Harnessing Virtual Adversarial Attack for Named Entity Recognition

Anonymous ACL submission

Abstract

Named entity recognition (NER) acts as a fundamental task in natural language processing. However, its robustness is currently barely studied. This paper finds that the conventional text attack for sentence classification can result in label mutation for NER, due to the naturally finer granularity of named entity ground truth. We therefore define a new style of text attack, *virtual attack*. *Virtual* indicates that the attack does not rely on the ground truth but the model prediction. On top of that, we propose a novel fast NER attacker, where we try to insert a "virtual boundary" into the text. It turns out the current strong language models (e.g. RoBERTa, DeBERTa) suffer from a high preference to wrongly recognize those virtual boundaries as entities. Our attack is shown to be effective on both English and Chinese, achieving a 70%-90% attack success rate, and is 50 times faster than the previous methods.

1 Introduction

Named Entity Recognition (NER) aims to find predefined named entities such as locations, persons or organizations in a text. As a fundamental task in natural language processing (NLP), NER plays an important role on various downstream tasks such as text generation (Clark et al., 2018), entity link (Sil and Yates, 2013), machine translation (Babych and Hartley, 2003; Nikoulina et al., 2012), etc. In recent years, NER has received extensive attention and various NER models have achieved impressive performances on benchmarks such as OntoNotes5.0 (Weischedel et al., 2013), WNUT2017 (Derczynski et al., 2017), MSRA (Levow, 2006), etc.

Despite the large number of studies on how to improve the prediction accuracy of NER, existing research on the robustness of current NER models is still lacking. In the text domain, a common practice to evaluate the robustness of an NER model is adversarial attack. However, a majority of the

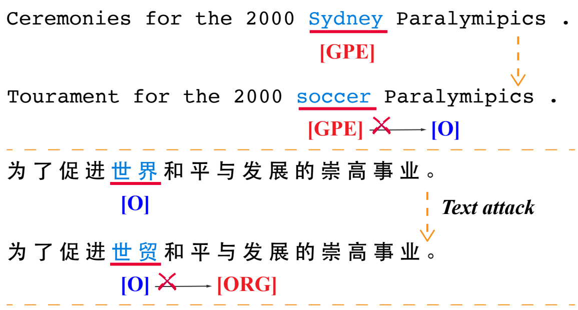

Figure 1: Examples where the conventional attacker results in label mutations. The examples are selected from OntoNotes.

nowadays studies mainly focus on sentence classification (e.g. sentiment analysis, language inference) (Gao et al., 2018; Iyyer et al., 2018; Jin et al., 2020; Garg and Ramakrishnan, 2020; Li et al., 2021) or question answering (Gan and Ng, 2019; Ribeiro et al., 2018; Tan et al., 2020b). More recently, Simoncini and Spanakis first to pay attention to the adversarial attack method for NER and develop a framework called SeqAttack. They define an NER-oriented goal function and adapt the above-mentioned sentence classification and question answering methods from the TextAttack (Morris et al., 2020) framework to NER. Lin et al. subsequently propose RockNER, where they combine entity-level and context-level word substitution to obtain the adversarial examples. However, there are still several key issues that remain to be solved:

• **Label Mutation.** The current attack methods for NER apply word insertion, swapping or substitution to the original example while keeping its ground truth unchanged by restricting the semantic difference. It is reasonable for text classification tasks, since the risk of modifying individual words to reverse the semantic of the entire sentence is low. However, for NER, the ground truth is weakly subject to semantic. Thus, it is more likely to obtain an unreliable adversarial example that do not match its ground truth, which we call *label mutation*. We show an example of label mutation in Figure 1, where a GPE entity *Sydney* (geopolitical) in the original example is replaced by *soccer*, and *world* (世界) is replaced by *WTO* (世贸). However, *soccer* obviously cannot be a GPE and *WTO* is an entity of organization (ORG). As a result of label mutation, we can not obtain a valid example, but a noisy example with unmatched labels.

• **Evaluating NER Attack.** Still in Figure 1, following the traditional criterion, if the model fails to predict *soccer* as GPE or predict *WTO* as a none-entity (O), such an attack will be deemed successful (i.e. the model is not robust against such an example). Due to the potential label mutation problem, it is hard for the current attack methods to justify the obtained adversarial examples since one by no means label them manually. Therefore, a more efficient method for evaluating the robustness of an NER model is urgently needed.

