher changes. Formally,

$$w = \left(f_\theta(\mathbf{x})_i - f_\theta(\mathbf{m})_i\right)^2 \tag{2}$$

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

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

where τ is a pre-defined threshold. Algorithm 1 presents the overall process of GradMask.

Algorithm 1 GradMask: Gradient-based Masking for Adversarial Example Detection.

```
Require: Input sequence $\mathbf{x} = [x_1, \cdots, x_T]$, target model $f_\theta$, top-$p$ masking portion $p$
Initialize $K = \lfloor T \times p \rfloor$.
Compute $f_\theta(\mathbf{x})_i$, where $i = c(\mathbf{x})$.    ▷ model pred. for x
Get $L := \{||\mathbf{g}_1||, \cdots, ||\mathbf{g}_T||\}$ via Equation (1).
Sort $L$ in descending order.
$\mathbf{m} := \mathbf{x}$
while $k \le K$ do
    $||\mathbf{g}||_t \leftarrow L[k]$
    $m_t \leftarrow$ [MASK]
end while
$w = (f_\theta(\mathbf{x})_i - f_\theta(\mathbf{m})_i)^2$
```

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 GradMask on two conventional NLP tasks: text CLaSsification (CLS) and Natural Language Inference (NLI). For classification, we use the IMDb [29], AG News [64], and Stanford Sentiment Treebank (SST) [47] 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 AG News (AG) dataset contains news articles from more than 2,000 news sources and the samples are categorized into the four largest topic classes. The SST dataset provides movie reviews with fine-grained sentiment labels. We turn them into binary positive/negative labels (SST-2) to follow the setting of FGWS [33].

In NLI, the task is to predict the entailment relationship between a pair of sentences - whether the second sentence (*Hypothesis*) is an *Entailment*, a *Contradiction*, or is *Neutral* with respect to the first one (*Premise*). We use the Multi-Genre NLI (MNLI) dataset [58] for this task. Table 1 gives an overview of the datasets.

Table 1: A summary of the datasets used in our work. M and MM denote matched and mismatched, and P and H represent premise and hypothesis, respectively.

Dataset	Task	Train	Test	# Classes	Avg. Len
IMDb	CLS	25k	25k	2	215
SST-2	CLS	67k	1.8k	2	20
AG	CLS	120k	7.6k	4	43
MNLI	NLI	393*k*	M 9.8*k* MM 9.8*k*	3	P:19.7/H:10.4

3.2 Target Models

We evaluate GradMask on three Transformer-based [53] sequence modeling architectures, which have been widely employed in NLP. We first consider large-scaled pre-trained language models, RoBERTa-base [27] and BERT-base [6], both of which have 124 million parameters. We also evaluate on a relatively smaller model called

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

Model	Dataset	Acc (%)
RoBERTa -base	IMDb	93.32
	AG	94.47
	SST-2	95.50
	MNLI	88.04/87.13 (M/MM)
BERT -base	IMDb	91.75
	AG	94.75
	SST-2	93.47
DistilBERT -base	IMDb	90.78
	AG	94.45
	SST-2	92.20
	MNLI	81.62/81.95(M/MM)

DistilBERT-base [42], which has approximately 40% fewer parameters than RoBERTa-base. Table 2 shows the standard task performance of the target models on each dataset.

The models are optimized by AdamW [28] with a linear adaptive learning rate scheduler. The texts in IMDb are comparatively longer than those in AG and SST-2. For the IMDb classification task the maximum sequence lengths are set to 200.[1] Further details about the model architectures and settings are provided in Appendix A. 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, including widely adopted state-of-the-art synonym substitution-based token-level attacks, as used in previous work [7, 33, 45, 60, 67].

- **BAE** or BERT-based Adversarial Examples [12] is a black-box attack for generating adversarial examples. BAE adopts a pretrained BERT for identifying target tokens. For our experiments, we adopted a replacement-based attack model called BAE-R.
- **A2T** or Attacking to Training [60] perturbs target tokens selected via a gradient-based search algorithm, and their synonyms are generated from a counter-fitted word embedding [34].
- **TextFooler** is a simple token-level black-box attack algorithm proposed by Jin et al. [19]. The target token is identified via cosine similarity between word embeddings of candidate tokens.
- **PWWS** or Probability Weighted Word Saliency [39] is a greedy word substitution-based attack algorithm. The word replacement order is determined by a word saliency score computed based on the change in the model's confidence. The word synonym set is built via the lexical database, WordNet [9].

We generate 1,000 pairs of clean examples and their corresponding adversarial examples for each attack algorithm by sampling from each test dataset. However, for attack algorithms with a lower attack success rate (ASR) such as BAE-R and A2T (*c.f.,*Table 3), we use the entire test dataset in order to maximise the number of adversarial examples generated. The number of samples and the ASR for different attacks are provided in Table 3. All attacks are implemented by using the publicly available TextAttack library [32], which has been widely used in NLP robustness research [11, 13, 60].

