Natural Language Adversarial Attack and Defense in Word Level

Xiaosen Wang,[*] **Hao Jin,**[*] **Kun He**[†]
School of Computer Science and Technology
Huazhong University of Science and Technology
WuHan, 430074, China
`{xiaosen,mailtojinhao,brooklet60}@hust.edu.cn`

Abstract

In recent years, inspired by a mass of researches on adversarial examples for computer vision, there has been a growing interest in designing adversarial attacks for Natural Language Processing (NLP) tasks, followed by very few works of adversarial defenses for NLP. To our knowledge, there exists no defense method against the successful synonym substitution based attacks that aim to satisfy all the lexical, grammatical, semantic constraints and thus are hard to be perceived by humans. We contribute to fill this gap and propose a novel adversarial defense method called *Synonym Encoding Method* (SEM), which inserts an encoder before the input layer of the model and then trains the model to eliminate adversarial perturbations. Extensive experiments demonstrate that SEM can efficiently defend current best synonym substitution based adversarial attacks with little decay on the accuracy for benign examples. To better evaluate SEM, we also design a strong attack method called Improved Genetic Algorithm (IGA) that adopts the genetic metaheuristic for synonym substitution based attacks. Compared with the first genetic based adversarial attack proposed in 2018, IGA can achieve higher attack success rate with lower word substitution rate, at the same time maintain the transferability of adversarial examples.

1 Introduction

Deep Neural Networks (DNNs) have made great success in various machine learning tasks, such as computer vision (Krizhevsky et al., 2012; He et al., 2016) and Natural Language Processing (NLP) (Kim, 2014; Lai et al., 2015; Devlin et al., 2018). However, recent studies have discovered that DNNs are vulnerable to *adversarial examples*

---

[*] The first two authors contributed equally.
[†] Corresponding author.

not only for computer vision tasks (Szegedy et al., 2014) but also for NLP tasks (Papernot et al., 2016), causing a serious threat to their safe applications. For instance, spammers can evade spam filtering system with adversarial examples of spam emails while preserving the intended meaning.

In contrast to numerous methods proposed for adversarial attacks (Goodfellow et al., 2015; Carlini and Wagner, 2017; Athalye et al., 2018) and defenses (Goodfellow et al., 2015; Guo et al., 2018; Song et al., 2019) in computer vision, there are only a few list of works in the area of NLP, inspired by the works for images and emerging very recently in the last two years (Zhang et al., 2019). This is mainly because existing perturbation-based methods for images cannot be directly applied to texts due to their discrete property in nature. Furthermore, if we want the perturbation to be barely perceptible by humans, it should satisfy the lexical, grammatical and semantic constraints in texts, making it even harder to generate text adversarial examples.

Current attacks in NLP can fall into four categories, namely modifying the characters of a word (Liang et al., 2017; Ebrahimi et al., 2017), adding or removing words (Liang et al., 2017), replacing words arbitrarily (Papernot et al., 2016), or substituting words with synonyms (Alzantot et al., 2018; Ren et al., 2019). However, the first three categories are easy to be detected and defended by spell or syntax check (Rodriguez and Rojas-Galeano, 2018; Pruthi et al., 2019). As synonym substitution aims to satisfy all the lexical, grammatical and semantic constraints, it is hard to be detected by automatic spell or syntax check as well as human investigation. To our knowledge, currently there is no defense method specifically designed against the synonym substitution based attacks.

In this work, we postulate that the model generalization leads to the existence of adversarial ex-

amples: a generalization that is not strong enough causes the problem that there usually exists some neighbors x' of a benign example x in the manifold with a different classification. Based on this hypothesis, we propose a novel defense mechanism called *Synonym Encoding Method* (SEM) that encodes all the synonyms to a unique code so as to force all the neighbors of x to have the same label of x. Specifically, we first cluster the synonyms according to the *Euclidean Distance* in the embedding space to construct the encoder. Then we insert the encoder before the input layer of the deep model without modifying its architecture, and train the model again to defend adversarial attacks. In this way, we can defend the synonym substitution based adversarial attacks effectively in the context of text classification.

Extensive experiments on three popular datasets demonstrate that the proposed SEM can effectively defend adversarial attacks, while maintaining the efficiency and achieving roughly the same accuracy on benign data as the original model does. To our knowledge, SEM is the first proposed method that can effectively defend the synonym substitution based adversarial attacks.

Besides, to demonstrate the efficacy of SEM, we also propose a genetic based attack method, called *Improved Genetic Algorithm* (IGA), which is well-designed and more effective as compared with the first proposed genetic based attack algorithm, GA (Alzantot et al., 2018). Experiments show that IGA can degrade the classification accuracy more significantly with lower word substitution rate than GA. Meanwhile, IGA keeps the transferability of adversarial examples as GA does.

2 Background and Related Work

Let W denote the word set containing all the legal words. Let $x = \{w_1, \ldots, w_i, \ldots, w_n\}$ denote an input text, $\mathcal{C}$ the corpus that contains all the possible input texts, and $\mathcal{Y} \in \mathbb{N}^K$ the output space where K is the dimension of $\mathcal{Y}$. The classifier $f : \mathcal{C} \to \mathcal{Y}$ takes an input x and predicts its label $f(x)$, and let $S_m(x, y)$ denote the confidence value for the y-th category at the softmax layer. Let $Syn(w, \sigma, k)$ represent the set of first k synonyms of w within distance σ, namely

$$Syn(w, \sigma, k) = \{\hat{w}_1, \ldots, \hat{w}_i, \ldots, \hat{w}_k | \hat{w}_i \in W \wedge \|w - \hat{w}_1\|_p \leq \ldots \leq \|w - \hat{w}_k\|_p < \sigma\}, \quad (1)$$

where $\|w - \hat{w}\|_p$ is the p-norm distance evaluated on the corresponding embedding vectors.

2.1 Natural Language Adversarial Examples

Suppose we have an ideal classifier $c : \mathcal{C} \to \mathcal{Y}$ that could always output the correct label for any input text x. For a subset of (train or test) texts $\mathcal{T} \subseteq \mathcal{C}$ and a small constant ϵ, we could define the natural language adversarial examples as follows:

$$\mathcal{A} = \{x_{adv} \in \mathcal{C} \mid \exists x \in \mathcal{T}, f(x_{adv}) \neq c(x_{adv})$$
$$= c(x) = f(x) \wedge d(x - x_{adv}) < \epsilon\}, \quad (2)$$

where $d(x - x_{adv})$ is a distance metric to evaluate the dissimilarity between the benign example $x = \{w_1, \ldots, w_i, \ldots, w_n\}$ and the adversarial example $x_{adv} = \{w'_1, \ldots, w'_i, \ldots, w'_n\}$. $d(\cdot)$ is usually defined as the p-norm distance: $d(x - x_{adv}) = \|x - x_{adv}\|_p = (\sum_i \|w_i - w'_i\|_p)^{\frac{1}{p}}$.