• **High Attacking Expense.** Existing attack methods usually require a large number of loops to search for the adversarial examples. For example, for substitution-based methods, they first need to generate a candidate vocabulary according to some pre-defined rules, and then try to replace the word in each position of the original sentence with every word in the candidate vocabulary. Such a manner leads to a huge computation cost.

To overcome the above issues, in this work, we propose a novel effective virtual attack called *ViBA: Virtual Boundary Attack*. (1) We first propose a new style of attack named Virtual Adversarial Attack which is agnostic to the ground truth and evaluate the robustness of an NER model by comparing the two model predictions before and after being attacked, thus free from label mutation. (2) Based on the idea of Virtual Adversarial Attack, our ViBA generates high-quality adversarial examples by inserting the "virtual boundary" into the text and the NER model will be fooled due to the co-occurrence of boundaries and entities. (3) Our ViBA has a very low search complexity and is 50 times faster than previous methods, while achieving an 80% attack success rate on the widely-used benchmarks. We also conduct empirical experiments to interpret the effectiveness of ViBA and verify the rationality of the motivation to insert boundary. Moreover, we propose two defense strategies to help the NER model defend against ViBA.

Figure 2: An example of virtual boundary attack (text in (a): *Israel will host the prime ministerial election in two weeks.*). The attacker tries to fool the model, leading to the paradox as depicted in (b) and (c), where the model mistakenly recognizes the boundary as an entity due to the co-occurrence.

An example of ViBA is shown in Figure 2. There are two unrobust phenomena: 1. For (a) and (b), when inserting boundary to generate an adversarial example, the model will recognize the boundary as an entity due to the co-occurrence of entity and boundary. 2. For (b) and (c), if the original entity is masked out, the model will not consider this boundary to be an entity. We regard an attack as successful if the adversarial example can cause one of these two paradoxes.

2 Method

This section lays out the background of the traditional adversarial attack. On top of that, we introduce virtual adversarial attack and then propose virtual boundary attack for NER.

2.1 Adversarial Attack

Generally, adversarial attack seeks to find out the worst-case modification on the original example which fools the model prediction. Specifically, let x and y be the input text as well as its ground truth, and $\mathcal{F}$ be the victim model, then the adversarial attack aims to find a specific neighbor of x that satisfies:

$$\mathcal{F}(\mathrm{x}+\delta) \neq y \tag{1}$$

where $\mathrm{x}+\delta$ refers to the adversarial example and δ is to a slight modification. Significantly, Eq.(1) is grounded on the label invariance (i.e. y) before and after the attack. In sentence classification (e.g. sentiment analysis, language inference), for example, δ is always bounded by semantic in the hope that the attack will not change the sentence label.

2.2 Virtual Adversarial Attack

Despite sentence classification, for NER, the semantic bound can no longer keep the invariance of y, since the named entities are largely pre-defined by human. As a result, imposing δ to x is more likely to cause label mutation (e.g. Table 1), where the adversarial example $\mathrm{x}+\delta$ does not meet the satisfaction of Eq.(1). Inspired by virtual adversarial training (Miyato et al., 2018), we propose virtual adversarial attack (*Vttack*) where *virtual* means the attack is agnostic to the ground truth.

Given x and a victim model $\mathcal{F}$, Vttack aims to find a neighbor of x that satisfies:

$$\mathcal{F}(\mathrm{x}+\delta) \neq \mathcal{F}(\mathrm{x}) \tag{2}$$

where $\mathcal{F}(\mathrm{x})$ refers to the original model prediction. Eq.(2) indicates that the attack seeks to find out the worst-case that flips the current model prediction. Such a process is independent of y.

The traditional attack attempts to find out the input point that pushes the model prediction away from the ground truth. However, Vttack attempts to find out the local unsmoothness of two model predictions. Thus, we can define a generalized criterion of Vttack:

$$\mathcal{F}(\mathrm{x}+\delta_1) \neq \mathcal{F}(\mathrm{x}+\delta_2) \tag{3}$$

where $\mathrm{x}+\delta_1$ and $\mathrm{x}+\delta_2$ are both neighbors of x.

