Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.042090

ARTICLE

Chinese Cyber Threat Intelligence Named Entity Recognition via RoBERTa-wwm-RDCNN-CRF

Zhen Zhen[1] **and Jian Gao**[1,2,*]

[1]School of Information Network Security, People's Public Security University of China, Beijing, 100038, China

[2]Key Laboratory of Safety Precautions and Risk Assessment, Ministry of Public Security, Beijing, 102623, China

*Corresponding Author: Jian Gao. Email: gaojian@ppsuc.edu.cn

Received: 18 May 2023 Accepted: 15 September 2023 Published: 31 October 2023

ABSTRACT

In recent years, cyber attacks have been intensifying and causing great harm to individuals, companies, and countries. The mining of cyber threat intelligence (CTI) can facilitate intelligence integration and serve well in combating cyber attacks. Named Entity Recognition (NER), as a crucial component of text mining, can structure complex CTI text and aid cybersecurity professionals in effectively countering threats. However, current CTI NER research has mainly focused on studying English CTI. In the limited studies conducted on Chinese text, existing models have shown poor performance. To fully utilize the power of Chinese pre-trained language models (PLMs) and conquer the problem of lengthy infrequent English words mixing in the Chinese CTIs, we propose a residual dilated convolutional neural network (RDCNN) with a conditional random field (CRF) based on a robustly optimized bidirectional encoder representation from transformers pre-training approach with whole word masking (RoBERTa-wwm), abbreviated as RoBERTa-wwm-RDCNN-CRF. We are the first to experiment on the relevant open source dataset and achieve an F1-score of 82.35%, which exceeds the common baseline model bidirectional encoder representation from transformers (BERT)-bidirectional long short-term memory (BiLSTM)-CRF in this field by about 19.52% and exceeds the current state-of-the-art model, BERT-RDCNN-CRF, by about 3.53%. In addition, we conducted an ablation study on the encoder part of the model to verify the effectiveness of the proposed model and an in-depth investigation of the PLMs and encoder part of the model to verify the effectiveness of the proposed model. The RoBERTa-wwm-RDCNN-CRF model, the shared pre-processing, and augmentation methods can serve the subsequent fundamental tasks such as cybersecurity information extraction and knowledge graph construction, contributing to important applications in downstream tasks such as intrusion detection and advanced persistent threat (APT) attack detection.

KEYWORDS

Cybersecurity; cyber threat intelligence; named entity recognition

1 Introduction

In recent years, cyber attacks have evolved into a new way for illegals to profit and cause negative impacts, which has caused great damage and threats to countries worldwide. For example, in 2010, Stuxnet attacked Iran's nuclear industry infrastructure, causing the failure of about 1,000 uranium

enrichment centrifuges and delaying Iran's nuclear program. In 2015, a Ukrainian power company's supervisory control and data acquisition system was compromised, causing hours-long power outages for 225,000 customers. In 2017, attackers exploited a vulnerability in Equifax's internal system. They successfully obtained sensitive information, including names, dates of birth, social security numbers, driver's license numbers, and other information, resulting in the theft of personal information from 143 million Americans. In 2020, attackers exploited SolarWinds' software update channels to successfully breach the networks of several government agencies, large corporations, and security companies, stealing a large amount of sensitive information.

To combat the attacks mentioned above, the sharing of CTI has emerged. Although there are many CTIs, the automation of natural language-based CTIs still faces challenges due to the complexity of natural language processing (NLP) and the specialized nature of the cyber security domain [1]. Although we have much human intelligence, we cannot make it into knowledge computers can process and utilize. It still requires a lot of manual intervention, which is time-consuming and has a limited effect on correlating relevant expertise on a large scale. As a key step in text understanding and processing, NER can extract important entities from text to form structured data based on entity types defined in advance, which brings hope for the structured transformation of CTI data.

Research on NER for Chinese CTIs is still relatively small, and related studies and datasets mostly focus on English intelligence [2]. Although the number of Chinese CTIs and the knowledge within them cannot be underestimated, few studies exist on them [3–9]. Currently, the main challenges facing the field are as follows:

(1) In the field of Chinese NER, pre-processing is a challenging yet crucial step. The limitations imposed by the length of input texts make it prone to unintentional truncation of targeted entities. The specialty of CTIs further complicates the situation due to the irregular writing styles by subscribers, textual errors caused by crawlers, and the presence of Chinese-English mixed content. However, none of the previous papers [3–9] describe the processing process, which brings difficulties for reproduction. Based on this, this paper details the proposed pre-processing method for Chinese CTI in Section 4.1, mainly including Section 4.1.1, Two-Stage Multi-Level Text Truncation Method, and Section 4.1.2, Data Augmentation.

(2) Infrequent long English words mixed in the Chinese CTIs present a significant challenge. In contrast to daily texts or texts in other domains, cyber practitioners tend to employ lengthy English words that are infrequent in typical language contexts. Examples of such words include computer file paths like "C:/Users/Public/Downloads/Winvoke.exe", tools names like "cobaltstrikebeacon", and so on. However, in Chinese PLMs, the tokenization of these English words usually results in subwords. Consequently, the tokens for an English word in the Chinese CTIs tend to be composed of multiple subword tokens, which poses great challenges for models to capture the complete information contained within the English word.

(3) Scarcity of labeled data. As the problem of NER in professional fields, there is a lack of annotation data for Chinese CTI NER. The authors of [5,6] attempted to solve the problem from the data annotation stage using active learning, but to our knowledge, no scholars have yet proposed an effective method from the perspective of data augmentation.

