Character-level White-Box Adversarial Attacks against Transformers via Attachable Subwords Substitution

Aiwei Liu, Honghai Yu, Xuming Hu, Shu'ang Li, Li Lin, Fukun Ma,
Yawen Yang, Lijie Wen[†]
Tsinghua University
{liuaw20, yhh21, hxm19, lisa18, lin-l16, mafk19, yyw19}@mails.tsinghua.edu.cn
wenlj@tsinghua.edu.cn[*]

Abstract

We propose the first character-level white-box adversarial attack method against transformer models. The intuition of our method comes from the observation that words are split into subtokens before being fed into the transformer models and the substitution between two close subtokens has a similar effect to the character modification. Our method mainly contains three steps. First, a gradient-based method is adopted to find the most vulnerable words in the sentence. Then we split the selected words into subtokens to replace the origin tokenization result from the transformer tokenizer. Finally, we utilize an adversarial loss to guide the substitution of attachable subtokens in which the Gumbel-softmax trick is introduced to ensure gradient propagation. Meanwhile, we introduce the visual and length constraint in the optimization process to achieve minimum character modifications. Extensive experiments on both sentence-level and token-level tasks demonstrate that our method could outperform the previous attack methods in terms of success rate and edit distance. Furthermore, human evaluation verifies our adversarial examples could preserve their origin labels.

1 Introduction

Adversarial examples are modified input data that could fool the machine learning models but not humans. Recently, Transformer (Vaswani et al., 2017) based model such as BERT (Devlin et al., 2019) has achieved dominant performance on a wide range of natural language process (NLP) tasks. Unfortunately, many works have shown that transformer-based models are vulnerable to adversarial attacks (Guo et al., 2021; Garg and Ramakrishnan, 2020). On the other hand, the adversarial attack could help improve the robustness of models through adversarial training, which emphasizes the importance of finding high-quality adversarial examples.

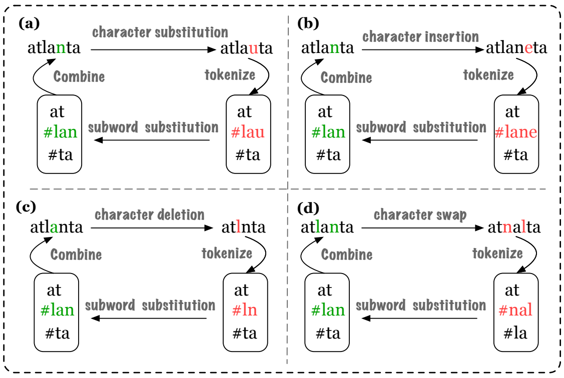

Figure 1: Subtoken substitution operation could achieve the same result as all four character modification operations.

Recently, some efficient and effective attacking methods have been proposed at token level (e.g. synonym substitution) (Guo et al., 2021) and sentence level (e.g. paraphrasing input texts) (Wang et al., 2020). However, this is not the case in character-level attack methods (e.g. mistyping words), which barely hinder human understanding and is thus a natural attack scenario. Most previous methods (Gao et al., 2018; Eger and Benz, 2020) achieve the character-level attack in a black box manner, which requires hundreds of attempts and the attack success rate is not good enough. White box attack methods are natural solutions to these drawbacks, but current character-level white box attack methods (Ebrahimi et al., 2018b,a) only work for models taking characters as input and thus fail on token-level transformer model.

Achieving character-level white box attack via single character modification is impossible for the transformer model, due to the gradient of characters being unavailable. We choose to implement the character-level attack via subtoken substitution based on the following two observations. (1) Nearly all transformer-based pre-training models adopt subword tokenizer (Sennrich et al., 2016), in which each word is split into subtokens containing one start subtoken and several subtoken attached

[†]Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7664 - 7676
December 7-11, 2022 ©2022 Association for Computational Linguistics

to it (attachable subtoken). (2) As shown in Figure 1, all character modifications (e.g. swap and insertion) can be achieved by subtoken substitution.

Based on the above observations, we propose CWBA, the first **C**haracter-level **W**hite-**B**ox **A**ttack method against transformer models via attachable subwords substitution. Our method mainly contains three steps: target word selection, adversarial tokenization, and subtoken search.

Since our CWBA requires specific words as input, finding the most vulnerable words is required. Our model first ranks the words according to the gradient of words from our adversarial goal. Then during the adversarial tokenization process, the top-ranked words are split into at least three subtokens, including a start subtoken and several attachable subtokens. Our CWBA method aims to replace these attachable subtokens to achieve character attack.

Due to the discrete nature of natural languages prohibits the gradient optimization of subtokens, we leverage the Gumbel-Softmax trick (Jang et al., 2017) to sample a continuous distribution from tokens and thus allow gradient propagation. The attachable subtokens are then optimized by a gradient descent method to generate the adversarial example. Meanwhile, to minimize the degree of modification, we also introduce visual and length constraints during optimization to make the replaced subtokens visually and length-wise similar.

Our CWBA method could outperform previous attack methods on both sentence level (e.g. sentence classification) and token level (e.g. named entity recognition) tasks in terms of success rate and edit distance. It is worth mentioning that CWBA is the first white box attack method applied to token-level tasks. Meanwhile, we demonstrate the effectiveness of CWBA against various transformer-based models. Human evaluation experiments verify our adversarial attack method is label-preserving. Finally, the adversarial training experiment shows that training with our adversarial examples would increase the robustness of models.