Though independent of the ground truth, both Eq.(2) and Eq.(3) should be grounded on the label invariance of two input points (i.e. x and $\mathrm{x}+\delta$ or $\mathrm{x}+\delta_1$ and $\mathrm{x}+\delta_2$). Fortunately, our practice showcases that it can be satisfied more easily.

2.3 Virtual Boundary Attack

We now present *Virtual Boundary Attack (ViBA)*. ViBA is a specific NER attack algorithm that belongs to Vttack, which inserts a specific boundary into the text and seeks to let the model mistakenly recognize it as an entity. The backbone is that the current NER model is highly sensitive to the left and right boundaries of each entity on which it relies for recognition. We thus exploit this property to fool the model.

We also call the inserted boundary "virtual boundary", which has the following two implications. (1) The inserted boundary may not be a real entity. Actually, it is hard to know. (2) The second is closely related to the definition of Vttack. ViBA does not need to care about whether it is a real entity. What it cares about is whether the model prediction of that boundary will be affected by another entity that contains the boundary. As shown in Figure 2 (b) and (c), the model recognizes *Is* (the prefix of *Israel*) as an GPE. Paradoxically, it is no more after *Israel* is masked. It indicates that the model pathologically assumes the co-occurring boundaries are relevant, which is not the way human does. This is exactly what happens in Eq.(3). Algorithm 1 summarizes the ViBA algorithm.

Algorithm 1 Virtual Boundary Attack

Input: Victim model $\mathcal{F}$, input example $\mathcal{X}$, safety distance w.
Output: Adversarial example $\mathfrak{X}$.

1: $\mathcal{Y} \leftarrow \mathcal{F}(\mathcal{X})$
2: $\mathcal{E} \leftarrow$ Extract each entity in $\mathcal{X}$ following $\mathcal{Y}$
3: $\mathcal{L} \leftarrow$ Locate each entity in $\mathcal{X}$ following $\mathcal{Y}$
4: $\mathcal{S} \leftarrow$ Decide safety area following $\mathcal{L}$ and w
5: **for** e in $\mathcal{E}$ **do**
6:     **for** j in $\{1 \sim n\} \setminus \mathcal{S}$ **do**
7:         **for** b in $\{e^{left}, e^{right}\}$ **do**
8:             $\mathcal{X}' \leftarrow$ Insert b before $\mathcal{X}_{[j]}$ in $\mathcal{X}$
9:             $\mathcal{X}'_m \leftarrow$ Mask e in $\mathcal{X}'$
10:             $\mathcal{Y}' \leftarrow \mathcal{F}(\mathcal{X}')$
11:             $\mathcal{Y}'_m \leftarrow \mathcal{F}(\mathcal{X}'_m)$
12:             **if** $\mathcal{Y}' \setminus \mathcal{Y}'_{[j]} \neq \mathcal{Y}$ **then**
13:                 **return** $\mathcal{X}'$
14:             **end if**
15:             **if** $\mathcal{Y}'_{[j]} \neq \mathcal{Y}'_{m[j]}$ **then**
16:                 **return** $\mathcal{X}'$
17:             **end if**
18:         **end for**
19:     **end for**
20: **end for**
21: **return** None

(1) Generate Original Prediction (line 1-3).

Given an input sentence $\mathcal{X} = \mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_n$, we first feed it into the victim model to obtain the original prediction $\mathcal{Y}$. which is a list of predicted named entity tags and has the same length with $\mathcal{X}$. Each tag in $\mathcal{Y}$ is a pre-defined abbreviated label such as "PER" for "Person", "LOC" for "Location", etc. Following the common usage of NER, we also use "O" to denote that a token is not a named entity. Then we extract the named entities

Test set	WNUT	OntoNotes
Examples	686 / 1287	4561 / 9479
Entities per ex.	1.57	2.45
Tokens per ex.	19.67	24.08
Test set	**MSRA**	**OntoNotes**
Examples	2344 / 4365	2392 / 4472
Entities per ex.	2.61	3.13
Tokens per ex.	47.34	45.06

Table 1: Statistics for each used test set. The situation for the training set is similar.

$\mathcal{E}$ as well as their corresponding locations $\mathcal{L}$.

(2) Decide Safety Areas (line 4).