(4) Lack of exploration for PLMs. In recent years, PLMs have developed rapidly. However, in the Chinese CTI NER, the exploration of PLMs only stops at BERT [9] and a lite BERT (ALBERT) [10].

For challenge (1), the proposed two-stage multi-level text truncation approach for Chinese CTI is described in detail in Section 4.1.1. For challenge (2), we adopt RDCNN, which has a bigger

contextual scope than a convolutional neural network (CNN) and less information loss than BiLSTM. For challenge (3), we present the proposed data augmentation method detailedly in Section 4.1.2. For challenge (4), we briefly introduce the development of PLMs in Section 2.2 and analyze their performance in Section 5.4.

Our contributions can be summarized as follows:

- The RoBERTa-wwm-RDCNN-CRF model is proposed, which enables an end-to-end NER in Chinese CTI, and its effectiveness exceeds the current best model in the field.
- Experiments are conducted on the latest and the first open-source dataset, and the related code is provided for future research work.
- For the first time, the data pre-processing process of Chinese CTIs is elaborated to provide a reference for related researchers.
- A data augmentation method in Chinese CTI is proposed and validated for the first time. An attempt is made to alleviate the problem of lack of labeled samples due to its overspecialization from the perspective of data augmentation.
- The effectiveness of the RDCNN model in Chinese CTI NER is explored in depth to verify the effectiveness of its dilated convolution module, residual module, the influence of the number of modules, dilation coefficient, number of filters, and kernel size.
- The effectiveness of different PLMs is first investigated in Chinese CTI NER.

The article's structure is as follows: Section 2 introduces the related literature. Section 3 defines the Chinese CTI NER problem. Section 4 presents the pre-processing method and the proposed RoBERTa-wwm-RDCNN-CRF model. Section 5 shows the experimental results, and we conclude in Section 6.

2 Related Work

In this section, we first introduce the work related to Chinese CTI NER, followed by the development of PLMs in NLP in recent years.

2.1 Chinese CTI NER

Deep learning (DL) is a subfield of machine learning that focuses on training deep neural networks to achieve high-level abstraction and learning of data. In recent years, with the help of DL, many research areas, including cybersecurity, have achieved great improvement [11–15]. Because the CTI NER area is relatively new, most researchers chose to use the prevailing DL due to its good performance with automatic feature extraction, especially in Chinese CTI NER, which is fresher even than the English. So, we choose to use DL in our research and will elaborate on the DL research in the area. As far as we know, most of the research on CTI NER and its application focuses on English intelligence [16–23], and relatively little research has been conducted on Chinese intelligence [3–9,24]. The details are as follows.

A few of them perform a coarse classification task simpler than the NER task [3,24]. In 2019, Long et al. proposed using a neural network model to solve the problem of extracting IOC from Chinese threat intelligence, which was highly dependent on feature engineering. Specifically, it used BiLSTM based on attention mechanism, multi-headed attention mechanism, and word features to achieve an 81.1% F1-score on the Chinese dataset [3]. Although done on Chinese intelligence, it does not accomplish the NER task and only extracts limited information. In 2020, Tsai et al. proposed a system called CTI ANT, which can automatically classify threat intelligence and mine key hot

words and Chinese advanced persistent threat reports according to the techniques offered in MITRE's ATT&CK framework [24]. Similar to [3], although it targets Chinese CTIs, it performs a simpler classification task, and its coarse-grained classification results cannot serve downstream applications.

For the named entity recognition task, Qin et al. proposed a CNN-BiLSTM-CRF model using a fused feature template that recognizes local contextual information in 2019, followed by a neural network model for global information, achieving an F1-score of 86% [4]. To solve the problem of a lack of labeled corpus for training, Li et al. proposed a triadic approach for cybersecurity knowledge extraction incorporating adversarial active learning in 2020 [5]. The model fuses the dynamic attention mechanism and BiLSTM-long short-term memory (LSTM) model to achieve joint extraction of entities and relations, and trains discriminator models based on adversarial networks to filter high-quality data to be labeled. However, the BiLSTM [4,5] used still has the problem of information loss when it deals with long sentences. In 2021, Xie et al. proposed a model to identify urgently labeled entities from uncertainty, confidence, and diversity perspectives using the active learning strategy [6]. Same to [5], although better results were achieved, the annotation model requires constant manual alternation with the model, which is inefficient. To solve the problem of Chinese and English mixed entities in Chinese CTI NER, Han et al. proposed a model based on multitask learning with adversarial training in 2021, which achieved a 91.81% F1-score. To solve the problem of unbalanced entity distribution, Guo et al. proposed the entity extraction model incorporating focal loss in 2022, which improved by 7.07% and 4.79% in F1-score compared with the mainstream BiLSTM and BiLSTM-CRF models, but there is still room for improvement in the extraction of entities with few samples [7]. Reference [6,7] respectively incorporate more useful prior knowledge into the model from different perspectives, but in the face of the problem of insufficient data, data augmentation methods have not been attempted in this field.