2.2 Text Adversarial Attacks

Here we provide a brief overview on three popular synonym substitution based adversarial attack methods.

Greedy Search Algorithm (GSA). Kuleshov et al. (2018) propose a greedy search algorithm to substitute words with their synonyms so as to maintain the semantic and syntactic similarity. GSA first constructs a synonym set W_s for an input text $x = \{w_1, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_n\}$:

$$W_s = \{Syn(w_i, \sigma, k) | 1 \leq i \leq n\}.$$

Initially, let $x_{adv} = x$. Then at each stage for $x_{adv} = \{w'_1, \ldots, w'_{i-1}, w'_i, w'_{i+1}, \ldots, w'_n\}$, GSA finds a word $\hat{w}'_i \in W_s$ that satisfies the syntactic constraint and minimizes the confidence value $S_m(\hat{x}, y_{true})$ where $\hat{x} = \{w'_1, \ldots, w'_{i-1}, \hat{w}'_i, w'_{i+1}, \ldots, w'_n\}$, and updates $x_{adv} = \hat{x}$. Such process iterates till x_{adv} becomes an adversarial example or the word substitution rate reaches a threshold.

Genetic Algorithm (GA). Alzantot et al. (2018) propose a population-based algorithm to replace words with their synonyms so as to generate semantically and syntactically similar adversarial examples. There are three operators in GA:

- $Mutate(x)$: Randomly choose a word w_i in text x that has not been updated and substitute w_i with $\hat{w}_i$, one of its synonyms $Syn(w_i, \sigma, k)$ that does not violate the syntax constraint by the "Google one billion words language model"

(Chelba et al., 2013) and minimize $S_m(\hat{x}, y_{true})$ where $\hat{x} = \{w_1, \ldots, w_{i-1}, \hat{w}_i, w_{i+1}, \ldots, w_n\}$ and $S_m(\hat{x}, y_{true}) < S_m(x, y_{true})$;

- $Sample(\mathcal{P})$: Randomly sample a text x_i from the population $\mathcal{P} = \{x_0, \ldots, x_i, \ldots, x_m\}$ with a probability proportional to $1 - S_m(x_i, y_{true})$;

- $Crossover(a, b)$: Construct a new text $c = \{w_1^c, \ldots, w_i^c, \ldots, w_n^c\}$, where w_i^c is randomly chosen from $\{w_i^a, w_i^b\}$ based on the input texts $a = \{w_1^a, \ldots, w_i^a, \ldots, w_n^a\}$ and $b = \{w_1^b, \ldots, w_i^b, \ldots, w_n^b\}$.

For a text x, GA first generates an initial population $\mathcal{P}^0$ of size m:

$$\mathcal{P}^0 = \{Mutate(x), \ldots, Mutate(x)\}.$$

Then at each iteration, GA generates the next generation of population through *crossover* and *mutation* operators:

$$x_{adv}^{i+1} = \arg\min_{\tilde{x} \in \mathcal{P}^i} S_m(\tilde{x}, y_{true}),$$
$$c_k^{i+1} = Crossover(Sample(\mathcal{P}^i), Sample(\mathcal{P}^i)),$$
$$\mathcal{P}^{i+1} = \{x_{adv}^{i+1}, Mutate(c_1^{i+1}), Mutate(c_2^{i+1}),$$
$$\ldots, Mutate(c_{m-1}^{i+1})\}.$$

GA will terminate when it finds an adversarial example or reaches the maximum number of iteration limit.

Probability Weighted Word Saliency (PWWS). Ren et al. (2019) propose a new synonym substitution based attack method called Probability Weighted Word Saliency (PWWS), which considers the word saliency as well as the classification confidence. Specifically, given a text $x = \{w_1, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_n\}$, PWWS first calculates the saliency of each word $S(x, w_i)$:

$$S(x, w_i) = S_m(x, y_{true}) - S_m(\bar{x}_i, y_{true})$$

where $\bar{x}_i = \{w_1, \ldots, w_{i-1}, unk, w_{i+1}, \ldots, w_n\}$ and "*unk*" means the word is removed from the text. Then PWWS calculates the maximum possible change in the classification confidence resulted from substituting word w_i with one of its synonyms:

$$\Delta S_{m_i}^*(x) = \max_{\hat{w}_i \in Syn(w_i, \sigma, k)} [S_m(x, y_{true}) - S_m(\hat{x}_i, y_{true})],$$

where $\hat{x}_i = \{w_1, \ldots, w_{i-1}, \hat{w}_i, w_{i+1}, \ldots, w_n\}$. Then, PWWS sequentially checks the words in descending order of $\phi(S(x, w_i))_i \cdot \Delta S_{m_i}^*(x)$, where

$\phi(z)_i = \frac{e^{z_i}}{\sum_{k=1}^n e^{z_k}}$, and substitutes the current word w_i with its optimal synonym w_i^*:

$$w_i^* = \arg\max_{\hat{w}_i \in Syn(w_i, \sigma, k)} [S_m(x, y_{true}) - S_m(\hat{x}_i, y_{true})].$$

PWWS terminates when it finds an adversarial example x_{adv} or it has replaced all the words in x.

2.3 Text Adversarial Defenses

There are only a few works for text adversarial defenses.

- In the character-level, (Pruthi et al., 2019) propose to place a word recognition model in front of the downstream classifier to defend character-level adversarial attacks by combating adversarial spelling mistakes.

- In the word level, for defenses on synonym substitution based attacks, only Alzantot et al. (2018) and Ren et al. (2019) incorporate the adversarial training strategy proposed in the image domain (Goodfellow et al., 2015) with their text attack methods respectively, and demonstrate that adversarial training can promote the model's robustness. However, there is no defense method specifically designed to defend the synonym substitution based adversarial attacks.

3 The Proposed Text Defense Method

We first introduce our motivation, then present our text defense method, *Synonym Encoding Method* (SEM).