3.4 Evaluation Metrics

We now evaluate the performance of GradMask and compare it to the state-of-the-art. FGWS [33] was mainly evaluated via F1 score, but we follow the standards from the out-of-distribution (OOD) sample detection literature [16, 36, 65] for a thorough and better understanding of the adversarial example detection methods.

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

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

where the superscripts + and − denote the positive and the negative classes, respectively, and $\mathcal{I}$ is the indicator function as defined in Equation (3) Based on these two rates, we evaluate the detection methods with the following evaluation metrics:

- **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 the receiver operating characteristic curve (ROC curve).
- **EER** or Equal Error Rate refers to a point where the FPR equals the false negative rate (FNR). It can be computed as 1−TPR in a ROC curve. Even though AUROC provides the overall performance in a varying threshold setting, detection algorithms would be employed in their optimal threshold setting. EER can be used to summarize the performance of the detection algorithm. A lower EER value indicates better performance of the system.
- **FPR95** refers to the FPR at 95% TPR. FPR95 quantifies how many clean examples have to be rejected to detect 95% of the adversarial examples. FPR95 is a very important metric for evaluating detection algorithms [1]. A lower FPR95 score is often strongly required for systems that require a high level of system safety or security.

4 RESULTS AND ANALYSIS

We first evaluate GradMask on widely employed NLP datasets and compare it with the baselines (Sections 4.1, 4.2, and 4.3). Then, we analyze the adversarially perturbed token detection performance of GradMask (Section 4.4). Subsequently, we investigate GradMask's potential against a non-synonym based (character-level) attack (Section 4.5). Finally, we conduct ablation studies to investigate the effectiveness of GradMask (Section 4.6). We also provide additional experimental results and analysis in the Appendix.

4.1 Detection for Text Classification Tasks

For adversarial example detection, we compare the performance of GradMask with that of FGWS [33], which is the state-of-the-art textual adversarial detection algorithm.[2]

As shown in Table 3, GradMask shows better AUROC, FPR95, and EER scores in most of the tasks. Particularly, it significantly

[1] This is mainly to enable a fair comparison with FGWS. FGWS has to compute a huge cosine similarity matrix between words, which causes a memory issue without a tight setting for the maximum sequence length.

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

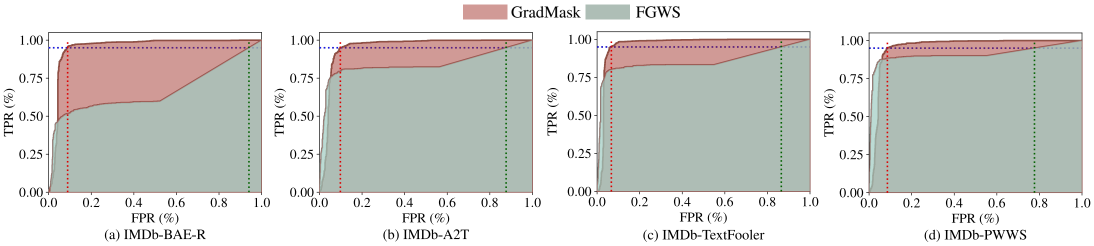
Table 3: Adversarial example detection results of FGWS and GradMask (GM). ASR denotes the attack success rate of an attack algorithm; higher ASR indicates a stronger attack. Higher (↑) AUROC, lower (↓) EER and FPR95 indicates a better detection algorithm. K denotes the number of masked tokens in an input text selected by GradMask .