Subsequently, with the rise of PLMs, Xie et al. proposed BERT-RDCNN-CRF in 2020 [9]. While the team achieved an F1-score of 89.88%, there are certain areas where their work could be improved. Firstly, they did not provide a public dataset, a comprehensive parameter reference for RDCNN, or conduct in-depth ablation experiments and analysis to assess the model's effectiveness. Additionally, they did not explore the prevailing pre-trained language models (PLMs) extensively. Although achieving an F1-score of 89.88%, the team did not provide an in-depth analysis and explanation of the model's effectiveness. Yang et al. proposed an advanced persistent threat attack entity identification model based on BERT and BiLSTM-CRF in 2022 [8], which achieved an F1-score of 90.06%. To improve efficiency, Zhou et al. proposed the ALBERT-BiLSTM-CRF model in 2023 [10], which arrived at 92.21% F1-score and had lower time and resource costs. Further exploration of the PLMs in the field of Chinese CTI NER is needed, along with the model's performance on additional datasets.

2.2 PLMs

In recent years, PLM has been a popular research direction in NLP. It can learn rich linguistic knowledge, including vocabulary, syntax, and semantics, through unsupervised training on large-scale text data, effectively improving the performance of NLP tasks.

BERT is a fundamental PLM proposed by Google in 2018 to address many challenges in NLP, such as semantic understanding, text classification, machine translation, and other tasks [25]. BERT is a deep bi-directional encoder based on the Transformer structure that uses two pre-training tasks, namely masked language model (MLM) and next sentence prediction (NSP), to train the model. It uses a large-scale text corpus for unsupervised learning in the pre-training phase, which can learn general

language knowledge and thus performs well in various NLP tasks. In addition, BERT is fine-tuned for different tasks, which can be quickly adapted to new tasks and achieves leading performances on many tasks. In the field of NLP, many scholars have applied BERT to different tasks, such as sentiment analysis, reading comprehension, and NER. Since then, many researchers have also improved BERT and proposed many variant models.

RoBERTa is a BERT-based PLM introduced by Facebook Artificial Intelligence Research [26]. Compared to BERT, RoBERTa uses larger text data, longer training time, and other optimization techniques in the pre-training phase, resulting in better performance in various NLP tasks. Specifically, RoBERTa uses a larger dataset in the pre-training stage and does not use the NSP task. Instead, RoBERTa uses more MLM tasks, some of which are executed on multiple embedding layers at different locations and positions, thus making the model more sensitive to different contexts. In addition, RoBERTa uses dynamic mask lengths, i.e., different samples use different mask lengths to allow the model better utilize all the information in the input sequences. Compared to BERT, RoBERTa requires longer pre-training time and larger computational resources, also leading to better performance on various tasks.

ALBERT is a lightweight PLM proposed by the Google Research team in 2019 [10], aiming to address BERT models' computational resource and time limitations. The main improvement of ALBERT is that the parameter sharing mechanism in BERT is changed from layer sharing to cross-layer sharing. In BERT, the parameters of each layer are trained independently, while in ALBERT, the parameters of all layers are shared, so the number of parameters of the model is significantly reduced, and the training and inference speed of the model is improved dramatically. In addition, ALBERT also adopts a new embedding layer regularization method, factorized embedding parameterization, which reduces the number of parameters of the model and improves the model's generalization ability by decomposing the parameter matrix of the embedding layer. The experimental results show that ALBERT has significantly reduced the number of parameters and computational resource consumption and improved performance on various NLP tasks compared with BERT.

Efficiently learning an encoder that classifies token replacements accurately (ELECTRA) is a novel PLM proposed by the Google Research team in 2020 [27]. Unlike other PLMs, ELECTRA employs a new pre-training task, replacing token detection. In ELECTRA's pre-training process, the model needs to replace some tokens in the original input text with other tokens. Then a discriminator is used to determine whether the tokens are replaced. This discriminator is a binary model to determine whether the original token has been replaced with a mask token. The main idea is to improve the generalization ability of the model by replacing tokens and training the discriminator to allow the model to understand the relationships and semantics between tokens better. In addition, compared with other PLMs such as BERT, ELECTRA can use computational resources more efficiently in the pre-training phase, improving the speed and efficiency of pre-training.

Masked language modeling as correction bidirectional representations from transformers (MacBERT) is a pre-trained model for Chinese NLP tasks obtained by improving and optimizing based on BERT [28]. Compared with BERT, MacBERT has been improved in the following aspects. Firstly, MacBERT expands the word list of BERT by adding some specific Chinese words and entities, such as names of people, places, and organizations, to improve the model's understanding of the Chinese language. Secondly, MacBERT uses a large number of external data sources to enhance the training data of the model, including web encyclopedias, question-and-answer communities, news, and forums, to improve the model's generalization ability. In addition, MacBERT learns multiple tasks simultaneously, such as text classification, question and answer, and NER, to improve the

model's adaptability and performance. Finally, MacBERT adopts a new attention mechanism called dot product and difference attention to better capture the relationship and importance between input sequences, thus improving the model's accuracy.

3 Problem Definition

For better research, we define the NER task for Chinese CTIs in this section.

The Chinese CTI NER task aims to recognize important CTI information, such as attack organizations and tools, and then classify them correctly, which can be viewed as a sequence annotation problem. Specifically, given a piece of cyber threat text $X = \langle x_1, x_2, \ldots, x_{n-1}, x_n \rangle$, for each word in the sentence, i.e., X_i, our task is to use the BIOES method [29] to label the location range and type of the entity vocabulary and later enable the model to learn and predict using the labels. Specifically, BIOES stands for beginning, inside, end, outside, and single, respectively. In addition, we add entity type information after the location label. For example, in this study, 'B' in the label 'B-Tools' is its location information and represents the beginning. At the same time, 'Tools' is the category information, so 'B-Tools' denotes that the word is the starting word of the named entity whose category is Tools. Specifically, the following Table 1 illustrates the labels of the sample "利用CMD执行命令".