3.1 Motivation

Let $\mathcal{X}$ denote the input space, $V_\epsilon(x)$ denote the ϵ-neighborhood of data point $x \in \mathcal{X}$, where $V_\epsilon(x) = \{x' \in \mathcal{X} | \|x' - x\| < \epsilon\}$. As illustrated in Figure 1 (a), we postulate that the generalization of the model leads to the existence of adversarial examples. More generally, given a data point $x \in \mathcal{X}$, $\exists x' \in V_\epsilon(x), f(x') \neq y_{true}'$ where x' is an adversarial example of x.

Ideally, to defend the adversarial attack, we need to train a classifier f that not only guarantees $f(x) = y_{true}$, but also assures $\forall x' \in V_\epsilon(x), f(x') = y_{true}'$. Thus, the most effective way is to add more labeled data to improve the adversarial robustness (Schmidt et al., 2018). Ideally, as illustrated in Figure 1 (b), if we have infinite labeled data, we can train a model $f : \forall x' \in V_\epsilon(x), f(x') = y_{true}'$ with high probability so that
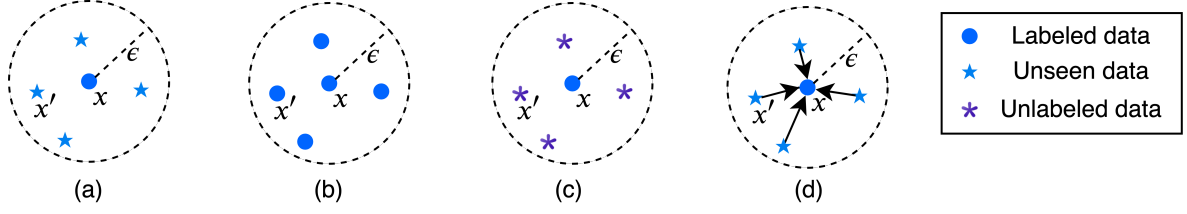
Figure 1: The neighborhood of a data point x in the input space. (a) Traditional training: there exists some data points x' that the model has never seen before and yields wrong classification, in other words, such data point x' is an adversarial example. (b) Adding infinite labeled data: this is an ideal case that the model has seen all the data points to resist adversarial examples. (c) Sharing label: all the neighbors share the same label with x. (d) Mapping neighborhood data points: mapping all neighbors to the center x so as to eliminate adversarial examples.

the model f is robust enough to adversarial examples. Practically, however, labeling data is very expensive and it is impossible to have infinite labeled data.

Because it is impossible to have infinite labeled data to train a robust model, as illustrated in Figure 1 (c), Wong and Kolter (2018) propose to construct a convex outer bound and guarantee that all data points in this bound share the same label. The goal is to train a model $f : \forall x' \in V_\epsilon(x), f(x') = f(x) = y_{true}$. Specifically, they propose a linear-programming (LP) based upper bound on the robust loss by adopting a linear relaxation of the ReLU activation and minimize this upper bound during the training. Then they bound the LP optimal value and calculate the elementwise bounds on the activation functions based on a backward pass through the network. Although their method does not need any extra data, it is hard to scale to realistically-sized networks due to the high calculation complexity.

In this work, as illustrated in Figure 1 (d), we propose a novel way to find a mapping $m : \mathcal{X} \to \mathcal{X}$ where $\forall x' \in V_\epsilon(x), m(x') = x$. In this way, we force the classification to be more smooth and we do not need any extra data to train the model or modify the architecture of the model. All we need to do is to insert the mapping before the input layer and train the model on the original training set. Now the problem turns into how to locate the neighbors of data point x. For image tasks, it is hard to find all images in the neighborhood of x in the input space, and there could be infinite number of neighbors. For NLP tasks, however, utilizing the property that words in sentences are discrete tokens, we can easily find almost all neighbors of an input text. Based on this insight, we propose a new method called *Synonym Encoding* to locate the neighbors of an input text x.

3.2 Synonym Encoding

We assume that the closer the meaning of two sentences is, the closer their distance is in the input space, and we can suppose that the neighbors of x are its synonymous sentences. To find the synonymous sentence, we can substitute words in the sentence with their synonyms. In this way, to construct the mapping m, all we need to do is to cluster the synonyms in the embedding space and allocate a unique token for each cluster. The details of synonym encoding are in Algorithm 1. Note that in our experiment, we implement the synonym encoding on GloVe vectors after counter-fitting (MrkÅąiÄŠ et al., 2016) which injects antonymy and synonymy constraints into vector space representations.

4 Improved Genetic based Text Attack

The current synonym substitution based text adversarial attacks have a constraint that they only substitute words at the same position once (Alzantot et al., 2018; Ren et al., 2019) or replace words with the first k synonyms of the word in the original input x (Kuleshov et al., 2018). This constraint can lead to local minimum for adversarial examples, and it is hard to choose a suitable k as different words may have different number of synonyms.

To address this issue, we propose an Improved Genetic Algorithm (IGA), which allows to substitute words in the same position more than once based on the current text x'. In this way, IGA can traverse all synonyms of a word no matter what value k is. Meanwhile, we can avoid local minimum to some extent as we allow the substitution of the word by the original word in the current position. In order to guarantee that the substituted word is still a synonym of the original word, each word in the same position can be replaced at most λ times and in our experiment, we set $\lambda = 5$.

Algorithm 1 *Synonym Encoding Algorithm*

Input: W: dictionary of words
   n: size of W
   σ: distance for synonyms
   k: number of synonyms for each word
Output: E: encoding result
 1: $E = \{w_1 : \text{None}, \dots, w_n : \text{None}\}$
 2: **for** each word $w_i \in W$ **do**
 3:   **if** $E[w_i] = \text{NONE}$ **then**
 4:     **if** $\exists \hat{w}_i \in Syn(w_i, \sigma, k), E[\hat{w}_i] \neq \text{NONE}$
     then
 5:       $w_i^* \leftarrow$ the closest $\hat{w}_i \in Syn(w_i, \sigma, k)$
 6:       $E[w_i] = E[w_i^*]$
 7:     **else**
 8:       $E[w_i] = w_i$
 9:     **end if**
10:     **for** each word $\hat{w}_i$ in $Syn(w_i, \sigma, k)$ **do**
11:       **if** $E[\hat{w}_i] = \text{NONE}$ **then**
12:         $E[\hat{w}_i] = E[w_i]$
13:       **end if**
14:     **end for**
15:   **end if**
16: **end for**
17: **return** E

Differs to the first genetic based text attack algorithm of Alzantot et al. (2018), we change the structure of the algorithm, including the operators for crossover and mutation. For more details of IGA, see Appendix A.1.