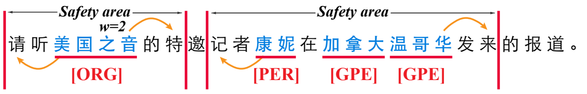

Figure 3: An example of safety areas.

To prevent the inserted boundary from destroying the original entities and their context, we set safety areas for the entities based on safe distance w. Any boundary can not be inserted in a safety area. That is, it is not allowed to insert a boundary inside a named entity and the distance between the inserted boundary and any named entity cannot be less than w. An example is shown as Figure 3.

(3) Generate Candidate Adversarial Example, Masked Example and their Predictions (line 5-11).

Next, we try to generate adversarial examples based on each named entity in the original prediction. For each named entity e in $\mathcal{E}$, its left and right boundaries are extracted first. Then, we go through every position outside the safety areas and insert the boundary to generate a candidate adversarial example $\mathcal{X}'$. To verify that it is indeed the co-occurrence of the inserted boundary and that the named entity causes a change to model prediction, we replace the named entity in the adversarial example with `[MASK]` token and get $\mathcal{X}'_m$. Subsequently, $\mathcal{X}'$ and $\mathcal{X}'_m$ are fed into the victim model to obtain two predictions.

(4) Check Success (line 12-17).

According to the definition of virtual adversarial attack, we use the following two criteria to judge whether an attack is successful:

Criterion 1 (line 12-14). This criterion corresponds to the Eq.(2) and we need to check the consistency of $\mathcal{Y}$ and $\mathcal{Y}'$. Since the boundary inserted at the current position j does not exist in the original sample, this position is ignored in $\mathcal{Y}'$ during comparison.

Criterion 2 (line 15-17). This criterion corresponds to the Eq.(3). We regard $\mathcal{X}'$, $\mathcal{X}'_m$ as $\mathcal{X}$ with two different perturbations. And then compare whether the model's predictions for the currently inserted boundary have changed. Meanwhile, this scenario is also in line with human intuition, that is, only the co-occurrence of the inserted boundary and the original entity will cause the model to be unrobust in the judgment of the insertion position.

3 Experiments

3.1 Datasets

We explore the effectiveness of our ViBA on three widely used public benchmarks of Chinese and English:

- **OntoNotes5.0** (Weischedel et al., 2013) is a multilingual NER dataset which contains three languages: Chinese, English and Arabic. There are eighteen types of named entities in this dataset, eleven of which are types like Person, Organization, etc and seven are values such as Date, Percent, etc. In this paper, we select the popular Chinese and English versions for our experiments.
- **MSRA** (Levow, 2006) is one of the most used Chinese NER datasets which accommodates three named entity types. The data in MSRA is collected from the news domain and is used as a shared task on SIGNAN backoff 2006.
- **WNUT2017** (Derczynski et al., 2017) is an English NER dataset which has six named entity types. This dataset focuses on identifying unusual, previously-unseen entities in the context of emerging discussions and it is more difficult to identify the entities in this dataset.

We present some statistical data of the above benchmarks, as shown in Table 1. The total number of the sentences containing at least one entity and the total number of the sentences in the dataset are shown in the Examples row. It is worth noting that all results in this paper are evaluated on the samples containing at least one entity. In addition, we also count the average number of entities contained in each sample and the average length of each sample. The split of training, test and development sets for the above three datasets is consistent with previous NER works.

	English				*Chinese*			
	WNUT		OntoNotes		MSRA		OntoNotes	
	ASR	SS	ASR	SS	ASR	SS	ASR	SS
BERT$_{\text{base}}$	57.1	98.0	**73.2**	98.1	91.2	98.8	85.5	98.7
RoBERTa$_{\text{large}}$	**65.6**	97.9	70.0	98.1	91.7	98.8	86.9	98.1
DeBERTa$_{\text{large}}$	56.1	98.0	70.7	98.1	-	-	-	-
MacBERT$_{\text{large}}$	-	-	-	-	**93.2**	98.8	**89.4**	98.6

Table 2: The attack success rate (ASR) and semantic similarity (SS) across different models on both English and Chinese NER datasets. A higher ASR suggests that the attacker is more effective in fooling the model.