Table 1: Examples of labeling methods

Sequence (English)	Execute commands using CMD								
Sequence (Chinese)	利	用	C	M	D	执	行	命	令
Sequence (Tag)	O	O	B-Tools	I-tools	E-Tools	O	O	O	O

4 Methods

As shown in Fig. 1, our model has two modules: pre-processing and the proposed NER model. Specifically, this section details the pre-processing methods in Section 4.1, including text truncation and data augmentation. After that, the RoBERTa-wwm-RDCNN-CRF model is elaborated in Section 4.2.

4.1 Data Pre-Processing

In the text pre-processing stage, we mainly performed text truncation to solve the limitation of the input sentence length due to the PLM. We also proposed an effective text augmentation method to improve the problem of the lack of labeled samples in the Chinese CTI NER domain. We are the first to publish the methods in detail in the area, hoping to provide a reference for other studies. We will introduce these two methods in detail in the following.

4.1.1 Two-Stage Multi-Level Text Truncation Method

As open source intelligence crawled from the web inevitably suffers from the problem of excessively long utterances. However, most PLMs based on BERT have restrictions on the input length [30]. If the length of the text to be classified exceeds 512 characters, the excess can only be discarded, and full utilization cannot be achieved, resulting in the loss of textual information. Therefore, we need to perform text truncation on the crawled raw data to enable a better distributed embedding representation.

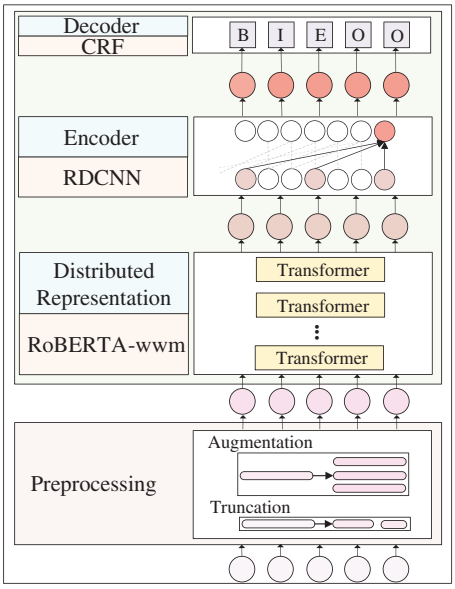

Figure 1: Flowchart of the proposed framework

The main task of text truncation is to split longer texts so they do not exceed the specified limits. However, direct segmenting sentences based on a specified length will likely cause a large amount of semantic information loss or even cut off the entity itself. For example, for the sentence "黑客计划攻击关键信息基础设施 (hackers plan to attack critical information infrastructure)", if the length limit is 10, it will be cut into "黑客计划攻击关键信息 (hackers plan to attack critical information)" and "基础设施 (infrastructure)", not only the meaning of the sentence becomes difficult to understand, but even the target entity "关键信息基础设施 (critical information infrastructure)" is cut off.

To address the above problem, a method based on sentence-level punctuation marks such as periods, question marks, and exclamation marks as segmentation characters can be used to split the text. However, due to the lack of standardization caused by the authors of CTI at the time of sharing and related practitioners at the time of crawling, common sentence-level ending symbols such as "." may not exist in the text, which may easily cause the problem of segmentation failure. To solve this problem, we propose a two-stage text truncation method based on the multi-level truncation and post-truncation verification to ensure the correctness of the truncated text length and content integrity. The specific method is shown in Fig. 2.

In this study, we generally have two text truncation phases, divided into a hierarchical truncation phase and a verification phase. There are three levels of truncators in the multi-level truncation phase. The first level of truncator is the sentence-level truncator, which is designed to handle more standardized statements by ".", "!", "?", and "....." to truncate the original text and obtain whole sentences. Then, to avoid the problem of long sentences caused by the publisher of threat intelligence using multiple ";", the clause-level truncator cuts the whole sentence into sub-clauses by ";". Because of the lack of standardization in CTIs, the symbols involved in the above process contain both English and Chinese versions.

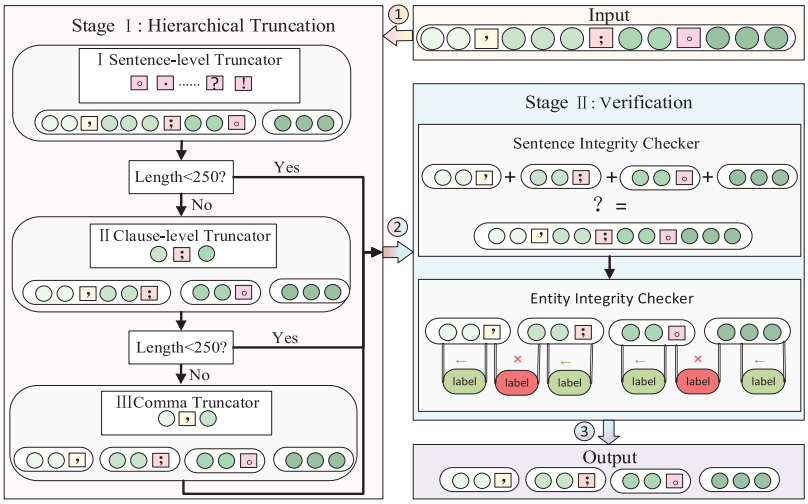

Figure 2: Schematic diagram of truncation

Finally, to avoid the problem that extreme non-standardization leads to texts that cannot be cut and sentences that do not conform to the specified length, we cut the text to within the fixed length using a comma truncator. It works as follows: firstly, find the position of the specified length. Then, find the first comma's position before the specific position, and truncate the text. The process is repeated until the clause is processed within the specified length. The reason for this approach, rather than direct truncation, is that direct truncation tends to hurt the meaning of the texts more and may even cut the entity apart. And by observing a large amount of CTIs, we have found that some texts do not have a period at all, resulting in the above whole sentence truncator cut invalid, while "," is more common in cybersecurity texts, so we think using a comma to cut better than a period.