5 Experiments

We evaluate the efficacy of SEM with four attacks, GSA (Kuleshov et al., 2018), GA (Alzantot et al., 2018), PWWS (Ren et al., 2019) and our IGA, on three popular datasets involving three different neural network classification models. The results demonstrate that SEM can significantly improve the robustness of neural networks and IGA can achieve better attack performance as compared with existing attacks.

5.1 Experimental Setup

We first provide an overview of datasets, classification models and baselines used in the experiments.

Datasets. In order to evaluate the efficacy of SEM, we choose three popular datasets: *IMDB*, *AG's News*, and *Yahoo! Answers*. *IMDB* (Potts, 2011) is a large dataset for binary sentiment classification, containing $25,000$ highly polarized movie reviews for training and $25,000$ for testing. *AG's*

News (Zhang et al., 2015) consists news article pertaining four classes: World, Sports, Business and Sci/Tech. Each class contains $30,000$ training examples and $1,900$ testing examples. *Yahoo! Answers* (Zhang et al., 2015) is a topic classification dataset from the "Yahoo! Answers Comprehensive Questions and Answers" version 1.0 dataset with 10 categories, such as Society & Culture, Science & Mathematics, etc. Each class contains 140,000 training samples and 5,000 testing samples.

Models. To better evaluate our method, we adopt several state-of-the-art models for text classification, including Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The embedding dimension for all models are 300 (Mikolov et al., 2013). We replicate the CNN's architecture from (Kim, 2014), which contains three convolutional layers with filter size of 3, 4, and 5 respectively, a max-pooling layer and a fully-connected layer. LSTM consists of three LSTM layers where each layer has 300 LSTM units and a fully-connected layer (Liu et al., 2016). Bi-LSTM contains a bi-directional LSTM layer whose forward and reverse have 300 LSTM units respectively and a fully-connected layer.

Baselines. We take the method of adversarial training (Goodfellow et al., 2015) as our baseline. However, due to the low efficiency of text adversarial attacks, we cannot implement adversarial training as it is in the image domain. In the experiments, we adopt PWWS, which is quicker than GA and IGA, to generate 10% adversarial examples of the training set, and re-train the model incorporating adversarial examples with the training data.

5.2 Evaluation on Defense Methods

To evaluate the efficacy of the SEM method, we randomly sample 200 correctly classified examples on different models from each dataset and use the above attack methods to generate adversarial examples with or without defenses. The more effective the defense method is, the smaller the classification accuracy of the model drops. Table 1 shows the efficacy of various attack and defense methods.

For each network model, we look at each row to find the best defense result under the setting of no attack, or GSA, PWWS, GA, and IGA attacks:

- Under the setting of no attack, adversarial training (AT) could improve the classification accuracy of the models on all datasets, as adversarial training (AT) is also the way to augment

Table 1: The classification accuracy (%) of various models on the datasets, with and without defenses, under adversarial attacks. For each model (Word-CNN, LSTM, or Bi-LSTM), if we look at each row, the highest classification accuracy for various defense methods is highlighted in **bold** to indicate the **best defense efficacy**; if we look at each column, the lowest classification accuracy under various adversarial attacks is highlighted in <u>underline</u> to indicate the best attack efficacy. NT: Normal Training, AT: Adversarial Training.

Dataset	Attack	Word-CNN			LSTM			Bi-LSTM		
		NT	AT	SEM	NT	AT	SEM	NT	AT	SEM
IMDB	No Attack	88.7	**89.1**	86.8	87.3	**89.6**	86.8	88.2	**90.3**	87.6
	GSA	13.3	16.9	**66.4**	8.3	21.1	**72.2**	7.9	20.8	**73.1**
	PWWS	4.4	5.3	**71.1**	2.2	3.6	**77.3**	1.8	3.2	**76.1**
	GA	7.1	10.7	**71.8**	2.6	9.0	**77.0**	1.8	7.2	**71.6**
	IGA	<u>0.9</u>	<u>2.7</u>	**65.1**	<u>0.9</u>	<u>1.8</u>	**71.2**	<u>0.9</u>	<u>2.7</u>	**69.3**
AG's News	No Attack	91.7	**92.2**	88.7	91.8	**92.1**	90.9	91.9	**92.3**	90.1
	GSA	33.0	37.8	**<u>63.9</u>**	45.9	56.7	**83.2**	36.8	42.1	**<u>79.8</u>**
	PWWS	30.7	41.5	**67.6**	50.0	55.7	**85.0**	39.1	44.0	**81.1**
	GA	24.1	40.6	**77.9**	43.8	57.3	**86.4**	37.4	44.4	**82.3**
	IGA	<u>21.5</u>	<u>35.5</u>	**70.3**	<u>40.0</u>	<u>55.3</u>	**81.8**	<u>32.7</u>	<u>41.1</u>	**81.5**
Yahoo! Answers	No Attack	68.4	**69.3**	65.8	71.6	**71.7**	69.0	72.3	**72.8**	70.2
	GSA	19.6	20.8	**<u>49.4</u>**	27.6	30.5	**<u>48.6</u>**	24.6	30.9	**<u>53.4</u>**
	PWWS	10.3	12.5	**52.6**	21.1	22.9	**54.9**	17.3	20.0	**57.2**
	GA	13.7	16.6	**59.2**	15.8	17.9	**66.2**	13.0	16.0	**63.2**
	IGA	<u>8.9</u>	<u>10.0</u>	**51.4**	<u>10.5</u>	<u>15.1</u>	**53.3**	<u>12.4</u>	<u>13.5</u>	**55.7**

the training data. Our defense method SEM reaches an accuracy very close to normal training (NT), which is a common phenomenon in image domain (Zhang and Wang, 2019; Song et al., 2019).

- Under the four attacks, however, the classification accuracy with normal training (NT) and adversarial training (AT) drops significantly. For normal training (NT), the accuracy degrades more than 75%, 42% and 40% on the three datasets respectively. And adversarial training (AT) cannot defend these attacks effectively, especially for PWWS and IGA on *IMDB* and *Yahoo! Answers*, where AT only improves the accuracy a little (smaller than 5%). By contrast, SEM can remarkably improve the robustness of the deep models for all the four attacks.

5.3 Defense for Transferability

In the image domain, the transferability of adversarial attack refers to its ability to decrease the accuracy of models using adversarial examples generated based on other models (Szegedy et al., 2014; Goodfellow et al., 2015). Papernot et al. (2016) find that the adversarial examples in NLP also ex-

hibit a good transferability. Therefore, a good defense method not only could defend the adversarial attacks but also resists the transferability of adversarial examples.