3.2 Metric

• **Attack Success Rate (ASR)** is the main measurement of the attacker's effectiveness towards the victim model (i.e. the ratio of the achieved eligible adversarial examples over all examples).

• **Semantic Similarity (SS)** serves as a measurement of the similarity between two examples (i.e. cosine similarity). We usually expect the adversarial example to fool the model while maintaining a high similarity to the original one. In this paper, we leverage *text2vec* for both English and Chinese (Xu, 2022).

3.3 Settings

We evaluate our ViBA on the BERT-base (Devlin et al., 2019), RoBERTa-large (Liu et al., 2019b) models of Chinese and English versions. In addition, DeBERTa-large (He et al., 2020) is leveraged for evaluation on the English datasets. MacBERT-large (Cui et al., 2020) is used for evaluation Chinese datasets.

Specifically, we first fine-tune the models on the training set and then use ViBA to generate adversarial examples on the test set. We set the hyper-parameter safety distance $w = 2$ for all the experiments. All experiments are conducted on a single NVIDIA RTX 3090 GPU.

3.4 Main Results

We evaluate our ViBA method for multiple models on different Chinese and English datasets, and the results are shown in Table 2. Among them, we evaluate the Chinese and English versions of BERT-base and RoBERTa-large on the Chinese and English datasets, respectively. MacBERT-large is only valid for Chinese, while DeBERTa-large has an only English version. Overall, as can be seen from our results, ViBA achieves a high success rate when attacking both Chinese and English datasets. The ASR on the Chinese datasets is as high as 85% - 93%. Although relatively lower on the English dataset, the ASR is ranging from 55% to 73% which is still an ideal performance. It is noteworthy that the English datasets generally have shorter sentences whose safe area will be smaller as we defined. So the smaller search space for ViBA will lead to a poor ASR on the English datasets. Overall, ViBA is a great attacker on the above benchmarks.

Table 2 also lists the average SS between the adversarial and original examples. It can be seen that the ASR of all datasets exceeds 98, which guarantees that (1) the semantics of the adversarial examples are nearly the same as the original sentences and (2) the adversarial examples are natural and look close to the original samples.

3.5 Time Analysis

The time complexity of ViBA to attack a sentence is about $O(m \times n)$, where m is the number of named entities in this sentence. Usually, m is much smaller than the sentence length n. Therefore, the time complexity is almost linear with the length of the sentence, which makes the attack speed very fast. To verify it, we reproduce the BAE (Garg and Ramakrishnan, 2020) adapted for NER in Seqattack (Simoncini and Spanakis, 2021) and compare it with our ViBA on the MSRA dataset. The results are shown in the Table 3.

	ASR	Speed (sec per ex.)
BAE	87.2	7.32
Ours	91.2	0.13 ($\times$56)

Table 3: Comparison of time cost.

Compared to the TEXTFOOLER (Jin et al., 2020), CLARE (Li et al., 2021), etc., BAE is already a fast attack algorithm. However, in addition to the obvious advantages of our ViBA over BAE in ASR, our ViBA is 56 times faster than

	ASR
Original	95.8
Mask Boundary	69.6
Mask Inner	86.4

Table 4: Compare the effects of mask boundary/inner words on model recognition performance.

	OntoNotes-en	**OntoNotes-ch**
Boundary Tokens	0.95	0.93
Other Tokens	0.96	0.95

Table 5: The cosine similarity of the hidden-states.

BAE which demonstrates its speed superiority.

4 Discussion

4.1 Interpretation

This section will interpret the effectiveness of our ViBA based on empirical experiments.

4.1.1 Boundary as Trigger

As mentioned in (Lin et al., 2021), the NER models tend to memorize the entity patterns instead of recognizing the entities by context-based reasoning. Following this view, we also imagine that the NER models may memorize some patterns of original named entities and cause ViBA to be effective. Some previous works (Peng and Dredze, 2016; Liu et al., 2019a; Tan et al., 2020a) have proven that integrating the boundary information into the NER models will enhance the ability of the models, which makes us suspicious of the boundary words. Thus we separate the boundary and inner words of the entities to probe which part may be the pattern memorized by the models.

Specifically, we first fine-tune the BERT-base model on the training set of the MSRA and evaluate its recognition performance of named entities on the test set. Then we mask out the boundary words and inner words of the entities in the test set respectively, and then evaluate the recognition performance of the model. The results are shown in Table 4, where all the results are F_1. When calculating F_1, we regard a named entity as correctly recognized only if its boundary and type are both recognized accurately.