To ensure the consistency of the text before and after the cut, we concatenate all the cut results in the validation stage to determine whether they are identical to the initial text. In addition, the position of entity labels is also shifted after the cut. While relabeling the position of entity labels, we also ensure that the tagged entities are not segmented by verifying that the new initial and ending positions of the labels are within the newly cut sentences. As shown in Fig. 2, the entity labeled in red is between the two cut sentences, which means that the requirement is unmet and the procedure must be reconsidered.

4.1.2 Text Augmentation

Although there are a lot of CTIs, the number of labeled texts is small due to the specialization of the cyber security field [4,31], especially in the Chinese domain [32,33], leading to inadequate model training and the inability to fit the data well. To solve this problem, we propose a simple yet effective data augmentation method to improve the performance of NER models in the CTI NER domain. In addition, because there are still many problems with open-source threat intelligence crawled in the network, such as duplicate fields, advertisements, and excessively long texts, it is difficult to summarize specific laws to circumvent these problems, so we avoid pre-processing the data in this regard, hoping that the model can build robustness with noise to better face the diverse forms of complex CTI texts.

The data augmentation method is as follows: the entity labels of the same category in the dataset are counted and then randomly replaced as the enhanced training text to train the model. For example, in Fig. 3, all the labels with the category "Region" are counted from the training set. Then the entities

with the entity type “Region”, such as “Maldives” are replaced with other entities of type “Region”, such as “Russia” or “Ukraine”, forming a new augmentation. Since the NER task only requires the recognition of entity boundaries and their types, the data augmentation uses the same entity substitution type without excessive changes in sentence semantics. In addition, although the PLM is effective, there is a problem in that the small number of fine-tuned samples makes it difficult to correct the inherent “bias” brought by the pre-training for certain entity types. For example, the entity type “region” is defined as “ The region targeted by the attacker or the region the attacker belongs to in the threat intelligence description”. However, PLMs tend to predict all the entities related to the region, causing the problem of a high false positive rate. By augmenting the data with certain types of entities, the entity cognitive bias of PLMs due to unsupervised learning can be mitigated to some extent.

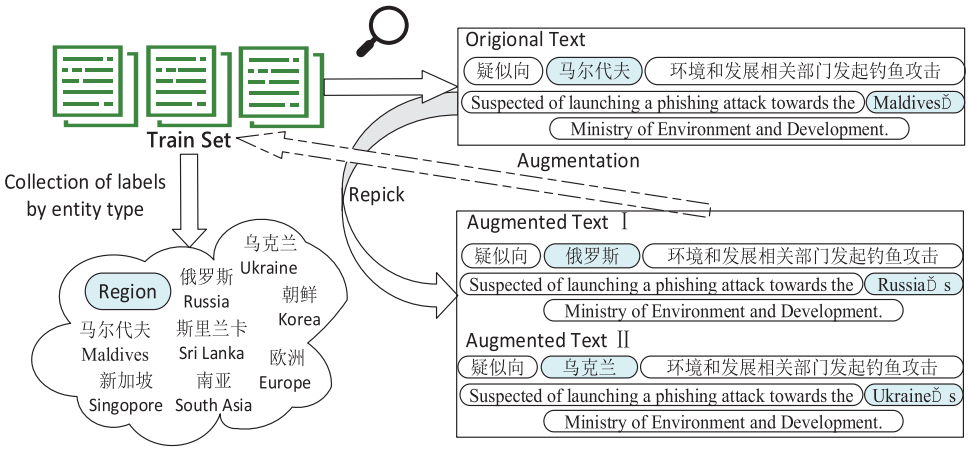

Figure 3: Example of data augmentation

4.2 RoBERTa-wwm-RDCNN-CRF Model

In this section, we first introduce the distributed representation model RoBERTa-wwm in Section 4.2.1. Then, we elaborate on the encoder RDCNN in Section 4.2.2. Finally, we explain the decoder CRF in Section 4.2.3.

4.2.1 RoBERTa-wwm

We select RoBERTa-wwm [24] for the pre-training model to obtain the semantic representation. For the Chinese version of the PLM, the official Chinese version BERT-base, released by Google, is sliced at character granularity, which does not consider the Chinese word separation, and can only mask characters instead of words. In this way, the resulting semantic representation is only at the character level when pre-training with Chinese corpus, which does not obtain word-level semantic representation, i.e., it is impossible to obtain word information during pre-training. The joint laboratory of Harbin Institute of Technology and iFLYTEK Research combines Chinese whole-word masking technology with RoBERTa model [26] to release a Chinese RoBERTa-wwm PLM [28].