To summarize, the main contributions of our paper are as follows:

- To the best of our knowledge, CWBA is the first character-level white box attack method against transformer models.
- Our CWBA method is also the first white box attack method applied to token-level tasks.
- We propose a visual constraint to make the replaced subtoken similar to the original one.
- Our CWBA method could outperform the previous attack methods on both sentence-level tasks and token-level tasks. [1]

2 Related Work

2.1 White box attack method in NLP

White box attack methods could find the defects of the model with low query number and high success rate, which have been successfully applied to image and speech data (Madry et al., 2018; Carlini and Wagner, 2018). However, applying white-box attack methods to natural language is more challenging due to the discrete nature of the text. To search the text under the guidance of gradient and achieve a high success rate, Cheng et al. (2019b,a) choose to optimize in the embedding space and search the nearest word, which suffers from high bias problems. To further reduce the bias, Cheng et al. (2020) and Sato et al. (2018) restrict the optimization direction towards the existing word embeddings. However, the optimization process of these methods is unstable due to the sparsity of the word embedding space. Other methods try to directly optimize the text by gradient estimation techniques such as Gumbel-Softmax sampling (Xu et al., 2021; Guo et al., 2021), reinforcement learning (Zou et al., 2020), metropolis-hastings sampling (Zhang et al., 2019). Our CWBA adopts the Gumbel-Softmax technique for subtokens to achieve the character-level white-box attack.

2.2 Attack method against Transformers

Transformer-based (Vaswani et al., 2017) pretraining models (Devlin et al., 2019; Liu et al., 2019) have shown their great advantage on various NLP tasks. However, recent works reveal that these pretraining models are vulnerable to adversarial attacks under many scenarios such as sentence classification (Li et al., 2020), machine translation (Cheng et al., 2019b), text entailment (Xu et al., 2020) and part-of-speech tagging (Eger and Benz, 2020). Most of these methods achieve attack in the black box manner, which are implemented by character modification (Eger and Benz, 2020), token substitution (Li et al., 2020) or sentence paraphrasing (Xu et al., 2020). However, these black-box attack methods usually require hundreds of queries

[1] Code and data are available at https://github.com/THU-BPM/CWBA

to the target model and the success rate cannot be guaranteed. To alleviate these problems, some white-box attack methods have been proposed including token-level methods (Guo et al., 2021) and sentence-level methods (Wang et al., 2020). Different from these methods, our CWBA is the first character-level white-box attack method for transformer-based models.

3 Methods

In this section, we detail our proposed framework CWBA for the character-level white-box attack method. In the following content, we first give a formulation of our attack problem, followed by a detailed description of the three key components: target word selection, adversarial tokenization, and subtoken search.

3.1 Attack Problem Formulation

We formulate the adversarial examples as follows. Given an input sentence $\mathbf{x} = (x_1, x_2, ..., x_n)$ with length $|n|$, suppose the classification model $\mathbf{H}$ could predict the correct corresponding sentence or token label y such that $\mathbf{H}(\mathbf{x}) = y$. An adversarial example is a sample $\mathbf{x}'$ close to $\mathbf{x}$ but causing different model prediction such that $\mathbf{H}(\mathbf{x}') \neq y$.

The process of finding adversarial examples is modeled as a gradient optimization problem. Specifically, given the classification logits vector $\mathbf{p} \in \mathbb{R}^K$ generated by model $\mathbf{H}$ with K classes, the adversarial loss is defined as the margin loss:

$$\ell_{\text{adv}}(\mathbf{x}, y) = \max\left(\mathbf{p}_y - \max_{k \neq y} \mathbf{p}_k + \kappa, 0\right), \quad (1)$$

which motivates the model to misclassify $\mathbf{x}$ by a margin $\kappa > 0$. The effectiveness of margin loss has been validated in many attack algorithms (Guo et al., 2021; Carlini and Wagner, 2018).

Given the adversarial loss ℓ_{adv}, the goal of our attack algorithm can be modeled as a constrained optimization problem:

$$\min \ell_{\text{adv}}\left(\mathbf{x}', y\right) \quad \text{subject to } \rho\left(\mathbf{x}, \mathbf{x}'\right) \leq \epsilon, \quad (2)$$

where ρ is the function measuring the similarity between origin and adversarial examples. In our work, the similarity is measured using the edit distance metric (Li and Liu, 2007).

3.2 Target Word Selection

Since our attack method takes specific words as the target and performs pre-processing to these words, obtaining the most critical words for target task prediction is required. To find the most vulnerable words, we sort the words based on the l2 norm value of gradient towards adversarial loss in Eq 1:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argsort}}\left(\|\nabla_{x_1}\ell_{\text{adv}}\|_2, ..., \|\nabla_{x_n}\ell_{\text{adv}}\|_2\right) \quad (3)$$

where $\nabla_{x_j}\ell_{\text{adv}}$ is the gradient of the j-th token. Note that word x_j may be tokenized into several subtokens $[t_{j0}...t_{jn}]$, and its gradient is defined as the average gradient of these subtokens:

$$\|\nabla_{x_j}\ell\|_2 = \text{avg}\left(\|\nabla_{t_{j0}}\ell\|_2, ..., \|\nabla_{t_{jn}}\ell\|_2\right), \quad (4)$$

where the loss ℓ is the adversarial loss ℓ_{adv} in our work. Our CWBA would take the first N words from the sorted word list $\hat{\mathbf{x}}$ as targets, where N is a task-related hyperparameter.

3.3 Adversarial Tokenization

The selected words are required to split into subtokens before performing the character-level attack. We observe that the transformer tokenizer has the following two properties: (1) The correctly spelled words usually won't split or only split into a few subtokens. (2) The misspelled words are tokenized into more subtokens than the correctly spelled words. For example, the word *boston* won't be segmented but after single character modification, *bosfon* would be tokenized into three subtokens *bo*, *#sf* and *#on*. To keep the tokenization consistency during the attack, we propose the adversarial tokenizer which tokenizes the correctly spelled words into more subtokens than the transformer tokenizer.

To further improve the tokenization consistency during the attack process, our main principle is to make the subtokens as long as possible, since longer subtokens are more difficult to combine with characters to form new subtokens[2]. Specifically, our tokenization contains the following steps:

1. Find the longest subwords in the first half of the word to form the longest start subtoken.
2. Find the longest subwords in the second half of the word to form the longest end subtoken.
3. Tokenize the rest part with the transformer tokenizer to generate the middle subtokens.