Model	Dataset	Attack	# Samples		ASR (%)	AUROC (%) ↑		EER (%) ↓		FPR95 (%) ↓		K
			TN	TP		FGWS	GM	FGWS	GM	FGWS	GM	
RoBERTa-base	IMDb	BAE-R	1000	1000	63.45	66.56	**95.15**	40.51	**7.35**	94.05	**8.90**	3
		A2T	1000	1000	52.25	84.04	**95.05**	19.11	**8.30**	87.66	**9.80**	1
		TextFooler	1000	1000	82.44	85.40	**96.40**	17.20	**5.60**	86.52	**6.70**	1
		PWWS	1000	1000	87.41	90.92	**95.43**	12.03	**7.35**	77.68	**8.60**	3
	AG	BAE-R	1000	1000	15.75	62.59	**83.82**	44.95	**19.15**	94.15	**35.20**	3
		A2T	937	937	13.04	75.09	**83.49**	27.80	**20.38**	90.26	**40.02**	2
		TextFooler	1000	1000	84.89	89.68	**96.53**	12.30	**5.35**	79.53	**5.60**	3
		PWWS	1000	1000	65.96	94.74	**95.69**	**6.46**	7.70	50.80	**9.30**	3
	SST-2	BAE-R	1000	1000	58.17	60.08	**79.40**	43.80	**23.30**	94.33	**61.70**	3
		A2T	349	349	20.07	65.57	**78.16**	33.95	**23.07**	93.14	**52.44**	1
		TextFooler	1000	1000	93.28	74.14	**84.82**	29.05	**17.10**	91.66	**35.40**	3
		PWWS	1000	1000	85.18	85.25	**85.49**	**16.76**	19.62	82.11	**38.50**	1
BERT-base	IMDb	BAE-R	1000	1000	54.98	65.56	**92.27**	40.79	**9.60**	94.02	**11.40**	3
		A2T	1000	1000	52.63	86.14	**91.51**	16.80	**9.85**	84.01	**12.90**	3
		TextFooler	1000	1000	98.72	85.46	**95.82**	16.85	**5.30**	85.36	**5.50**	3
		PWWS	1000	1000	96.90	89.49	**95.93**	13.19	**6.70**	78.21	**8.20**	3
	AG	BAE-R	895	895	12.43	58.12	**75.33**	47.04	**31.06**	94.51	**62.57**	1
		A2T	717	717	9.96	70.49	**75.59**	31.03	**30.13**	91.15	**65.83**	1
		TextFooler	1000	1000	86.96	88.18	**95.60**	15.10	**8.20**	80.21	**11.00**	3
		PWWS	1000	1000	66.01	93.45	**95.00**	**8.30**	9.65	60.39	**15.10**	3
	SST-2	BAE-R	1000	1000	59.10	60.52	**80.04**	43.15	**24.65**	93.98	**58.00**	3
		A2T	415	415	24.38	68.56	**73.45**	31.81	**30.12**	92.22	**54.94**	1
		TextFooler	1000	1000	96.34	75.93	**83.04**	25.95	**19.70**	90.24	**37.70**	3
		PWWS	1000	1000	88.81	**84.53**	83.91	**16.85**	20.30	83.06	**42.40**	3
DistilBERT-base	IMDb	BAE-R	1000	1000	72.89	66.41	**92.46**	40.97	**12.70**	94.01	**17.10**	1
		A2T	1000	1000	68.40	87.73	**91.56**	14.44	**12.85**	80.85	**18.70**	3
		TextFooler	1000	1000	93.02	87.96	**94.03**	14.97	**9.95**	81.81	**12.40**	1
		PWWS	1000	1000	99.60	91.87	**94.02**	11.09	**10.70**	72.17	**16.00**	3
	AG	BAE-R	1000	1000	14.31	59.98	**78.45**	46.10	**26.5**5	94.25	**57.90**	1
		A2T	861	861	11.99	73.43	**78.43**	29.91	**27.1**2	90.42	**55.75**	1
		TextFooler	1000	1000	89.53	90.33	**95.54**	12.65	**7.90**	76.97	**10.0**	3
		PWWS	1000	1000	73.64	**95.31**	94.70	**6.45**	9.70	46.79	**13.90**	3
	SST-2	BAE-R	1000	1000	62.34	62.88	**78.18**	40.90	**26.15**	93.69	**61.00**	3
		A2T	502	502	29.90	72.87	**74.42**	**27.39**	30.38	90.63	**47.01**	3
		TextFooler	1000	1000	96.90	74.65	**80.80**	28.55	**22.25**	91.13	**45.50**	3
		PWWS	1000	1000	89.13	**85.58**	82.07	**15.45**	22.25	81.45	**45.50**	3

outperforms FGWS for all the models (RoBERTa, BERT-Base and DistilBERT) in terms of the FPR95 score, which is an important metric for systems with high security requirements. In addition, it achieves notably better AUROC and EER scores in most of the experiment scenarios. This tendency is well presented in Figure 3, which shows ROC curves of FGWS and GradMask for the RoBERTa model trained on IMDb. The ROC curves of FGWS tend to increase steeply and remain stable. However, as TPR increases, the FPR of FGWS tends to proportionally increase after some point. Especially for BAE-R (Figure 3 (a)), the tendency of the FPR increase is quite steep. In contrast, GradMask tends to reach 95% TPR at a considerably lower FPR and has larger AUROC scores.

Another interesting observation is that there is a proportional relationship between the adversarial robustness of the model and the AUROC of GradMask. For example, an average ASR of four adversarial attacks on RoBERTa, BERT, and DistilBERT models fine-tuned on IMDb are 71.39%, 75.81%, and 83.48%, respectively. The lower average ASR indicates that the model is more robust to adversarial attacks. The average AUROC scores of these models are 95.51%, 93.88%, and 93.02%, individually. Similarly, we observe the same trend in EER of GradMask. We investigate this aspect further in Appendix B.3 by evaluating GradMask's performance with an adversarially trained [14] robust model, where we fine-tune a DistilBERT model on adversarial examples crafted by TextFooler.