Based on the Chinese Wikipedia corpus, in pre-training, it first uses the Harbin Institute of Technology Language Technology Platform to segment the texts in the corpus and then randomly masks a part of the words, i.e., mask all the Chinese characters that form the same word, and conducts the training. In this way, the semantic representation generated by this pre-training model contains

word information and is more suitable for Chinese NER tasks. Specifically, compared with BERT [3], RoBERTa-wwm improves the pre-training method in three aspects:

(1) In the masking scheme, RoBERTa-wwm uses a full-word mask instead of a single-character mask. As shown in Table 2, BERT randomly masks a certain percentage of individual characters in a sentence. In contrast, RoBERTa-wwm, with the full-word masking strategy, masks all characters belonging to the target word or phrase. The pre-training task of the wwm-based model helps capture semantic features at the Chinese word level, thus improving the model's overall performance.

Table 2: Example of a full word mask for RoBERTa-wwm

[Original text]
有一个网络攻击组织一直持续针对**评论人士**进行展开攻击。
A cyber attack group has been continuously targeting **commenters** to launch attacks.
[BERT masking method]
有一个网络攻击组织一直持续针对**[MASK][MASK]**人士进行展开攻击。
A cyber attack group has been continuously targeting **[MASK]** to launch attacks.
[RoBERTa-wwm masking method]
有一个网络攻击组织一直持续针对**[MASK][MASK][MASK][MASK]**进行展开攻击。
A cyber attack group has been continuously targeting **[MASK]** to launch attacks.

(2) In the pre-training task, dynamic masking is used instead of static masking to capture the context's semantic features. Static masking in BERT is a random selection of 15% tokens for each sequence and replaces them with [MASK], and the masked tokens do not change during the pre-training process. With the dynamic masking used in RoBERTa-wwm, the masked words are the tokens reselected in each iteration cycle.

(3) Removing NSP tasks in the pre-training phase. In BERT pre-training, positive samples are selected from the same document and negative samples from documents with different topics, so the neural network model can easily distinguish them. This training approach often does not match the new downstream tasks, and RoBERTa-wwm removes this task from its pre-training.

At the end of the pre-training phase, the RoBERTa-wwm language model can be plugged into the fully connected layer to adapt to downstream tasks in a fine-tuned manner.

4.2.2 RDCNN

As explained in Section 1, the choice of using RDCNN as the encoder in our study is motivated by its ability to provide a larger contextual scope, which is facilitated by its design of dilation. The principle underlying the RDCNN architecture can be described as follows:

(1) Dilated Convolutional Neural Network (CNN)

In the convolutional layer of the dilated CNN, the convolution operation can be represented by Eq. (1).

$$\boldsymbol{s_i} = \sum_{j=-h}^{h} f_j \boldsymbol{x_{i+j\times d}} + b \tag{1}$$

where i is the position of the word in the input sentence, $\boldsymbol{x}$ is the input sentence, $\boldsymbol{x_i}$ is the i_{th} word in the input sentence, $2h+1$ is the window size, f is the filter function, j is the relative position of offset i, f_j is the parameter value of the corresponding filter function at a distance j from character i, and $d > 1$ is the expansion coefficient of the dilated CNN. When $d = 1$, the dilated CNN at this time is equivalent to the conventional CNN.

As shown in Eq. (2), c_i is the information obtained after further functional deformation of the information extracted from the convolution, where $\boldsymbol{W}$ is the convolution kernel weight and b is the bias.

$$c_i = \text{relu}\,(\boldsymbol{W}\boldsymbol{s_i} + b) \tag{2}$$

The convolutional feature map is obtained by Eq. (1).

$$C = (c_1, c_2, \ldots, c_i, \ldots, c_n) \tag{3}$$

As shown in Fig. 4, the schematic diagram of an expanded convolutional block with an expansion factor of 3, a kernel size of 3, and 3 dilated convolutional layers in one block.

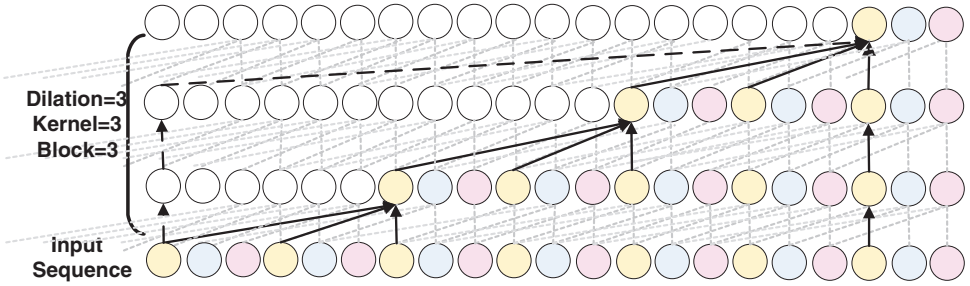

Figure 4: Schematic diagram of the dilated convolution block

(2) Residual Connection

Simple stacking of the null convolution can cause degradation problems in the network during training. As shown in Eq. (4), the residual connection [34] contains a branch that leads to the residual function $F(\boldsymbol{x})$ for transformation, whose output is added to the input $\boldsymbol{x}$ of the block:

$$\boldsymbol{o} = \boldsymbol{x} + F\,(\boldsymbol{x}) \tag{4}$$

The original information can be directly propagated to higher levels through residual connections. It has been shown repeatedly that deep networks would benefit from introducing residual connectivity [34,35]. As shown in Fig. 5, subfigure (a) shows the convolutional layer without residual connectivity, and subfigure (b) shows the residual block we propose to use. Specifically, within the residual block are an inflated convolutional block, a normalization layer, and a nonlinear layer. We apply LayerNorm [32] to perform normalization to achieve fast operations and model stability and use a rectified linear unit (ReLU) [36].