After these steps, we obtain the longest start and end subtokens and our algorithm would substitute the middle subtokens, which keeps the maximum consistency of tokenization during the attack.

[2] More details and statistics are provided in the appendix

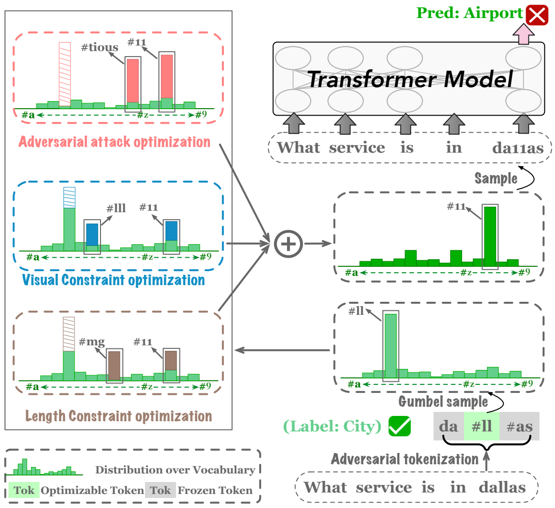
Pred: Airport
Transformer Model
What service is in da11as
Sample
#tious
#11
#a
#z
#9
Adversarial attack optimization
#lll
#11
Visual Constraint optimization
#mg
#11
Length Constraint optimization
#11
#ll
Gumbel sample
(Label: City)
da #ll #as
Adversarial tokenization
What service is in dallas
Distribution over Vocabulary
Tok Optimizable Token
Tok Frozen Token

Figure 2: Details of the subtoken search module.

3.4 Subtoken Search

After obtaining the vulnerable words and tokenizing them into subtokens in an adversarial way, the subtoken search module aims to find new subtokens for substitution to construct adversarial examples.

As shown in Figure 2, to allow gradient propagation, the target subtoken is first transformed from the discrete distribution to a continuous distribution by the Gumbel-softmax trick (Jang et al., 2017). Then the continuous distribution is optimized by three objectives: adversarial attack, visual constraint, and length constraint to search the adversarial examples with minimal modifications. The final adversarial examples could be sampled from the optimized Gumbel-softmax distribution.

Computing gradients using Gumbel-softmax
Since the origin subtoken input is represented in the discrete categorical distribution over vocabulary, the gradient could not be propagated directly. We adopt the Gumbel-softmax approximation to derive the soft estimation of the gradient.

Specifically, for any token $x_i \in \mathcal{V}$ from a fixed vocabulary $\mathcal{V} = \{1, ..., V\}$, we denote its one-hot distribution as ϕ_i. The Gumbel-softmax distribution π_i could be represented as follows:

$$(\pi_i)_j = \frac{\exp\left(\left(\phi_{i,j} + g_{i,j}\right)/T\right)}{\sum_{v=1}^{V} \exp\left(\left(\phi_{i,v} + g_{i,v}\right)/T\right)}, \quad (5)$$

where j indicates the jth token in the dictionary, $g_{i,j}$ is sampled from the uniform distribution $U(0, 1)$ to introduce randomness and T is the temperature parameter of the Gumbel-softmax distribution. ϕ_i could be updated by gradient through the Gumbel-softmax estimation π_i. Let $\mathbf{e}$ be the embedding lookup table of the transformer, the embedding vector of the distribution π_i can be defined as:

$$\mathbf{e}\left(\pi_i\right) = \sum_{j=1}^{V} \left(\pi_i\right)_j \mathbf{e}(j), \quad (6)$$

which is the input to the transformer model.

Adversarial attack objective
To search for the desired subtoken substitution which could mislead the model, an effective objective function is required. In practice, We adopt the margin loss in Eq 1. Given the whole sentence vector from Gumbel-softmax distribution $\mathbf{e}(\boldsymbol{\pi}) = \mathbf{e}\left(\pi_1\right), ..., \mathbf{e}\left(\pi_n\right)$, the adversarial loss is represented as $\ell_{\text{adv}}\left(\mathbf{e}(\boldsymbol{\pi}), y\right)$. Note that the Gumbel-softmax distribution is only applied on the target subtokens while the discrete distribution keeps unchanged on other subtokens such that $\pi_i = \phi_i$. For example, only the subword *#ll* in Figure 2 is sampled by Gumbel-softmax method while the subtokens *da* and *#as* maintains one-hot distribution.

However, the attack success of continuous distribution π_i does not guarantee the top one probability tokens in π_i could fool the target model. To reduce the gap between the distribution $\boldsymbol{\pi}$ and one-hot word distribution, we sample a discrete distribution from $\boldsymbol{\pi}$ and the adversarial loss is also applied on it, which is defined as follows:

$$(\hat{\pi}_i)_j = \begin{cases} \frac{(\pi_i)_j}{|(\pi_i)_j|}, & (\pi_i)_j = \max\left(\pi_i\right) \\ 0, & (\pi_i)_j \neq \max\left(\pi_i\right) \end{cases}. \quad (7)$$

The distribution $\hat{\boldsymbol{\pi}}$ is the one-hot distribution of the previous top probability tokens, of which the gradient is retained. Finally, our adversarial loss could be represented as:

$$\ell_{\text{adv}} = \ell_{\text{adv}}\left(\mathbf{e}(\boldsymbol{\pi}), y\right) + \lambda_{adv}\ell_{\text{adv}}\left(\mathbf{e}(\hat{\boldsymbol{\pi}}), y\right), \quad (8)$$

where the first term could quickly explore the candidate tokens and the second term further exploits the attack effect of the top one probability token. λ_{adv} is a hyper-parameter that balances the trade-off of exploration and exploitation.