We also observe that FGWS tends to underperform against BAE-R, A2T, and TextFooler attacks while it shows comparable scores for identifying PWWS attacks in some setups. A possible explanation may be related to the nature of the synonym search strategy. FGWS identifies synonyms of infrequent words in input texts via WordNet, which is also adopted in PWWS to build synonym sets. Thus, FGWS may face some difficulty in finding a synonym set for a perturbation generated by attack algorithms that do not use WordNet. In contrast, our proposed method GradMask, which minimizes the assumptions about potential attack algorithms, generally shows better performance for all evaluation setups.



(a) IMDb-BAE-R (b) IMDb-A2T (c) IMDb-TextFooler (d) IMDb-PWWS

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

Table 4: Adversarial example detection results of FGWS and GradMask(GM) on the MNLI dataset.

Model	Dataset	Attack	# Samples		ASR (%)	AUROC (%)↑		EER (%) ↓		FPR95 (%)↓		K
			TN	TP		FGWS	GM	FGWS	GM	FGWS	GM	
RoBERTa-base	MNLI-M	BAE-R	1000	1000	64.23	52.77	**69.99**	50.96	**33.80**	95.17	**73.50**	1
		A2T	1000	1000	49.85	66.34	**69.92**	37.96	**33.95**	92.82	**65.50**	1
		TextFooler	1000	1000	91.41	70.25	**74.24**	34.35	**29.50**	92.00	**55.40**	1
		PWWS	1000	1000	83.06	**76.94**	74.15	**27.38**	31.05	88.88	**65.47**	1
	MNLI-MM	BAE-R	1000	1000	64.60	55.13	**73.84**	48.73	**31.25**	94.76	**65.80**	1
		A2T	1000	1000	49.50	66.53	**74.69**	35.71	**29.80**	92.97	**58.70**	1
		TextFooler	1000	1000	91.91	70.37	**75.65**	33.55	**27.85**	92.04	**55.60**	1
		PWWS	1000	1000	83.54	**77.94**	77.78	**26.05**	27.63	88.55	**58.47**	1

Finally, GradMask and FGWS tend to show consistently better performance in detecting strong attacks such as TextFooler and PWWS 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.

4.2 Detection for Natural Language Inference

We further compare GradMask with FGWS on an NLI task using the MNLI dataset. As shown in Table 4, the overall performance of GradMaskis significantly better than that of FGWS except for the PWWS attack scenario, which is consistent with the experimental results on the classification tasks presented in Section 4.1. One interesting observation is that GradMask tends to show better performance across the evaluation metrics on the MNLI-MM dataset than on the MNLI-M dataset despite the lower clean example prediction accuracy of the model on MNLI-MM dataset (M: 88.04, MM: 87.13). In contrast, FGWS shows no distinctive differences in performance between the MNLI-M and MNLI-MM datasets.

4.3 Comparison with Anomaly Detection Methods

We conducted additional experiments to compare GradMask with popular anomaly detection algorithms such as Maximum Softmax Probability (MSP) [15] and One-Class Support Vector Machine with linear kernel (OCSVM) [43] that are widely adopted as baselines in various anomaly detection problems [1, 23, 44, 59].

From the results in Table 5, we notice that GradMask significantly outperforms the baselines by a large margin. These results are consistent with the results reported in Section 4.1. GradMask achieves significantly lower FPR95 and EER scores than those of MSP and OCSVM for all the datasets. Additionally, OCSVM achieves the best performance with a linear kernel and the results are similar to that of MSP. This can be attributed to the linear property of model prediction distributions.

Table 5: Comparison with anomaly detection methods (MSP and OCSVM) for PWWS attack detection results on RoBERTa.

Dataset	Method	AUROC (%) ↑	EER (%) ↓	FPR95 (%) ↓
IMDb	MSP	92.23	14.50	34.43
	OCSVM	92.23	14.50	34.43
	GM	**95.43**	**7.35**	**8.60**
AG	MSP	94.44	12.07	24.60
	OCSVM	94.44	12.07	24.60
	GM	**95.69**	**7.70**	**9.30**
SST-2	MSP	**87.86**	**18.51**	58.90
	OCSVM	87.86	18.51	58.90
	GM	85.49	19.62	**38.50**

We further analyze the statistics of the features extracted from MSP and GradMask methods. Table 6 presents two statistics of the extracted features, mean (Avg) and standard deviation (Std). The values are averaged over 1,000 samples. As shown in the table, the overall mean differences between the w (*c.f.,* Equation (2)) of adversarial examples (w-A) and w of clean examples (w-C) are higher than that of MSP, which implies that GradMask feature w is more distinguishable. Specifically, for IMDb, MSP shows 43.3 (= 92.88−49.58), but GradMask shows 55.47 (59.75−4.28) at $K = 3$.