To evaluate the ability of preventing the transferability of adversarial examples, we generate adversarial examples on each model under normal training, and test them on other models with or without defense on *Yahoo! Answers*. The results are shown in Table 2. Almost on all models with adversarial examples generated by other models, SEM could yield the highest classification accuracy.

5.4 Evaluation on IGA vs. GA

For text attacks, we compare the proposed IGA with GA from various aspects, including attack efficacy, word substitution rate, example generation efficiency, transferability and human evaluation.

Attack Efficacy. As shown in Table 1, looking at each column, we see that under normal training (NT) and adversarial training (AT), IGA can always achieve the lowest classification accuracy, which corresponds to the highest attack success rate, on all models and datasets among the four attacks. Under the third column of SEM defense, though

Table 2: The classification accuracy (%) of various models for adversarial examples generated on other models on *Yahoo! Answers* for evaluating the transferability. * indicates that the adversarial examples are generated based on this model.

Attack	Word-CNN			LSTM			Bi-LSTM		
	NT	AT	SEM	NT	AT	SEM	NT	AT	SEM
GSA	19.6*	52.7	**57.5**	52.6	58.4	**61.8**	52.7	57.5	**59.6**
PWWS	10.3*	54.4	**57.2**	46.5	57.7	**61.4**	53.1	**57.9**	56.8
GA	13.7*	49.9	**55.2**	43.0	**66.0**	65.6	56.4	65.5	**67.4**
IGA	8.9*	53.7	**56.5**	46.5	56.6	**58.6**	53.1	56.8	**57.2**
GSA	47.2	52.7	**55.9**	27.6*	60.9	**62.1**	53.8	61.9	**62.8**
PWWS	43.7	54.7	**56.5**	21.1*	59.9	**61.8**	50.6	59.2	**61.4**
GA	41.0	48.5	**54.8**	15.8*	57.4	**57.5**	43.4	58.2	**60.9**
IGA	47.8	53.0	**55.2**	10.5*	53.4	**58.3**	53.5	59.0	**61.1**
GSA	43.7	53.4	**54.3**	52.9	57.7	**58.7**	24.6*	60.4	**61.1**
PWWS	41.7	48.5	**49.7**	41.2	58.0	**60.4**	17.3*	**60.1**	58.6
GA	36.9	45.7	**49.0**	43.0	58.8	**60.7**	13.0*	58.2	**60.0**
IGA	44.8	50.6	**52.3**	45.1	57.4	**58.7**	12.4*	60.1	**61.1**

IGA may not be the best among all attacks, IGA always outperforms GA.

Word Substitution Rate. Besides, as depicted in Table 3, IGA can yield lower word substitution rate than GA on most models. Note that for SEM, GA can yield lower word substitution rate, because GA may not replace the word as most words cannot bring any benefit for the first replacement. This indicates that GA stops at local minimum while IGA continues to substitute words and gains a lower classification accuracy, as demonstrated in Table 1.

Generating Efficiency. Moreover, IGA is four times faster than GA overall because it needs less iterations to generate adversarial examples and does not have the syntax check module. In our experiments, it usually takes 6-10 minutes for IGA to generate an example while GA needs 35-45 minutes on average.

Transferability. As shown in Table 2, the adversarial examples generated by IGA maintain roughly the same transferability as those generated by GA. For instance, if we generate adversarial examples on Word-CNN (column 2, NT), GA can achieve better transferability on LSTM with NT (column 5) while IGA can achieve better transferability on LSTM with AT and SEM (column 6, 7).

Human Evaluation. To further verify that the perturbations in the adversarial examples generated by IGA are hard for humans to perceive, we also perform a human evaluation on *IMDB* with 35 volunteers. We first randomly choose 100 benign examples that can be classified correctly and generate adversarial examples by GA and IGA on the three models so that we have a total of 700 examples. Then we randomly split them into 7 groups where each group contains 100 examples. We ask every five volunteers to classify one group independently. The accuracy of human evaluation on benign examples is 93.7%.

As shown in Table 4, the classification accuracy of human on adversarial examples generated by IGA is slightly higher than those generated by GA, and is slightly closer to the accuracy of human on benign examples, which means the adversarial examples generated by IGA are more realistic to humans. This is counter-intuitive as we do not adopt the syntax check module as GA does, however, as IGA chooses synonyms within a small embedding distance and tries to avoid local minimum to achieve a lower word substitution rate, leading IGA a slightly better human evaluation.

Summary. IGA outperforms GA in all the aspects we compare. IGA achieves the highest attack success rate compared with other synonyms substitution based adversarial attacks and yields lower word substitution rate than GA. Besides, the adversarial examples generated by IGA maintains the same transferability as GA and ours are slightly harder for human to perceive the perturbation. To give an intuitive experience, we list some adver-
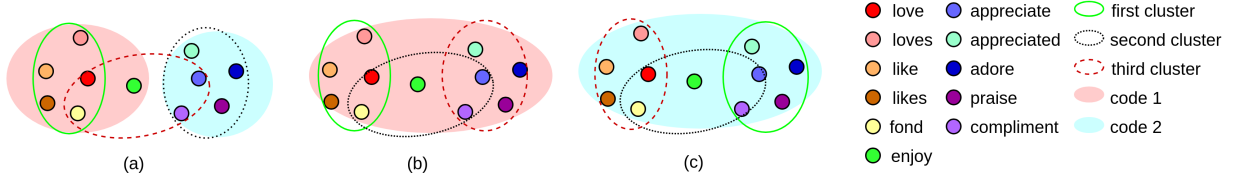
Figure 2: An illustration for different orders to traverse the synonyms at the second line of Algorithm 1, shown in the word embedding space. (a) First traverse words in the left, then words in the right and finally words in the middle. The synonyms are encoded into two different codes (left and right). (b) First traverse words in the left, then words in the middle and finally words in the right. All the synonyms are encoded into a unique code of the left. (c) First traverse words in the right, then words in the middle and finally words in the left. All the synonyms are encoded into a unique code of the right.

Table 3: The word substitution rate (%) for GA and IGA on different models.