Visual constraint objective
To minimize the edit distance (Eq. 2) caused by subtoken substitution, the visual constraint restricts the substituted subtoken visually similar to the original one, such as the *#ll* and *#11* in Figure 2.

In practice, we first generate the images of all subtokens in *helvetica* font. Then the pre-trained

ResNet50 network (He et al., 2016) is adopted to transform all the token images into vectors. Let $\mathbf{v}(i)$ be the visual embedding of i-th token in the vocabulary. Similar to Eq. 6, the visual embedding of distribution π_i could be represented as follows:

$$\mathbf{v}\left(\pi_i\right)=\sum_{j=1}^{V}\left(\pi_i\right)_j \mathbf{v}(j). \tag{9}$$

The visual constraint aims to minimize the gap of visual embedding between origin subtoken x_i and distribution π_i, which is defined as:

$$\ell_{\text{vis}}=\sum_{I}\|\mathbf{v}(\pi_i)-\mathbf{v}(x_i)\|_2, \tag{10}$$

where I is the set of all replaceable subtokens and $\| \|_2$ is the l2 normalization operation.

Length constraint objective

To further reduce character modifications, the length constraint objective aims to keep the length of subtoken unchanged during the attack process.

Similar to the visual constraint objective, the length of the distribution π_i could be defined as:

$$\mathbf{l}\left(\pi_i\right)=\sum_{j=1}^{V}\left(\pi_i\right)_j \mathbf{l}(j), \tag{11}$$

where $\mathbf{l}(i)$ is the length of the i-th token. And the length constraint loss could be represented similarly to the visual constraint loss:

$$\ell_{\text{len}}=\sum_{I}\|\mathbf{l}(\pi_i)-\mathbf{l}(x_i)\|_2. \tag{12}$$

Objective function

Our final objective is the combination of adversarial attack objective, visual constraint objective, and length constraint objective:

$$\mathcal{L}=\ell_{\text{adv}}+\lambda_{vis}\ell_{\text{vis}}+\lambda_{len}\ell_{\text{len}}, \tag{13}$$

where $\lambda_{vis}, \lambda_{len} > 0$ are hyperparameters that controls the degree of constraints. The final loss $\mathcal{L}$ is minimized using the gradient descent method.

Note that the number of attacked words is difficult to set in long sentences. So we search between two hyperparameters N_1 and N_2 until the adversarial loss could be well optimized. The process of our algorithm is summarized in algorithm 1.

Algorithm 1 CWBA Attack

```
1: Input words: $\mathbf{x} = (x_0, ..., x_n)$, label: $y$
2: Get sorted word list $\hat{\mathbf{x}} = [x_{top-1}, x_{top-2}, ...]$ from Eq 1
   based on the importance of words
3: for $k = k_1$ to $k_2$ do
4:     $\mathbf{s} = topk(\hat{\mathbf{x}})$ // Get the most important k words
5:     $\mathbf{h} \leftarrow []$ // Input token distribution
6:     for $x_i \in \mathbf{x}$ do
7:         if $x_i \in \mathbf{s}$ then
8:             $[t_0, t_1 ..., t_n] = \text{adv\_tokenize}(x_i)$ (Sec 3.3)
9:             $[\phi_0, \phi_1 ..., \phi_n] = \text{Onehot}([t_0, t_1 ..., t_n])$
10:            $[\pi_1, ., \pi_{n-1}] = \text{Gumbel}([\phi_1, ., \phi_{n-1}])$ (Eq 5)
11:            // Only search the middle subtokens
12:            $\mathbf{h} = \mathbf{h} \cup [\phi_0] \cup [\pi_1, ..., \pi_{n-1}] \cup [\phi_n]$
13:        else
14:            $[t_0, t_1 ..., t_n] = \text{transformer\_tokenize}(x_i)$
15:            $[\phi_0, \phi_1 ..., \phi_n] = \text{Onehot}([t_0, t_1 ..., t_n])$
16:            $\mathbf{h} = \mathbf{h} \cup [\phi_0, ..., \phi_n]$
17:        end if
18:    end for
19:    for $i = 0$ to MAX_ITER do
20:        Get loss $\mathcal{L}$ from Eq 13 based on input $\mathbf{h}$
21:        Update $\mathbf{h}$ using gradient descent method
22:    end for
23:    Get adversarial loss $\ell_{\text{adv}}$ from Eq 8
24:    // Whether Adversarial loss is well optimized
25:    if $\ell_{\text{adv}} < \kappa$ then
26:        Jump to line 3 // Attack fail, search more words
27:    end if
28:    Sample sentence $\tilde{\mathbf{x}}$ from $\mathbf{h}$
29:    if $f(\tilde{\mathbf{x}}) \neq y$ then
30:        // Attack success
31:    end if
32: end for
33: return $\tilde{\mathbf{x}}$
```

4 Experiments

We conduct extensive experiments on eight datasets across two tasks (sentence classification and token classification) and four transformer models (BERT, RoBERTa, XLNet, and ALBERT) to show the effectiveness of CWBA on white-box attack scenarios and give a detailed analysis to show its advantage.

4.1 Experiment Setup

Datasets. The sentence classification datasets include **DBPedia** (Lehmann et al., 2015), **AG News** (Zhang et al., 2015) for article/news categorization and **Yelp Reviews**(Zhang et al., 2015), **IMDB** (Maas et al., 2011) for sentiment classification. And the token classification datasets include **ATIS** (Tür et al., 2010), **SNIPS**[3] for slot filling and **CONLL-2003** (Tjong Kim Sang and De Meulder, 2003), **Ontonotes** (Pradhan et al., 2013) for named entity recognition (NER).

Baselines. For the sentence classification task, we adopt five competitive baselines which are various

[3] https://github.com/sonos/nlu-benchmark

in attack settings. **GBDA** (Guo et al., 2021) is a white-box attack method which performs token replacement under the gradient guidance. **BERT-Attack** (Li et al., 2020), **BAE** (Garg and Ramakrishnan, 2020) and **TextFooler** (Jin et al., 2020) aim to replace tokens in the black-box manner. **DeepWordBug** (Gao et al., 2018) is a black-box attack method which modifies characters of the most vulnerable words. Note that the edit distance of adversarial examples generated by token replacement is much higher than that of character modification.