As we can see from the results that BERT-base achieves 95.8 F_1 on the original MSRA test set, which is an excellent performance. However, after masking the boundary words of all the named entities, the F_1 of the model on the test set drops sharply by 26.2, compared with the 9.4 F_1 drop of the inner words. Such a phenomenon indeed verifies that the NER model is more sensitive to the boundary words than the inner words, and it tends to recognize the named entities relying on the boundary words. This is also the reason why our ViBA chooses to insert the boundary of the entity into the sentence. The above analyses justify the motivation of our ViBA to insert sentences with boundaries.

4.1.2 Robustness of Encoder and Decoder

The structure of the BERT-style NER models can be summarized as the encoder-decoder structure. The encoder usually leverages a strong pre-trained language model, and the decoder is usually served by the models such as MLP classifier, conditional random field (CRF), etc. The encoder encodes the input sentence into contextual hidden-states. The subsequent decoder performs token-level classification and classifies each word into a pre-defined NER label according to the hidden-state of each word. In this section, we want to figure out why our ViBA can attack successfully.

Our most concerned key question is why the phenomenon in Figure 2 occurs for a successful adversarial example. That is, the adversarial example can make the victim model recognize the inserted boundary as a named entity, but if the original entity is masked and does not co-occur with the inserted boundary, then the model will not predict the inserted boundary as an entity.

Since hidden-states are the only medium between them, we analyze the robustness of the encoder and decoder from the stability of the hidden-states. Specifically, first we generate successful adversarial examples. For each adversarial example $\mathfrak{X}$, it is fed into the NER model to obtain its hidden-states $\mathcal{H}$. Then we mask out the original entity in this adversarial example to get the $\mathfrak{X}_m$ and also input it into the NER model to obtain hidden-states $\mathcal{H}_m$. Then we select the representations of the inserted boundary from the $\mathcal{H}, \mathcal{H}_m$ and calculate the cosine similarity between them. Similar to this dosage, we also calculate the cosine similarity for other tokens. We conduct experiments with BERT-base on OntoNotes-en and OntoNotes-ch datasets. The average values of the cosine similarities are shown in Table 5 .

From the results, we figure out that for the inserted boundary tokens, the cosine similarity of the hidden-states between the $\mathcal{H}$ and $\mathcal{H}_m$ reaches

	OntoNotes-en		OntoNotes-ch	
	ASR	**F_1**	**ASR**	**F_1**
FreeLB	70.5	**89.5**	86.0	85.2
ASA	72.2	89.3	86.8	**85.3**
p	**ASR**	**F_1**	**ASR**	**F_1**
0	72.9	89.2	85.5	85.0
0.3	63.7	88.8	87.1	84.7
0.5	67.7	88.3	85.4	83.6
0.8	69.8	83.1	71.5	63.0

Table 6: The results of masking out the boundary tokens for the encoder.

0.93 in two datasets. It is worth noting that the hidden-states of BERT-base are as high as 768 dimensions, and the cosine similarity so close to 1 shows that the inserted boundary does not result in a significant deviation of the encoder. Similar to this phenomenon, other tokens also obtain an average similarity of 0.95 in two datasets, which further verifies that the encoder is relatively stable to the two sentences $\mathfrak{X}$ and $\mathfrak{X}_m$. According to the above analysis, it can be concluded that when the representation output by the encoder changed slightly in the position of the inserted boundary, the prediction of this boundary by the decoder will be confused. We summarize that for such an encoder-decoder NER model, our ViBA mainly attacks the unrobustness of the decoder.

4.2 Defense Strategy: Boundary Cut

As concluded in Section 4.1, there are two main reasons why our ViBA is effective (1) The NER model is very sensitive to the boundary words of the named entities that tends to recognize the entities depending on the boundary words, and it perhaps also memorizes some boundary patterns. (2) For the NER model of the encoder-decoder structure, its decoder is not robust and even if the hidden-states input to it change slightly, the prediction will be converted.