Dataset	Attack	Word-CNN			LSTM			Bi-LSTM		
		NT	AT	SEM	NT	AT	SEM	NT	AT	SEM
IMDB	GA	9.3	9.3	**4.3**	10.0	10.6	**4.0**	6.3	8.0	**3.1**
	IGA	**6.4**	**7.8**	6.3	**5.3**	**6.9**	5.7	**5.6**	**6.6**	7.5
AG's News	GA	**10.3**	17.7	6.6	18.3	21.2	9.8	**12.0**	21.4	8.5
	IGA	12.1	**11.4**	**6.3**	**14.4**	**12.0**	**9.6**	13.5	**11.6**	**4.6**
Yahoo! Answers	GA	12.4	9.5	4.7	12.5	15.8	8.1	13.9	15.3	**4.7**
	IGA	**6.4**	**7.1**	**3.7**	**7.9**	**8.2**	**4.5**	**3.6**	**8.2**	5.1

Table 4: Classification accuracy (%) on adversarial examples by human evaluation.

	Word-CNN	LSTM	Bi-LSTM
GA	88.9	88.0	89.0
IGA	**89.3**	**89.0**	**89.3**

sarial examples generated by GA and IGA in Appendix A.3.

5.5 Discussion on the sensitivity of SEM

In this subsection, we discuss some sensitive issues on SEM, including the order to traverse the word and the impact of the hyper-parameters.

As shown in Figure 2, the order to traverse the word at the second line of Algorithm 1 can actually influence the final synonyms encoding code for a word, and it can even lead to different codes for the same synonyms set. However, the aim of synonym encoding is to find an encoder to defend adversarial examples rather than to find an exact and unique code for each synonym set. Note that for a text $x = \{w_1, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_n\}$, if we just replace an arbitrary word w_i with one of its synonyms $\hat{w}_i$ to obtain a new text

$\hat{x} = \{w_1, \ldots, w_{i-1}, \hat{w}_i, w_{i+1}, \ldots, w_n\}$, we usually have $f(x) = f(\hat{x})$. Therefore, different codes for the same synonyms set hardly influence the efficacy of SEM and this randomness might also make it harder to be attacked.

Besides, we further explore how the hyper-parameters ϵ and k in Algorithm 1 influence the efficacy of SEM as shown in Appendix A.2. According to the analysis, it is significant to find suitable ϵ and k to achieve a good trade-off on the accuracy of both benign examples and adversarial examples and in our experiments, we set $\epsilon = 0.5$ and $k = 10$.

6 Conclusion

In this work, we propose the first word-level adversarial defense method called *Synonym Encoding Method* (SEM) for text classification. SEM encodes the synonyms of each word to defend synonym substitution based adversarial attacks, which are currently the best text attack methods. Extensive experiments show that SEM can effectively defend adversarial attacks and degrade the transferability of adversarial examples, at the same time SEM can maintain the classification accuracy on benign data.

In addition, we propose a word-level adversarial attack called Improved Genetic Algorithm (IGA), which achieves higher attack success rate with lower word substitution rate, as compared with the first genetic based attack algorithm proposed in 2018 (Alzantot et al., 2018). At the same time, IGA could maintain the transferability of adversarial examples as GA does.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *Empirical Methods in Natural Language Processing (EMNLP)*.

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

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv Preprint arXiv:1312.3005*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. 2018. Countering adversarial images using input transformations. *International Conference on Learning Representations (ICLR)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition (CVPR)*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Empirical Methods in Natural Language Processing (EMNLP)*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems (NeurIPS)*.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural nanguage classification problems. *OpenReview submission OpenReview:r1QZ3zbAZ*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. *AAAI Conference on Artificial Intelligence (AAAI)*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *International Joint Conference on Artificial Intelligence (IJCAI)*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *International Joint Conference on Artificial Intelligence (IJCAI)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *International Conference on Learning Representations (ICLR)*.

Nikola MrkÅąiÄĞ, Diarmuid ÃŞ SÃľaghdha, Blaise Thomson, Milica GaÅąiÄĞ, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. *IEEE Military Communications Conference (MILCOM)*.

Christopher Potts. 2011. On the negativity of egation. *Proceedings of Semantics and Linguistic Theory (SALT)*.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. *Association for Computational Linguistics (ACL)*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. *Association for Computational Linguistics (ACL)*.

Nestor Rodriguez and Sergio Rojas-Galeano. 2018. Shielding googleâĂŹs language toxicity model against adversarial attacks. *arXiv preprint arXiv:1801.01828*.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander MÄĚdry. 2018. Adversarially robust generalization requires more data. *Neural Information Processing Systems (NeurIPS)*.

Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. 2019. Improving the generalization of adversarial training with domain adaptation. *International Conference on Learning Representations (ICLR)*.

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

Eric Wong and J. Zico Kolter. 2018. Provable defenses via the convex outer adversarial polytope. *International Conference on Machine Learning (ICML)*.

Haichao Zhang and Jianyu Wang. 2019. Defense against adversarial attacks using feature scattering-based adversarial training. *Neural Information Processing Systems (NeurIPS)*.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2019. Generating textual adversarial examples for deep learning models: A survey. *arXiv Preprint arXiv:1901.06796*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Neural Information Processing Systems (NeurIPS)*.

A Appendices

A.1 Details of IGA

Here we introduce our Improved Genetic Algorithm (IGA) in details and show how IGA differs from the first proposed generic attack method, GA (Alzantot et al., 2018). Regard a text as a chromosome, there are two operators in IGA:

- $Crossover(a,b)$: For two texts a and b where $a = \{w_1^a, \ldots, w_{i-1}^a, w_i^a, w_{i+1}^a, \ldots, w_n^a\}$ and $b = \{w_1^b, \ldots, w_{i-1}^b, w_i^b, w_{i+1}^b, \ldots, w_n^b\}$, randomly choose a crossover point i from 1 to n, and generate a new text $c = \{w_1^a, \ldots, w_{i-1}^a, w_i^a, w_{i+1}^b, \ldots, w_n^b\}$.

- $Mutate(x, w_i)$: For a text $x = \{w_1, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_n\}$ and a word w_i, replace w_i with $\hat{w}_i$ where $\hat{w}_i \in Syn(w_i, \sigma, k)$ to generate a new text $\hat{x} = \{w_1, \ldots, w_{i-1}, \hat{w}_i, w_{i+1}, \ldots, w_n\}$ that minimizes $S_m(\hat{x}, y_{true})$.

The details of IGA are described in Algorithm 2.