For the token classification task, we adopt two baselines. **Zéroe** (Eger and Benz, 2020) explores several character-level black-box attack methods of which we choose the vision method as our baseline for its excellent attack effect. **DeepWordBug** is also adopted as a competitive baseline, which modifies the characters of keywords (e.g. entity in named entity recognition).

More details about these baselines are provided in the appendix.

Hyper-parameters. The input token distribution is optimized by Adam (Kingma and Ba, 2015) with a learning rate of 0.3 for 100 iterations (token classification) or 300 iterations (sentence classification). The margin κ of adversarial loss is set to 7 for sentence classification and 5 for token classification. And the T of Gumbel-Sampling (Eq. 5) is set to 1. The loss weights λ_{adv}, λ_{vis} and λ_{len} (Eq.13) are set to 1, 0.1, 2 respectively. The N_1 and N_2 are set to 2, 15 and 1, 2 for sentence classification and token classification tasks respectively.

Models. We attack four transformer models with our `CWBA` method: BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020). We first fine-tune these models on target datasets and then perform the attack. All of these models utilize the subword tokenization method. The BERT tokenizer adds a prefix ## to the attachable subwords. The tokenizer of RoBERTa adds a prefix Ġ to the start subwords. And the tokenizer of XLNet and ALBERT adds a prefix _ to the start subtokens.

4.2 Quantitative Evaluation.

Sentence-level attacks. Table 1 shows the attack performance on the sentence classification task with the BERT classifier. Following the previous works (Guo et al., 2021), we randomly select 1000 inputs from the test set as attack targets. Our method searches the number of attacked words between N_1 and N_2 until the attack succeeds. The adversarial accuracy (Adv.Acc.) is the accuracy of the last searched examples. The Edit Dist represents the sum of edit distances for all modified words.

Datasets	Clean Acc.	Attack Alg.	Adv.Acc.	#Queries	Edit Dist
AG News	95.1	CWBA(ours)	**3.2**	6.1	**17.3**
		DeepWordBug	23.7	319	23.4
		GBDA	3.5	**5.8**	76.3
		BERT-Attack	10.6	213	83.4
		BAE	13.0	419	65.2
		TextFooler	12.6	357	97.5
Yelp	97.3	CWBA(ours)	**4.0**	**7.2**	**18.5**
		DeepWordBug	27.7	543	19.8
		GBDA	4.4	7.6	84.1
		BERT-Attack	5.1	273	95.3
		BAE	12.0	434	63.1
		TextFooler	6.6	743	74.6
IMDB	93.0	CWBA(ours)	**5.4**	8.5	**23.4**
		DeepWordBug	28.4	134	24.5
		GBDA	5.6	**5.2**	84.5
		BERT-Attack	11.4	454	91.3
		BAE	24.0	592	88.2
		TextFooler	13.6	1134	77.1
DBPedia	99.2	CWBA(ours)	**6.9**	**5.3**	**19.1**
		DeepWordBug	19.4	453	34.5
		GBDA	7.1	5.6	79.1
		BERT-Attack	8.5	487	86.5
		BAE	10.4	398	94.3
		TextFooler	9.5	829	83.4

Table 1: Attack result on sentence classification datasets with finetuned BERT classifiers.

Datasets	Clean F1.	Attack Alg.	Adv.F1.	success rate	#Queries	Edit Dist
ATIS	96.7	CWBA(ours)	**9.3**	**90.0**	**2.4**	**2.0**
		DeepWordBug	27.4	71.2	58.8	3.4
		Zéroe	15.2	84.3	23.4	3.8
SNIPS	95.8	CWBA(ours)	**15.3**	**86.3**	**2.6**	**1.9**
		DeepWordBug	29.5	70.1	43.9	3.5
		Zéroe	25.3	73.5	56.1	3.0
CONLL2003	93.2	CWBA(ours)	**14.4**	**87.5**	**3.0**	**2.9**
		DeepWordBug	38.4	62.3	47.9	7.5
		Zéroe	33.5	66.5	49.8	7.1
OntoNotes	87.6	CWBA(ours)	**5.8**	**96.2**	**2.1**	**2.1**
		DeepWordBug	26.3	73.9	26.1	3.5
		Zéroe	18.4	83.4	31.2	3.1

Table 2: Attack result on token classification datatsets with finetuned BERT classifiers.

Overall, our `CWBA` outperforms the previous baselines in terms of adversarial accuracy and edit distance on all datasets. More specifically, compared to the previous best methods, our `CWBA` could further reduce the model's accuracy and the edit distance by 0.3 and 6.0 on average respectively. Meanwhile, our required query number is similar to the GBDA model and far less than other black-box methods. Also, our `CWBA` outperforms the character-level attack method DeepWordBug by a large gap (20.0 adversarial accuracy on average), which demonstrates the advantages of the white-box attack.

Architecture	Datasets	Clean Acc.(F1.)	Adv.Acc.(F1.)	#Queries	Edit Dist
ALBERT	**ATIS**	96.3	**2.2**	**2.1**	**2.0**
	OntoNotes	87.3	**0**	**1.8**	**1.6**
	AG News	93.8	**2.8**	**5.6**	**14.5**
	Yelp	96.9	**3.7**	**5.5**	**15.6**
XLNet	**ATIS**	96.2	<u>16.6</u>	4.8	<u>3.2</u>
	OntoNotes	87.6	7.1	3.0	<u>2.5</u>
	AG News	94.6	4.6	5.7	16.8
	Yelp	96.5	5.5	6.2	19.1
RoBERTa	**ATIS**	96.6	13.0	<u>5.8</u>	2.5
	OntoNotes	87.7	<u>17.0</u>	<u>4.3</u>	2.2
	AG News	94.7	<u>7.5</u>	<u>7.2</u>	<u>20.1</u>
	Yelp	97.2	<u>8.4</u>	<u>6.9</u>	<u>21.1</u>

Table 3: Attack result on three different transformer based pretrained language models.