In this section, we propose a Boundary Cut strategy that can enhance the model's resistance to ViBA from two aspects: (1) Decouple the information of boundary and inner words on the encoder side, thus reducing the model's sensitivity to entity boundary tokens. (2) Dropout the hidden-states to improve the robustness of the decoder.

4.2.1 Mask Out the Boundary for Encoder

Since the NER model is sensitive to boundary tokens, a very straightforward idea is to decouple boundary words and inner words. We achieve this goal with the simplest way of masking out the boundary words at the input of the encoder. In detail, we randomly mask out the left and right boundary tokens of an entity with a probability p during the fine-tuning phase. Then we evaluate the attack effect of the model on the test set. In addition, to explore whether masking out the boundary words during training has an impact on the model's ability to recognize the named entities, we also evaluate it on the test set. We apply BERT-base to conduct experiments on the OntoNotes5.0-en and OntoNotes5.0-ch datasets. The results are shown in Table 6.

It can be seen from the results that compared with the case without masking ($p = 0$), after masking out the boundary words, almost all ASR has a significant decrease, which shows that the dosage of masking out boundary words is useful for decoupling the boundary information and inner words information and can indeed help the NER model to resist ViBA. An exception happens when $p = 0.3$ which makes the model more fragile. Our explanation for this anomaly is that masking out the boundary words will cause a trade-off. On the one hand, it can reduce the model's sensitivity to the boundary by decoupling information of the boundary and the inner words, thus to decrease the ASR. On the other hand, it will also bring noise, which may lead to insufficient training and makes the model fragile. In this case, it may be that the former outweighs the latter. When observing the recognition effect on NER, the F_1 of all experiments just slightly decreases as $p = 0.3, 0.5$ which indicates that the noise introduced by masking out boundary does not cause much loss of recognition performance. And when $p = 0.8$, it is not so surprising that there is a large drop in the recognition performance with such big noise. Overall, when the probability is within a reasonable range, the practice of masking out boundary can effectively help the NER model to resist ViBA without significantly reducing the performance of recognition. Based on our experiments, $p = 0.5$ works best.

We select two adversarial training (AT) methods that are FreeLB (Zhu et al., 2020) and ASA (Wu and Zhao, 2022) as our baselines. Compared with them, although our F_1 is relatively lower, we have

	OntoNotes-en		OntoNotes-ch	
	ASR	$\mathbf{F_1}$	**ASR**	$\mathbf{F_1}$
WP	70.4	88.4	88.4	84.7
p	**ASR**	$\mathbf{F_1}$	**ASR**	$\mathbf{F_1}$
0	72.9	89.2	85.5	85.0
0.3	70.2	88.8	85.7	85.1
0.5	70.8	88.7	84.7	85.0
0.8	75.1	87.6	80.4	84.3

Table 7: The results of applying the dropout to the hidden-states for the decoder and the weight perturbation baseline.

a significantly more advantageous ASR.

4.2.2 Dropout the Hidden-States for Decoder

Since the decoder is relatively unrobust to the hidden-states output by the encoder and ViBA mainly fools the decoder, improving the robustness of the decoder is also a direct idea. Therefore, we propose to apply dropout (Hinton et al., 2012) to the hidden-states in order to alleviate this problem. Specifically, while also considering that the NER model is sensitive to boundary words, we randomly dropout the left and right boundaries of an entity on top of the output hidden-states with a probability p. Following Section 4.2.1, we also conduct experiments on the OntoNotes5.0-en and OntoNotes5.0-ch datasets. The victim model is BERT-base with a vanilla MLP decoder. We take a classic weight perturbation (WP) method (Wen et al., 2018) which can improve model robustness as the baseline.

As shown in Table 7, ASR drops significantly on both OntoNotes-en and OntoNotes-ch when $p = 0.5, 0.3$, meanwhile the F_1 on the test set is almost unaffected. Compared with weight perturbation, we also outperform it with a lower ASR and higher F_1. We can conclude that such a concise dropout method can help the victim model resist ViBA without affecting its recognition accuracy. Also, the model is fragile due to the undertraining problem, and it is understandable to have poor ASR and F_1 when $p = 0.8$.