Algorithm 2 *The Improved Genetic Algorithm*

Input: x: input text, y_{true}: true label for x, M: maximum number of iterations
Output: x_{adv}: adversarial example
1: **for** each word $w_i \in x$ **do**
2:     $\mathcal{P}_i^0 \leftarrow Mutate(x, w_i)$
3: **end for**
4: **for** $g = 1 \rightarrow M$ **do**
5:     $x_{adv} = \arg\min_{x_i \in \mathcal{P}^g} S_m(x_i, y_{true})$
6:     **if** $f(x_{adv}) \neq y_{true}$ **then**
7:        **return** x_{adv}
8:     **end if**
9:     $\mathcal{P}_1^g \leftarrow x_{adv}$
10:    **for** $i = 2 \rightarrow |\mathcal{P}^{g-1}|$ **do**
11:       Randomly sample $parent_1$, $parent_2$ from $\mathcal{P}^{g-1}$
12:       $child = Crossover(parent_1, parent_2)$
13:       Randomly choose a word w in $child$
14:       $\mathcal{P}_i^g \leftarrow Mutate(child, w)$
15:    **end for**
16: **end for**
17: **return** x_{adv}

Compared with GA, IGA has the following differences:

- $Initialization$: GA initializes the first population randomly, while IGA initializes the first population by replacing each word by its optimal synonym, so our population is more diversified.

- $Crossover$: To better simulate the reproduction and biological crossover, we randomly cut the text from two parents and concat two fragments into a new text rather than randomly choose a word of each position from the two parents.

- $Mutation$: Different from GA, IGA allows to replace the word that has been replaced before so that we can avoid local minimum.

The selection of the next generation is similar to GA, which greedily chooses the optimal offspring, and then generates other offsprings by $Mutate(Crossover(\cdot, \cdot))$ on two randomly chosen parents. But as $Mutate$ and $Crossover$ are different, IGA has very different offsprings. Besides, we think that the syntax check module in GA is not necessary and time-consuming because the synonyms can assure the syntax to some extent, and we did not adopt the syntax check module to accelerate the algorithm.

A.2 Hyper-parameters study on SEM

In this subsection, we explore how hyper-parameters ϵ and k of SEM influence the efficacy using three models on *IMDB* with or without adversarial attacks. We try different ϵ ranging from 0 to 1.2 and k ranging from 5 to 15. The results are illustrated in Figure 3 and 4 respectively.

On benign data, as shown in Figure 3(a) and 4(a), the classification accuracy of the models decrease a little when ϵ or k increases. Because a bigger ϵ or k indicates that we need less words to train the model, which could degrade the efficacy of the models. Nevertheless, the classification accuracy does not decrease much as SEM could maintain the semantic invariance of the original text after encoding.

Then we show the defense efficacy of SEM on the three models when changing the value of ϵ, as shown in Figure 3(b)-(d). When $\epsilon = 0$, SEM could not take any impact, we see that the accuracy is the lowest under all attacks. When ϵ increases, SEM starts to defend the attacks, the accuracy increases rapidly and reaches the peak when $\epsilon = 0.5$. Then the accuracy decays slowly if we continue to increase ϵ. Thus, we choose $\epsilon = 0.5$ to have a good trade-off on the accuracy of benign examples and adversarial examples.

Finally, we show the defense efficacy of SEM

on the three models when changing the value of k, as shown in Figure 4(b)-(d). When $k = 5$, some synonyms cannot be encoded into the same code, we see that SEM indeed has some impact when compared with adversarial training. When k increases, more synonyms can be encoded into the same code and SEM can defend the attack effectively, the accuracy increases rapidly and reaches the peak when $k = 10$. Then the accuracy decays slowly and becomes stable if we continue to increase k. Thus, we choose $k = 10$ to be a good trade-off on the accuracy of benign examples and adversarial examples.

In conclusion, small ϵ or k causes that some synonyms cannot be encoded correctly which leads bad defense efficiency, while too large ϵ or k might let SEM encode some words which are not synonyms and influence the efficacy. In our experiment, we choose $\epsilon = 0.5$ and $k = 10$.

A.3 Adversarial Examples Generated by GA and IGA

To show the generated adversarial examples, we randomly pick some benign examples from *IMDB* and generate adversarial examples by GA and IGA respectively on several models. The examples are shown in Table 6 to Table 7. We see that IGA substitutes less words than GA on these models under normal training.
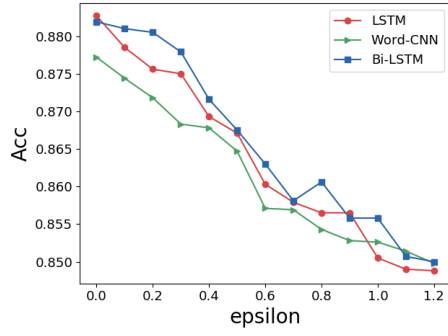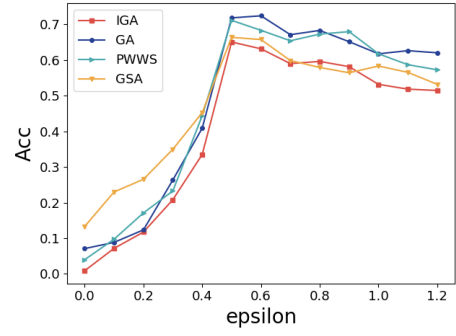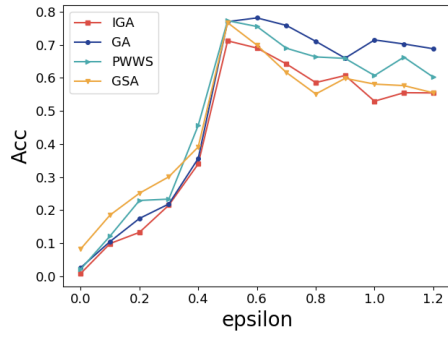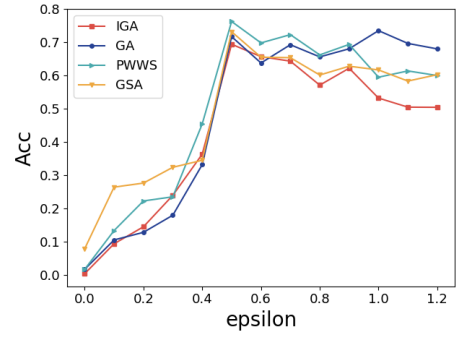
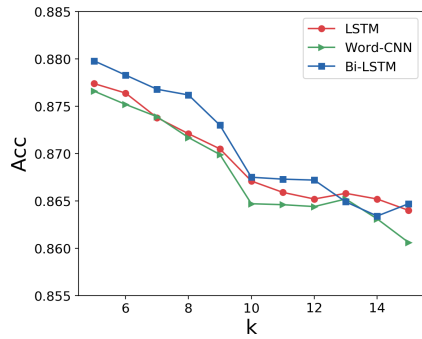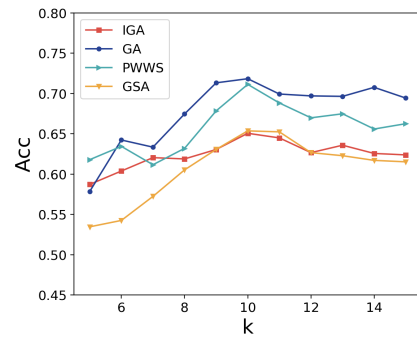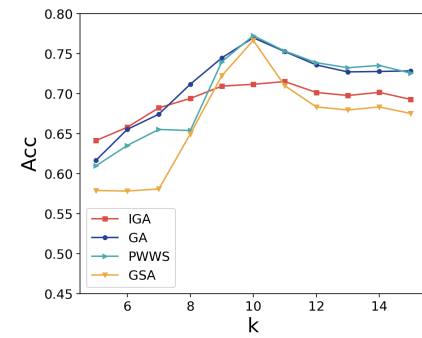| (a) Models under no attack	(b) Word-CNN under attacks
(c) LSTM under attacks	(d) Bi-LSTM under attacks