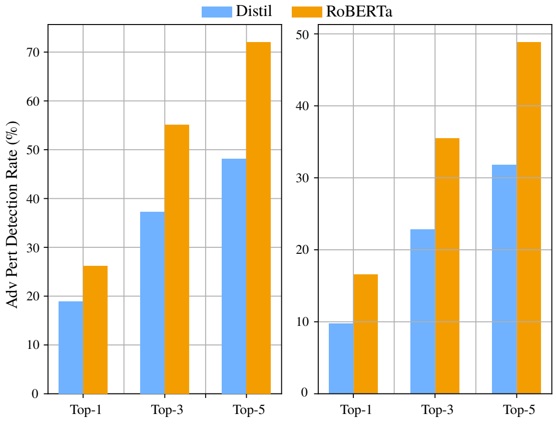
4.4 Adversarial Token Detection

We now analyze how GradMask attributes the model prediction on adversarial examples at the token level. Figure 4 shows perturbed token detection rates for DistilBERT and RoBERTa on two datasets,

Table 6: Statistics (Avg and Std) of extracted features. The first row of each dataset denotes the maximum softmax probability (MSP) of the RoBERTa model for adversarial (Conf-A) and clean (Conf-C) samples, respectively. The subsequent rows show the mean and standard deviation of w of GradMask while varying the number of mask tokens K.

Dataset	K	w-A/Conf-A (Avg±Std)	w-C/Conf-C (Avg±Std)
IMDb	MSP	-/49.58±49.67	-/92.88±13.53
	1	32.48±29.39/-	2.81±12.03/-
	2	53.71±36.92/-	3.84±18.04/-
	3	59.75±34.53/-	4.28±18.85/-
AG	MSP	-/49.43±49.55	-/89.75±15.58
	1	25.11±24.04/-	2.09±11.01/-
	2	47.18±31.39/-	3.32±16.03/-
	3	50.84±30.18/-	3.77±16.79/-

Figure 4: Adversarially perturbed token detection rates at top-1, top-2 and top-5 for GradMask.



(a) IMDb (b) AGNews

IMDb and AG. 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-K values of w in Equation (2).

In case of DistilBERT, we notice that GradMask shows 48.17% and 31.82% detection rates for IMDb and AG within the top-5 predictions, respectively. On the other hand, for RoBERTa, it shows 72.04% and 48.85% detection rates for IMDb and AG within the top-5 predictions.

4.5 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 [38] from the TextAttack library [32]. Even though character-level attacks are known to be relatively simple to defend at a preprocessing stage with a spell or a grammar checker [38], our motivation for this experiment is to demonstrate the potential of GradMask against non-synonym based attacks.

We first generated 400 pairs of clean examples and their corresponding adversarial examples from AG against RoBERTa-base and Distilbase, respectively. We compare GradMask with the MSP algorithm [15] in Table 7. We see that our method shows promising results with AUROC scores of 85.58% and 75.80% for RoBERTa-base and DistilBERT-base, respectively.

Table 7: Detection results against a character-level attack.

Dataset	Model	AUROC (%) ↑		EER (%) ↓		FPR95 (%) ↓		K
		MSP	GM	MSP	GM	MSP	GM	
AG	RoBERTa	65.76	**85.58**	38.38	**18.38**	81.00	**58.75**	3
	DistilBERT	70.66	**75.80**	33.50	**29.50**	76.00	**72.25**	1

4.6 Ablation Study

To understand GradMask better, we conduct two ablation studies:

***(i) Impact of square operation in Equation* (2).** As mentioned, the main rationale for using the square operation in Equation (2) is to make changes in confidence caused by the masking operation more distinctive. We compare the detection performance of the *squared* model confidence change w to the *without-squared* confidence change, $\sqrt{w}$ on three datasets with the RoBERTa-base model. As shown in Table 8, the square operation significantly improves the detection performance for all evaluation measures. On average, AUROC and EER scores are improved 21.28% and 14.06%, respectively. Especially, FPR95 scores are remarkably dropped by 77.57% on average. In the case of AG, the standard deviation of $\sqrt{w}$ of adversarial example is 44.21% but that of w is 31.78%.

(ii) Effectiveness of the gradient-based masking strategy. We compare our gradient-based search strategy against a *brute-force* search (BF), which identifies the best masking position by masking a token in an input text one at a time. Each masked sequence $\mathbf{m}_t$ with a masked token at position t is then fed into the target model to get a prediction $f_\theta(\mathbf{m}_t)_i$, where $i = c(\mathbf{x})$. This process gives T such confidence scores and the maximum confidence change caused by masking a token at t' is considered as the best masking position.

Based on the assumption that masking an adversarial token in an adversarial example tends to drop the model confidence significantly, the masking position t' identified by the BF search can be considered as the optimal masking position. However, as shown in Table 9, GradMask significantly outperforms BF in all evaluation metrics. Specifically, we observe that BF search is too aggressive in that it changes a model's prediction on a clean example too steeply. In the case of GradMask for AG, the average squared confidence change for clean examples (*i.e.,* w-**C** as defined in Section 4.3) is around 2.8, while the average w-**C** is 40.97 for BF. Thus, this study shows the effectiveness of GradMask in identifying masking positions. Note that this study was performed with randomly sampled 100 pairs of clean examples and the corresponding adversarial examples due to the computational cost of BF search.