5 Related Work

Current works on adversarial attack concentrate on text classification, question answering (QA), machine translation, machine reading comprehension, etc. For examples, Gao et al. propose a DeepWordBug algorithm which can effectively fool the deep-learning classifier by small perturbations in a black-box scenario. Iyyer et al. propose a SCPNs network which generates adversarial examples based on syntactic information for text classification task. Jin et al. present a famous TEXTFOOLER baseline which attacks the BERT-style models with excellent effectiveness, utility-preserving ability and efficiency. BAE is proposed by Garg and Ramakrishnan, which is a black box attack aiming at text classification and generates adversarial examples by contextual perturbations. CLARE (Li et al., 2021) produces fluent and grammatical outputs through a mask-then-infill procedure. (Gan and Ng, 2019) attacks the question paraphrasing in the question answering dataset. Tan et al. perturb the inflectional morphology of words to generate plausible and semantically similar adversarial examples. However, none of them aim at the NER task.

Recently, many researchers begin to focus on the robustness of NER models. For example, Mayhew et al. study the impact of capitalization in NER on the model. Das and Paik explore the influence of the surrounding context perturbation on the entity. But none of them propose an algorithm to efficiently generate NER adversarial examples.

Nowadays, there are only a few studies that propose adversarial examples generation methods for NER which are still very lacking. Although Seqattack (Simoncini and Spanakis, 2021) adapts some above-mentioned attack methods for text classification text to NER, it does not propose a new approach. RockNer (Lin et al., 2021) and Breaking BERT (Dirkson et al., 2021) are rare works of adversarial example generation for NER. But essentially, they will bring up the three problems as we mentioned in the introduction.

6 Conclusion

This paper targets to study the robustness of current dominant NER models. Due to label mutation, existing evaluation methods for NER robustness are unreliable. Therefore, we propose Virtual Adversarial Attack which bypasses the problem of label mutation. On top of that, we present Virtual Boundary Attack (ViBA) for NER by inserting a specific boundary into the text, which is able to generate high-quality adversarial examples efficiently. Moreover, we interpret the effectiveness of ViBA and further propose a boundary cut strategy that can help the model defend against ViBA.

References

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.

Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2250–2260, New Orleans, Louisiana. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Sudeshna Das and Jiaul Paik. 2022. Resilience of named entity recognition models under adversarial attack. In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 1–6.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

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

Anne Dirkson, Suzan Verberne, and Wessel Kraaij. 2021. Breaking bert: Understanding its vulnerabilities for biomedical named entity recognition through adversarial attack. *ArXiv preprint*, abs/2109.11308.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

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

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv preprint*, abs/2006.03654.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv preprint*, abs/1207.0580.

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

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Wenyang Gao, Jun Yan, Ryan Moreno, and Xiang Ren. 2021. RockNER: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3728–3737, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wei Liu, Tongge Xu, Qinghua Xu, Jiayu Song, and Yueran Zu. 2019a. An encoding strategy based word-character LSTM for Chinese NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2379–2389, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Stephen Mayhew, Nitish Gupta, and Dan Roth. 2020. Robust named entity recognition with truecasing pretraining. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8480–8487. AAAI Press.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

John 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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Vassilina Nikoulina, Agnes Sandor, and Marc Dymetman. 2012. Hybrid adaptation of named entity recognition for statistical machine translation. In *Proceedings of the Second Workshop on Applying Machine Learning Techniques to Optimise the Division of Labour in Hybrid MT*, pages 1–16, Mumbai, India. The COLING 2012 Organizing Committee.

Nanyun Peng and Mark Dredze. 2016. Improving named entity recognition for Chinese social media with word segmentation representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 149–155, Berlin, Germany. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2369–2374. ACM.

Walter Simoncini and Gerasimos Spanakis. 2021. Seqattack: On adversarial attacks for named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 308–318.

Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang, and Fei Huang. 2020a. Boundary enhanced neural span classification for nested named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9016–9023.

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

R Weischedel, M Palmer, M Marcus, E Hovy, S Pradhan, L Ramshaw, N Xue, A Taylor, J Kaufman, M Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. linguistic data consortium, philadelphia, pa (2013).

Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger B. Grosse. 2018. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Hongqiu Wu and Hai Zhao. 2022. Adversarial self-attention for language understanding. *ArXiv preprint*, abs/2206.12608.

Ming Xu. 2022. Text2vec: Text to vector toolkit. `https://github.com/shibing624/text2vec`.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.