Token-level attacks. The attack performance for the BERT classifier towards four token classification datasets is shown in Table 2. Similarly, we randomly select 1000 inputs from the test set as attack targets. For SNIPs and ATIS where the size of the test set is below 1000, the whole test set is selected. The success rate is the percentage of entities whose predictions are changed after the attack. And we report the average edit distance for entities.

In general, our CWBA surpasses the previous black-box attack methods by a large margin. To be specific, our CWBA could achieve a higher attack success rate (13.08% on average) than the previous best method with a smaller edit distance (1.9 on average). Furthermore, our required query number is much less than the previous methods (32.8 on average). These experimental results demonstrate the effectiveness of our white-box attack method for the token classification task.

Attack different transformer models. To illustrate the generalizability of our CWBA towards transformer models, we report the attack result on three different transformer-based models in Table 3. We select two benchmarks for sentence and token classification respectively, where all dataset settings are the same as above.

It could be observed that our CWBA has an excellent attack performance on these transformer models. Meanwhile, we found that ALBERT is more vulnerable to attacks than other transformer models while even the best-performing models RoBERTa and XLNet are easy to be attacked. These experimental results illustrate that the current pre-training models are vulnerable to character modifications.

Ablation study. We conduct ablation studies to show the effectiveness of different modules of CWBA to the overall attack performance. The experiments are performed on CONLL and AG News datasets with the BERT classifier. CWBA w random word selection random selects target words instead of searching by gradient. CWBA w/o visual constraint and w/o length constraint removes the visual and length constraint respectively. CWBA w/o visual & length constraint only keeps the adversarial attack object without any constraint.

Dataset	Technique	Adv Acc.(F1.)	#Queries	Edit Dist
CONLL	CWBA	9.1	4.5	**2.7**
	w random word selection	59.4	7.3	2.9
	w/o visual constraint	8.7	3.8	3.3
	w/o length constraint	9.3	4.2	3.2
	w/o visual & length constraint	**6.2**	**3.9**	4.7
AG News	CWBA	3.2	6.1	**17.3**
	w random word selection	54.2	13.4	26.7
	w/o visual constraint	2.7	5.6	25.6
	w/o length constraint	3.0	5.8	26.7
	w/o visual & length constraint	**2.4**	**4.8**	37.8

Table 4: Ablation study of how different modules contribute to the attack performance on BERT classifier.

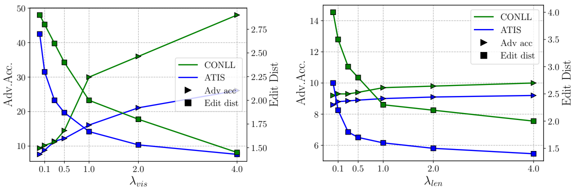

Figure 3: Adversarial accuracy and edit distance during the growth of λ_{vis}(left) and λ_{len}(right).

Our observations from the experimental results in Table 4 are as follows: (1) The target word selection module has a huge impact on the attack success rate. (2) The visual constraint could significantly reduce the edit distance while sacrificing a little attack performance. Further analysis would help us balance the attack success rate and edit distance. (3) The length constraint has little effect on the attack performance but could effectively reduce the edit distance. In general, all our modules have positive effects on the attack performance.

4.3 Analysis

Effectiveness of visual and length constraint. To further investigate how the visual and length constraint contributes to the attack performance, we visualize the changing trend of adversarial accuracy and edit distance when λ_{vis} and λ_{len} grow from zero in Figure 3. When tuning one hyperparameter, the other hyperparameter would be set to 0.1. These experiments are performed on two token classification datasets: CONLL and ATIS.

It can be seen that with λ_{vis} increases, the edit distance decreases while the attack performance

Dataset	Ori.Acc.(F1.)	Adv.Acc.(F1.)	Identification	Correction
ATIS	97.2	94.5	99.1	93.3
SNIPS	96.3	92.1	97.8	90.5
Yelp	91.2	82.2	92.4	83.1
AG News	90.3	80.8	93.4	82.4

Table 5: Human evaluation for the ability to identify and correct adversarial examples.

Dataset	Technique	Adv Acc.(F1.)	#Queries	Edit Dist
ATIS	`CWBA`	**9.3**	2.4	**2.0**
	`CWBA`+ Adv.Training	58.5	9.3	4.8
	DeepWordBug	**27.4**	58.8	**3.4**
	DeepWordBug + Adv.Training	44.8	38.4	4.8
	Zéroe	**15.2**	23.4	3.8
	Zéroe + Adv.Training	56.3	19.5	**3.2**
AG News	`CWBA`	**3.2**	6.1	**17.3**
	`CWBA`+ Adv.Training	60.3	12.4	31.2
	DeepWordBug	**23.7**	319	**23.4**
	DeepWordBug + Adv.Training	57.9	412	33.5

Table 6: Model robustness improvement after adversarial learning with the generated adversarial examples.

drops (the accuracy after the attack increases). Meanwhile, with the increase of λ_{len}, the attack performance is almost unchanged while the edit distance still reduces. So we conclude that λ_{vis} influences more on the attack success rate than λ_{len}, which may be because the transformer-based language models are robust to visual similarity changes to some extent. These experiment results inspire us to adopt a larger λ_{len} than λ_{vis}.

Human evaluation. To examine whether the attacked text preserves its original label, we set up human evaluations to measure the quality of the generated text. We first ask human judges to make predictions on both the original and attacked texts. Then we ask them to identify and correct the modified words in the attacked text. The evaluations are conducted on 100 selected sentences from ATIS, SNIPs DBPedia, and AG News datasets respectively. For each dataset, we ask three human evaluators to measure the quality of examples.