5 CONCLUSION

In this paper, we have proposed a simple adversarial example detection scheme, GradMask, which utilizes gradient signals as a guidance to detect adversarially perturbed tokens. This guidance

Table 8: Ablation study of the square operation in Equation (2).

Dataset	Feature	↑ AUROC(%)	↓ EER(%)	↓ FPR95(%)	K
IMDb	$\sqrt{w}$	71.24	26.35	100	1
	w	**95.45**	**8.80**	**14.30**	1
AG	$\sqrt{w}$	77.35	20.95	99.70	1
	w	**95.28**	**9.30**	**13.90**	1
SST-2	$\sqrt{w}$	63.80	32.60	99.70	1
	w	**85.49**	**19.62**	**38.50**	1

Table 9: Comparison of masking candidate search methods.

Dataset	Search	AUROC ↑	EER ↓	FPR95 ↓	CPU Time ↓
	Method	(%)	(%)	(%)	sec
IMDb	BF	62.41	44.00	73.00	581.22
	GM-K = 1	96.08	8.00	12.00	14.60
AG	BF	73.73	29.50	51.00	117.68
	GM-K = 1	94.01	10.00	17.00	16.03

additionally provides a weak interpretation about its decision by informing us which tokens are critical to a model's prediction. The experimental results show that GradMask is a promising approach as an adversarial attack detection algorithm for NLP 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. In conclusion, our detection strategy can serve as a useful tool for identifying adversarial attacks for protecting the text classification systems.

6 ACKNOWLEDGEMENTS

This work is partly supported by SIMTech-NTU Collaborative Research Programme on Complex Systems and IAF-PP Industry Alignment Fund (Grant No: M21J6a0080). We gratefully acknowledge our anonymous reviewers for their helpful comments. We would also like to thank Anh-Duc Nguyen for his valuable input.

REFERENCES

[1] Ahmed Aldahdooh, Wassim Hamidouche, Sid Ahmed Fezza, and Olivier Déforges. 2021. Adversarial Example Detection for DNN Models: A Review.

[2] 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*. Association for Computational Linguistics, Brussels, Belgium, 2890–2896.

[3] 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*.

[4] 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*.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. arXiv:1607.06450 [stat.ML]

[6] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186.

[7] Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards Robustness Against Natural Language Word Substitutions. In *International Conference on Learning Representations*.

[8] 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)*. Association for Computational Linguistics, Melbourne, Australia, 31–36.

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

[10] Angus Galloway, Graham W. Taylor, and Medhat Moussa. 2018. Attacking Binarized Neural Networks. In *International Conference on Learning Representations*.

[11] Yujian Gan, Xinyun Chen, Qiuping Huang, Matthew Purver, John R. Woodward, Jinxia Xie, and Pengsheng Huang. 2021. Towards Robustness of Text-to-SQL Models against Synonym Substitution. 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)*. Association for Computational Linguistics, Online, 2505–2515.

[12] 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)*. Association for Computational Linguistics, Online, 6174–6181.

[13] Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness Gym: Unifying the NLP Evaluation Landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*. Association for Computational Linguistics, Online, 42–55.

[14] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

[15] Dan Hendrycks and Kevin Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *Proceedings of International Conference on Learning Representations* (2017).

[16] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. 2019. Deep Anomaly Detection with Outlier Exposure. In *International Conference on Learning Representations*.

[17] 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*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.

[18] 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)*. Association for Computational Linguistics, Hong Kong, China, 4129–4142.

[19] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT Really Robust? Natural Language Attack on Text Classification and Entailment. *arXiv preprint arXiv:1907.11932* (2019).

[20] 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.

[21] 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*.

[22] Thai Le, Noseong Park, and Dongwon Lee. 2021. A Sweet Rabbit Hole by DARCY: Using Honeypots to Detect Universal Trigger's Adversarial Attacks. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL'2021)* (2021).

[23] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 7167–7177.

[24] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 552–561.

[25] 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.

[26] Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Understanding Neural Networks through Representation Erasure.

[27] 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* (2019).

[28] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[29] 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*. Association for Computational Linguistics, Portland, Oregon, USA, 142–150.

[30] 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.

[31] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial Training Methods for Semi-supervised Text Classification. In *International Conference on Learning Representations*.

[32] 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.

[33] 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*. Association for Computational Linguistics, Online, 171–186.

[34] Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting Word Vectors to Linguistic Constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 142–148.

[35] 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*.

[36] 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*.

[37] Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, Ann Arbor, Michigan, 115–124.

[38] 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*, Vol. abs/1905.11268.

[39] 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*. Association for Computational Linguistics, Florence, Italy, 1085–1097.