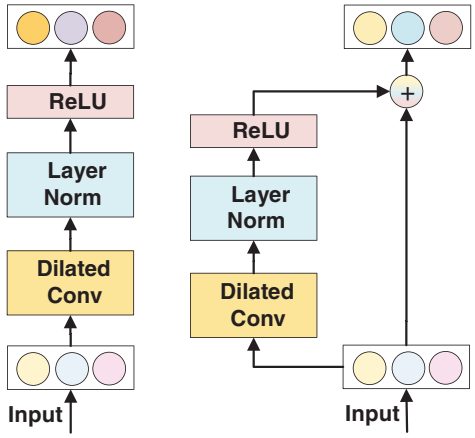

Figure 5: Diagram of different connection methods

4.2.3 CRF

We use CRF as the tag decoding layer to compute the most likely named entity classes. The CRF ensures valid final prediction results by learning some constraint rules to reduce the occurrence of illegal sequences with the following constraints:

(1) Entities need to start with "B-". The first character of a sentence can only begin with "B-" or "O", and entities or sentence capitals cannot start with " I-".

(2) Consecutive tags such as those headed by "B-Region" can only be followed by "I-Region" tags or "O" tags.

The loss function of the CRF layer is composed of two types of scores, a state score ($S_{emission}$) and a transfer score ($S_{transmition}$). The state score is the state matrix P_K, obtained from the output of the RDCNN, representing the probability of each word in the sequence corresponding to any one label, respectively. It can also be defined formally here as $X_{wi,yj}$, which represents the probability that the word w_i corresponds to each of the labels y_i. The transfer score is a parameter initialized by the CRF layer and represents the probability that the next label is y_j when the current word in the sequence corresponds to label y_i. It is called the inter-label transfer matrix, formally defined as $t_{yi,yj}$, and it learns the above constraint along the training.

If the sequence length is N and the total number of labels is M, there are M^N combinations of all labeled paths, and the true label sequence is the path with the highest probability. Then when calculating the loss function, we need to maximize Eq. (5).

$$-loss = \frac{P_{realpath}}{P_1 + P_2 + \ldots + P_{M^N}} \tag{5}$$

We define $P_i = e^{S_i}$ where S_i, as shown in Eq. (6), is the sum of the transfer and state scores on a path.

$$S_i = S_{emission} + S_{transition} = X_{w_0,\hat{y}_0} + X_{w1,\hat{y}1} + \ldots + X_{w_N,\hat{y}_N} + t_{\hat{y}_0,\hat{y}_1} + t_{\hat{y}_1,\hat{y}_2} + \ldots + t_{\hat{y}_{N-1},\hat{y}_N} \tag{6}$$

For computational purposes, -*loss* will be converted into a minimized logarithm with base e. As shown in Eq. (7), the expression is

$$
\begin{aligned}
loss &= -\ln\left(\frac{P_{realpath}}{P_1 + P_2 + \ldots + P_{M^N}}\right) = \\
&-\ln\left(\frac{e^{S_{realpath}}}{e^{S_1} + e^{S_2} + \ldots + e^{S_{M^N}}}\right) = \\
&-\left[\ln e^{S_{realpath}} - \ln\left(e^{S_1} + e^{S_2} + \ldots + e^{S_{M^N}}\right)\right] = \\
&-\left[S_{realpath} - \ln\left(e^{S_1} + e^{S_2} + \ldots + e^{S_{M^N}}\right)\right]
\end{aligned}
\tag{7}
$$

Once $S_{realpath}$ and S_i are found, the optimal solution can be obtained quickly using the Viterbi algorithm.

5 Results and Discussion

In this section, we focus on the public dataset [37] on the effectiveness of each component of the RoBERTa-RDCNN-CRF model, the effectiveness of different PLMs in the field of Chinese CTI, and the effectiveness of data augmentation methods are explored. Specifically, in this section, we will cover the following:

- To provide a detailed introduction to the NER dataset for Chinese CTI [37].
- To give a detailed description of the experimental setup, including comparative models, parameter setting, experimental environments, and evaluation metrics.
- To compare with the SOTA models: validation of the effectiveness of RoBERTa combined with RDCNN.
- To compare the effects of different PLMs in Chinese CTI NER.
- To verify the validity of different parts of the RDCNN module.
- To investigate the stability of RDCNN parameter changes.
- To validate the data augmentation method.

5.1 Chinese CTI NER Dataset

We use the latest and the first publicly available dataset[1] [37] to conduct our experiments. The dataset, published by Zhou et al. in 2023, contains about 200 open-source Chinese threat intelligence on 13 cybersecurity companies and annotated data on named entities and relationships in the Chinese cybersecurity domain. To the best of our knowledge, we will research it for the first time as a newly released dataset.

In this study, we only use the NER dataset. The dataset involves 3744 threat sentences and 205921 words, and five entity types are defined: Attacker, Tools, Industry, Region, and Campaign. The specific meanings can be seen in Table 3 below.

This dataset's training, validation, and test sets are distributed in the ratio of 8:1:1. As shown in Fig. 6 below, the distribution of each entity type in the different datasets is counted. It is easy to see that the Attacker type has the largest share in the training set with 32%, but the percentage in the test set is reduced to 20%. Meanwhile, the share of Region type increases from 21% to 34%. The inconsistency between the training set and the test set further tests the robustness of the model.