The human evaluation results are presented in Table 5. And we could observe that: (1) Human judges could predict most of the attacked text correctly, which demonstrates our generated examples are label-preserving. (2) Although most of the modified words could be identified, most of these misspelled words could be corrected by humans, which does not hinder understanding. (3) Perturbations on the words in sentence classification datasets are harder to identify and correct than that in token classification datasets because of the larger edit distance.

Adversarial training. To further explore whether the adversarial examples could help improve the model's robustness, we perform an adversarial training experiment by training the model with the combination of the original and the adversarial examples. Specifically, the adversarial examples are selected from the attacked texts of the ATIS and DBPedia training sets. Furthermore, we compare the attack results towards the original model and the adversarially trained model using our `CWBA` and other character-level attack methods.

We present the adversarial training results in Table 6. Overall, the robustness of our model towards character-level attacks improves tremendously after the adversarial training. Specifically, the attack success rate of our `CWBA` decreases drastically (53.15 on average), while the required editing distance and the number of queries become much larger (6.6 and 8.4 on average). Similarly, the attack performance of other character-level attack methods drops severely in all metrics. These experimental results indicate that our generated texts could preserve the origin label.

Tokenization analysis. Our `CWBA` works on the tokenized subtokens. However, the adversarial texts are re-tokenized before being fed to the model, where the re-tokenization result may not be the same set of origin subtokens. For example, the subtokens *bo-*, *sl-* and *on* would be re-encoded into *bos-* and *lon*. We further analyze how tokenization inconsistency affects attack performance.

In practice, we observe that the tokenization inconsistency doesn't impact the attack performance by much. Specifically, 31.2% words are not re-tokenized to the same subtoken set but only lead to 3% attack failures, which indicates the re-tokenized examples are still adversarial. Similar observations are reported in previous works (Guo et al., 2021).

We further analyze the reason of attack failures and find 65% of failed examples are not optimized well and tokenization inconsistency leads to 35% attack failures, which indicates that tokenization inconsistency has a limited impact on our method.

Case study. To intuitively show the effectiveness of `CWBA`, we select three cases to compare the original and adversarial texts. These cases are sampled from AG News, Yelp (sentence classification), and

Dataset			Label
AG News	Ori	fund pessimism grows new york (cnn / money) - money managers are growing more pessimistic about the economy, corporate profits and us stock market returns, according to a monthly survey by merrill lynch released tuesday.	Business
	Adv	fund pessimism grows new york (cnת / money) - money managers are growing more pessimistic about the econ0my, corporate profits and us stock market returns, according to a monthly survey by merrill lynch released tuesday.	Sci/Tech
Yelp	Ori	seattle may have just won the 2014 super bowl, but the steelers still rock with six rings, baby!!! just stating what all steeler fans know : a steel dynasty is still unmatched no matter what team claims the title of current super bowl champs.	Positive
	Adv	seattle may have just won the 2014 super bowl, but the steelers still robk with six riigs, bacy!!! just staging what all steeler fans know : a steel dynasty is still unmatched no matter what team claims the title of current super bowl champs.	Negative
OntoNotes	Ori	So far, the French have failed to win enough broad - based support to prevail. Prediction span: French	NORP
	Adv	So far, the Fr€nch have failed to win enough broad - based support to prevail. Prediction span: Fr€nch	ORG

Table 7: The generated adversarial examples. The origin label is the correct prediction and the adversarial label is adverse prediction. The first two examples are from sentence classification task while the third case is from the token classification task. The target tokens and labels for token classification are underlined.

OntoNotes (token classification) datasets.

As seen in Table 7, the generated adversarial sentences are semantically consistent with their original texts, while the target model makes incorrect predictions. Meanwhile, we could observe that many words are visually similar during the attack (e.g. *cnn* and *cn*ת), which shows the effectiveness of our visual constraint. The number of words to be attacked for sentence-level tasks is larger than the token-level tasks, and the concrete number is also uncertain (two in the first case and four in the second case). For token classification tasks like NER, the attacked words are usually the entities.

5 Conclusions

In this paper, we propose CWBA, the first character-level white-box attack method for transformer models. We substitute the attachable subtokens to achieve character modification. The Gumbel-Softmax technique is adopted to allow gradient propagation. Meanwhile, the visual and length constraint help preserve the semantics of adversarial text. Experiments on both sentence-level and token-level tasks on various transformer models demonstrate the effectiveness of our method.

Acknowledgement

The work was supported by the National Key Research and Development Program of China (No.2019YFB1704003), the National Nature Science Foundation of China (No.62021002), Tsinghua BNRist and Beijing Key Laboratory of Industrial Big Data System and Application.

Limitations

The major limitation of CWBA has been discussed in the tokenization analysis part in section 4.3: the generated words may be re-tokenized into different subtokens. Although most examples are still adversarial, it introduces uncontrollability. We hope future works could introduce more constraints to alleviate this problem.

Also, we achieve character modification by subword substitution, but not all combinations of characters exist in the vocabulary. Therefore, the effect of our attack method depends on the size of the vocabulary.

References

Nicholas Carlini and David A. Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 1–7. IEEE Computer Society.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Minhao Cheng, Wei Wei, and Cho-Jui Hsieh. 2019a. Evaluating and enhancing the robustness of dialogue systems: A case study on a negotiation agent. 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 3325–3335, Minneapolis, Minnesota. Association for Computational Linguistics.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3601–3608.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019b. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics, pages 4324–4333, Florence, Italy. 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.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018a. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

Steffen Eger and Yannik Benz. 2020. From hero to zéroe: A benchmark of low-level adversarial attacks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 786–803, Suzhou, China. 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.