[40] Jia Robin. 2020. *Building robust natural language processing systems*. Ph.D. Dissertation. Stanford University, Stanford, California.

[41] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *arXiv preprint arXiv:1907.10641* (2019).

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

[43] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 2000. Support Vector Method for Novelty Detection. In *Advances in Neural Information Processing Systems*, Vol. 12. MIT Press.

[44] Alireza Shafaei, Mark Schmidt, and James J. Little. 2019. A Less Biased Evaluation of Out-of-distribution Sample Detectors.. In *BMVC*. 3.

[45] 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 (2020).

[46] 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*.

[47] 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*. Association for Computational Linguistics, Seattle, Washington, USA, 1631–1642.

[48] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15, 56 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

[49] 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* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 3319–3328.

[50] 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*.

[51] 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)*. ACL, Bangkok, Thailand, 4153–4169.

[52] 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*. Association for Computational Linguistics, Online, 2920–2935.

[53] 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*, Vol. 30. Curran Associates, Inc., 5998–6008.

[54] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. In *Empirical Methods in Natural Language Processing*.

[55] 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*. Association for Computational Linguistics, Online, 1102–1112.

[56] Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural Language Adversarial Attacks and Defenses in Word Level. *CoRR* (2019).

[57] 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 (2020).

[58] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana). Association for Computational Linguistics, 1112–1122.

[59] Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, Taylan Cemgil, S. M. Ali Eslami, and Olaf Ronneberger. 2020. Contrastive Training for Improved Out-of-Distribution Detection.

[60] Yoo Jin Yong and Qi Yanjun. 2021. Towards Improving Adversarial Training of NLP Models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 945–956.

[61] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *Computer Vision – ECCV 2014*. Springer International Publishing, Cham, 818–833.

[62] Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial Attacks on Deep-Learning Models in Natural Language Processing: A Survey. *ACM Trans. Intell. Syst. Technol.* 11, 3 (apr 2020).

[63] 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.* (2020).

[64] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, Vol. 28. Curran Associates, Inc., 649–657.

[65] Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain Detection for Natural Language Understanding in Dialog Systems.

[66] 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)*. Association for Computational Linguistics, Hong Kong, China, 4904–4913.

[67] 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)*. Association for Computational Linguistics, Online, 5482–5492.

[68] 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*.

A MODEL PARAMETERS

Table 10 summarizes the parameter settings of the target models used for adversarial example detection experiments.

Table 10: Parameter settings of target models. AL and MaxLen denote the adaptive linear learning rate scheduler and maximum sequence length, respectively.

Model	Parameters	
RoBERTa -base	Optimizer	AdamW
	Batch Size (IMDb)	24(16)
	Epochs	10
	LearningRate	10^-5
	LearningRate Scheduler	AL
	MaxLen	200
BERT -base	Optimizer	AdamW
	Batch Size (IMDb)	24(16)
	Epochs	10
	LearningRate	10^-5
	LearningRate Scheduler	AL
	MaxLen	200
DistilBERT -base	Optimizer	AdamW
	Batch Size (IMDb)	24(16)
	Epochs	10
	LearningRate	10^-5
	LearningRate Scheduler	AL
	MaxLen	200

B SUPPLEMENTARY EXPERIMENTS

This section provides a supplementary analysis of GradMask for a better understanding of the algorithm. We first investigate the relationship between the multi-masking effect and detection performance of GradMask in Appendix B.1. We then study the performance of GradMask for a task that is sensitive to a single critical token (Appendix B.2). Subsequently, we conduct an experiment to analyze a performance of GradMask with adversarially robust model (Appendix B.3). Finally, we discuss about the word frequency assumption and adversarial robustness (Appendix B.4).

During the experiments, we used a RoBERTa-base model for each task and generated adversarial examples via PWWS attack with TextAttack library [32] for Appendix B.1 and Appendix B.2. For each task, 1,000 clean examples and their corresponding 1,000 adversarial examples are used as a test set.

B.1 GradMask with Multi-Masking.

We investigate the impact of the masking strategy of GradMask on its detection performance. Table 11 summarizes the experiment results on three different datasets. As shown in the table, the overall EER and FPR95 scores tend to improve as the number of masks in input texts increases but AUROC decreases for some datasets. One of the possible explanations is that the increased number of masked tokens may discard some critical information of input texts and the information loss decreases the detection performance at some operational settings.

Table 11: Adversarial example detection results of GradMask with the multi-masking strategy.