Figure 3: The classification accuracy for various ϵ ranging from 0 to 1.2 for three models on *IMDB*.

(a) Models under no attack	(b) Word-CNN under attacks
(c) LSTM under attacks	(d) Bi-LSTM under attacks

Figure 4: The classification accuracy for various k ranging from 5 to 15 for three models on *IMDB*.
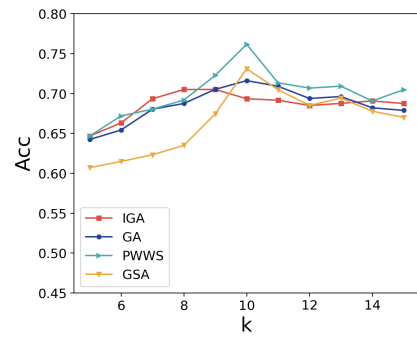
Table 5: The adversarial examples generated by GA and IGA on *IMDB* using Word-CNN model.

	Confidence(%)	Prediction	Text
Original	97.9	1	I enjoyed this film which I thought was well written and acted , there was plenty of humour and a provoking storyline, a warm and enjoyable experience with an emotional ending.
	99.7	0	I am sorry but this is the worst film I have ever seen in my life. I cannot believe that after making the first one in the series, they were able to get a budget to make another. This is the least scary film I have ever watched and laughed all the way through to the end.
	95.8	1	This is a unique masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!
GA	50.6	0	I cared this film which I thought was well written and acted, there was plenty of humour and a igniting storyline, a tepid and enjoyable experience with an emotional ending.
	92.7	1	I am sorry but this is the harshest film I have ever seen in my life. I cannot believe that after making the first one in the series, they were able to get a budget to make another. This is the least scary film I have ever watched and laughed all the way through to the end.
	59.0	0	This is a sole masterpiece made by the nicest director permanently lived in the ussr. He knows the art of film making and can use it much well. If you find this movie, buy or copy it!
IGA	88.3	0	I enjoyed this film which I think was well written and acted, there was plenty of humour and a causing storyline, a lukewarm and enjoyable experience with an emotional ending.
	70.8	1	I am sorry but this is the hardest film I have ever seen in my life. I cannot believe that after making the first one in the series, they were able to get a budget to make another. This is the least scary film I have ever watched and laughed all the way through to the end.
	54.8	0	This is a sole masterpiece made by the best director permanently lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!

Table 6: The adversarial examples generated by GA and IGA on *IMDB* using LSTM model.

	Confidence(%)	Prediction	Text
Original	99.9	1	I enjoyed this film which I thought was well written and acted , there was plenty of humour and a provoking storyline, a warm and enjoyable experience with an emotional ending.
	97.2	0	I am sorry but this is the worst film I have ever seen in my life. I cannot believe that after making the first one in the series, they were able to get a budget to make another. This is the least scary film I have ever watched and laughed all the way through to the end.
	99.7	1	This is a unique masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!
GA	88.2	0	I enjoyed this film which I thought was well written and proceeded, there was plenty of humorous and a igniting storyline, a tepid and enjoyable experience with an emotional terminate.
	99.9	1	I am sorry but this is the hardest film I have ever seen in my life. I cannot believe that after making the first one in the series they were able to get a budget to make another. This is the least terrifying film I have ever watched and laughed all the way through to the end.
	68.9	0	This is a unique masterpiece made by the best superintendent ever lived in the ussr. He knows the art of film making and can use it supremely alright. If you find this movie, buy or copy it!
IGA	72.1	0	I enjoyed this film which I thought was well written and acted, there was plenty of humour and a provoking storyline, a lukewarm and agreeable experience with an emotional ending.
	99.8	1	I am sorry but this is the hardest film I have ever seen in my life. I cannot believe that after making the first one in the series, they were able to get a budget to make another. This is the least scary film I have ever watched and laughed all the way through to the end.
	86.2	0	This is a sole masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!

Table 7: The adversarial examples generated by GA and IGA on *IMDB* using Bi-LSTM model.

	Confidence(%)	Prediction	Text
Original	99.6	1	I enjoyed this film which I thought was well written and acted , there was plenty of humour and a provoking storyline, a warm and enjoyable experience with an emotional ending.
	97.0	0	I am sorry but this is the worst film I have ever seen in my life. I cannot believe that after making the first one in the series, they were able to get a budget to make another. This is the least scary film I have ever watched and laughed all the way through to the end.
	95.3	1	This is a unique masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!
GA	98.2	0	I enjoyed this film which I thought was well written and proceeded, there was plenty of humorous and a igniting storyline, a tepid and enjoyable experiment with an emotional terminate.
	78.8	1	I am sorry but this is the hardest film I have ever seen in my life. I cannot believe that after making the first one in the series, they were able to get a budget to make another. This is the least scary film I have ever watched and laughed all the way through to the end.
	98.8	0	This is a unique masterpiece made by the best superintendent ever lived in the ussr. He knows the art of film making and can use it supremely alright. If you find this movie buy or copy it!
IGA	81.2	0	I enjoyed this film which I thought was alright written and acted, there was plenty of humour and a arousing storyline, a lukewarm and enjoyable experiment with an emotional ending.
	78.8	1	I am sorry but this is the hardest film I have ever seen in my life. I cannot believe that after making the first one in the series they were able to get a budget to make another. This is the least scary film I have ever watched and laughed all the way through to the end.
	98.9	0	This is a sole masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very alright. If you find this movie buy or copy it!