Yotam Gil, Yoav Chai, Or Gorodissky, and Jonathan Berant. 2019. White-to-black: Efficient distillation of black-box adversarial attacks. *arXiv preprint arXiv:1904.02405*.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Xuming Hu, Fukun Ma, Chenyao Liu, Chenwei Zhang, Lijie Wen, and Philip S Yu. 2021a. Semi-supervised relation extraction via incremental meta self-training. In *Proc. of EMNLP: Findings*.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and S Yu Philip. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *Proc. of EMNLP*, pages 3673–3682.

Xuming Hu, Chenwei Zhang, Yawen Yang, Xiaohe Li, Li Lin, Lijie Wen, and S Yu Philip. 2021b. Gradient imitation reinforcement learning for low resource relation extraction. In *Proc. of EMNLP*, pages 2737–2746.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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 *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

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

Yujian Li and Bi Liu. 2007. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1091–1095.

Shuliang Liu, Xuming Hu, Chenwei Zhang, Shu'ang Li, Lijie Wen, and Philip S. Yu. 2022. Hiure: Hierarchical exemplar contrastive learning for unsupervised relation extraction. In *Proc. of NAACL*.

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

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4323–4330. ijcai.org.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794.

Yongliang Shen, Xiaobin Wang, Zeqi Tan, Guangwei Xu, Pengjun Xie, Fei Huang, Weiming Lu, and Yueting Zhuang. 2022. Parallel instance query network for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 947–961.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3936–3942. International Joint Conferences on Artificial Intelligence Organization.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015. Effective lstms for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, Bing Qin, et al. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565. Citeseer.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop, SLT 2010, Berkeley, California, USA, December 12-15, 2010*, pages 19–24. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. 2020. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6134–6150, Online. Association for Computational Linguistics.

Lei Xu, Ivan Ramirez, and Kalyan Veeramachaneni. 2020. Rewriting meaningful sentences via conditional bert sampling and an application on fooling text classifiers. *arXiv preprint arXiv:2010.11869*.

Ying Xu, Xu Zhong, Antonio Jimeno Yepes, and Jey Han Lau. 2021. Grey-box adversarial attack and defence for sentiment classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4078–4087, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569, Florence, Italy. Association for Computational Linguistics.

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

Xin Zhang, Guangwei Xu, Yueheng Sun, Meishan Zhang, Xiaobin Wang, and Min Zhang. 2022. Identifying chinese opinion expressions with extremely-noisy crowdsourcing annotations. In *Proc. of ACL*, pages 2801–2813.

Xin Zhang, Guangwei Xu, Yueheng Sun, Meishan Zhang, and Pengjun Xie. 2021. Crowdsourcing learning as domain adaptation: A case study on named entity recognition. In *Proc. of ACL*, pages 5558–5570.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*.

Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. 2020. A reinforced generation of adversarial examples for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3486–3497, Online. Association for Computational Linguistics.

A Attachable subwords analysis

Subwords len	Num in Vocab	Potential Num	Ratio
1	26	26	1
2	438	676	0.65
3	1438	17576	0.08
4	1573	456976	0.03
5	695	11876696	0.5×10-5

Table 8: The subword number of different lengths in the vocabulary.

To better illustrate the principles of adversarial tokenization, we list the statistics for the number of attachable subwords with different lengths in Table 8. We can see that with the length increases, the proportion of subwords in vocabulary among all potential subwords with the same length is getting smaller. For example, all the 26 attachable subwords with length 1 (*#a* - *#z*) are in the vocabulary list, but some subwords with length 2 (*#rz*) doesn't.

Based on these observations, we conclude that the longer subtokens are more difficult to combine with characters to form new subwords, which is the principle of the adversarial tokenization module.

B Details of baselines

Token classification task. Token classification is a natural language understanding task in which a label is assigned to some tokens in the text. Some popular token classification subtasks are Named Entity Recognition (NER) (Zhang et al., 2021; Shen et al., 2021; Tan et al., 2021; Shen et al., 2022; Zhang et al., 2022) and Slot Filling (Chen et al., 2019; Zhang et al., 2017).

Since the token classification model generates a label for each token, the token itself cannot be replaced, and the structure of the sentence also cannot be modified. Therefore, neither sentence-level (Xu et al., 2020) nor word-level attacks (Eger and Benz, 2020) can be applied to the token classification task, only character-level attacks are available in this scenario.

The current character-level attack methods can be divided into two categories. The first class of methods performs a white-box attack against a model taking characters as input. The most representative methods is **HotFlip** (Ebrahimi et al., 2018b). Other works are mainly variants of HotFlip (Ebrahimi et al., 2018a; Gil et al., 2019). Another class of methods performs black-box attacks on the model, where main representative methods are **DeepWordBug** (Gao et al., 2018) and **Zéroe** (Eger and Benz, 2020). These methods do not require the input to the model to be characters. Since our approach attacks models with word-level input, we mainly take DeepWordBug and Zéroe as our baselines.

Sentence classification task Sentence classification is one of the simplest NLP tasks in the Natural Language Processing field that have a wide range of applications including sentiment analysis (Tang et al., 2015, 2014) and relation extraction (Hu et al., 2021a,b, 2020; Liu et al., 2022).

Since the sentence classification method outputs a label for the whole sentence, both word-level attacks and character-level attacks can be performed on it. Word level attack method is the most widely used text attack method, which can be classified into the black-box method and white-box method. The classical black-box approach mainly uses a rule-based approach to do synonym replacement, of which the most representative is **TextFooler** (Jin et al., 2020). Recent methods like **BERT-Attack** (Li et al., 2020) and **BAE** (Garg and Ramakrishnan, 2020) replace words in context with the help of semantic information from the pre-trained model (Devlin et al., 2019). The representative work of white-box attack is **GBDA** (Guo et al., 2021), which performs token replacement under the gradient guidance. In this work, we adopt token-level attack methods TextFooler, BAE, BERT-Attack, GBDA and character-level attack method DeepWordBug as our baselines.