Model	Dataset	K	AUROC (%) ↑	EER (%) ↓	FPR95 (%) ↓
RoBERTa	IMDb	1	**95.45**	8.80	14.30
		2	95.32	7.50	10.60
		3	95.43	**7.35**	**8.60**
	AG	1	95.28	9.30	13.90
		2	95.44	8.00	12.60
		3	**95.69**	**7.70**	**9.30**
	SST-2	1	**85.49**	19.62	**38.50**
		2	84.17	20.10	43.30
		3	84.11	**19.35**	40.90
BERT	IMDb	1	95.47	8.65	15.30
		2	95.82	**6.60**	9.30
		3	**95.93**	6.70	**8.20**
	AG	1	**95.09**	10.35	16.70
		2	94.77	10.70	15.30
		3	95.00	**9.65**	**15.10**
	SST-2	1	83.83	22.00	51.30
		2	83.72	21.10	45.80
		3	**83.91**	**20.30**	**42.40**

Table 12: Adversarial example detection results of RoBERTa-Large model for WG dataset. Acc denotes detection accuracy.

Dataset	Method	AUROC (%)	FPR95 (%)	Acc (%)
WG	MSP	52.45	93.95	53.73
	OCSVM	55.15	92.83	54.62
	GM	**60.78**	**91.37**	**60.34**

B.2 Detection of Adversarial Attack in Winograd Schema Challenge.

One of the potential criticisms of masking-based textual adversarial example detection approaches is the information loss caused by their masking strategies. It is likely that the gradient-based token saliency evaluation approach may decide to mask a critical token that is important for a model's prediction and drop the confidence of the model prediction.

However, as shown in Table 6, model confidence changes for clean examples are not significant in most cases. A possible explanation is that the models are able to capture sufficient contexts from neighboring texts. Nevertheless, we further investigate this possible issue on a task that relies on a few critical tokens. To this end, we investigate the proposed method on the Winograd Schema Challenge [24]. The Winograd Schema Challenge (WSC) is a benchmark for commonsense reasoning and natural language understanding. The Winograd schema consists of a pair of sentences differing in one or two words with a highly ambiguous pronoun that is difficult to solve for statistical models.

One of WSC benchmark datasets is WinoGrande (WG) dataset [41]. WG dataset is split into 40k training samples and 1.2k validation samples. We first trained a RoBERTa-Large model on the training set and our best model achieves an accuracy of 72% against the validation set. Again, we sampled 1,000 clean examples from the validation set and generated 1,000 adversarial examples via PWWS attack.

As shown in Table 12, GradMask achieves the best performances for all evaluation metrics. However, its scores are significantly lower than those of other tasks such as IMDb and AG. We conjecture that the overall performances of GradMask can be improved further as the model's standard performance increases because GradMask relies on the standard task performance of models for extracting better features.

B.3 GradMask with Adversarially Robust Model

We investigate the detection performance of GradMask with an adversarially robust model. For this, we train a Distil-BERT-base via adversarial training (AT) [14] to improve the adversarial robustness of the model. We first sample 1,000, 3,000 and 3,000 adversarial examples (AE) via TextFooler from the Movie Review (MR)[3] [37], SST-2, and AG datasets' training sets, respectively. The sampled examples are then used to fine-tune the models and the trained models are attacked again by TextFooler. The post-attack accuracy (A-Acc) of the trained models are summarized in Table 13.

Table 13: Comparison of adversarial example detection results of GradMask on an adversarially robust model (Distil-BERT-base trained with adversarial examples). For example, MR-AE means Distil-BERT-base is trained with adversarial examples of MR dataset. C-Acc and A-Acc refer to clean testset accuracy and post-attack accuracy, respectively.

Dataset	↑ C-Acc	↑ A-Acc	↑ AUROC(%)	↓ EER(%)	↓ FPR95(%)	K
MR	83.20	2.85	53.86	45.49	96.75	2
MR-AE	79.74	8.91	74.09	31.13	67.55	2
AG	94.45	9.90	95.54	7.90	7.90	3
AG-AE	93.70	16.76	95.31	8.55	11.00	3
SST-2	92.20	2.85	80.80	22.25	22.25	1
SST-2-AE	83.41	6.14	34.39	60.65	97.10	1

The models are evaluated using 1,000 pairs of clean examples of each testset and their corresponding adversarial examples. As shown in the table, AT tends to improve A-Acc but it typically hurts the clean testset accuracy (C-Acc). Particularly, the C-Acc of the model trained via adversarial training with adversarial examples of SST-2 (SST-2-AE) drops significantly and the overall detection performance of GradMask declines. However, the models with AG-AE and MR-AE which show marginal C-Acc performance drop tend to reach or outperform the performance of the models trained on clean datasets.

B.4 Discussion on Word Frequency and Adversarial Robustness

According to Mozes et al. [33], 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 [17]. 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 by masking those tokens. We first evaluated the model performance to observe how the vocabulary constraint affects the model performance. As shown in Table 14, 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 [39] against the word frequency constrained models. 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 [33], the models trained on V' are expected to be robust against adversarial attacks. However, from the results in Table 14, we notice that they showed significant brittleness against adversarial attacks. For instance, 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.*

Table 14: 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'.

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

[3] MR dataset consists of movie reviews labels with positive or negative sentiments.