[1] https://github.com/MuYu-z/CDTier

Table 3: Entity type definitions

Entity name	Definition
Attacker	The person, group, or organization that carried out a specific malicious attack in a threat intelligence description.
Tools	Malware, legitimate software, or self-developed attack tools targeting a specific domain used by attackers in the threat intelligence description.
Industry	The target industry of the attacker in the campaign.
Region	The region targeted by the attacker or the region the attacker belongs to in the threat intelligence description.
Campaign	Indicates a specific attack activity initiated by an attacker.

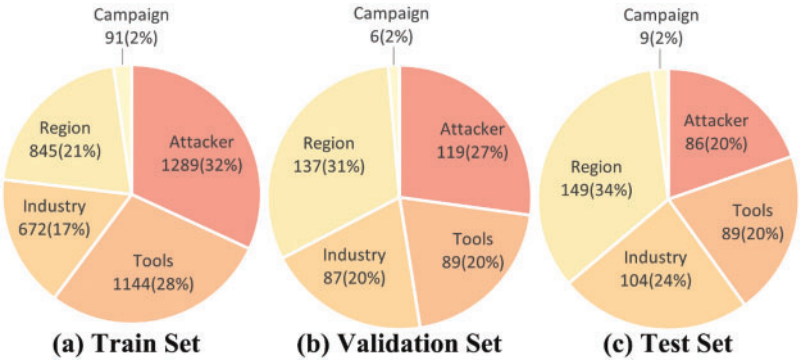

Figure 6: Pie chart of the distribution of entities in the dataset

In addition, the distribution of the lengths of the entities in the different datasets is shown in Fig. 7. In particular, the "pink star" icon represents the average length of each type of entity. It is not hard to see that the average length of Attacker and Tools is large. From the box and scatter plots in Fig. 7, it is easy to see that there are more outliers for the Industry, Attacker, and Tools types of entities.

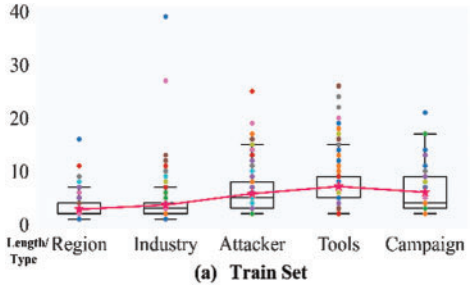

Figure 7: (Continued)

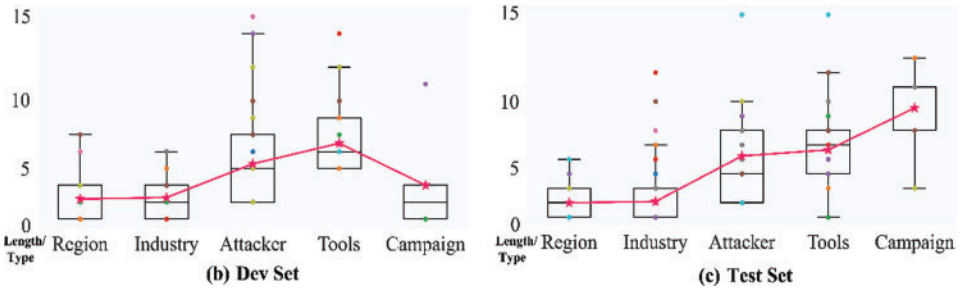

Figure 7: Entity length distribution statistics

5.2 Experimental Setup

5.2.1 Evaluation Metrics

We use precision, recall, and F1-score, which are more used in NER, as evaluation metrics. When there are N classes of entities, for the i_{th} class ($i = 1, 2, \ldots, N$), we have

$$P_i = \frac{TP_{\text{i}}}{TP_{\text{i}} + FN_i} \times 100\% \tag{8}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \times 100\% \tag{9}$$

$$F_i = \frac{2P_iR_i}{P_i + R_i} \times 100\% \tag{10}$$

where TP_i is the number of correctly identified entities of type i; FP_i is the number of incorrectly labeled entities of type i; FN_i is the number of missed identifications; P_i is the precision of identifying entities of type i; R_i is the recall of type i; F_i is the comprehensive value to evaluate the identification effect. For all entity types, the arithmetic average cannot reflect the actual recognition effect, so the weighted average of each type of index is used to evaluate the overall recognition effect of the model, and the expression is shown as follows:

$$P_T = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i} \times 100\% \tag{11}$$

$$R_T = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FN_i} \tag{12}$$

$$F_T = \frac{2P_TR_T}{P_T + R_T} \times 100\% \tag{13}$$

5.2.2 Experimental Parameter Settings

The experimental parameter settings may impact the performance of the deep learning models. The model parameters we used are shown in the following Table 4. Among them, the size of

feature embedding stands for the embedding size of BiLSTM and gated recurrent units (GRU). We experimented with embedding sizes of 64, 128, 256, and 384 for BiLSTM-based and BiLSTM + GRU-based models with both BERT and RoBERTa-wwm, finding 128 was the best hyperparameter. Max sequence length represents the number of Chinese characters we preserved in the pre-processing period. We conducted experiments with parameters 100, 150, 200, and 250, finding 250 was the best. However, limited by GPU capacity, we were unable to test longer sentences. The parameters of the residual inflation CNN (the number of residual blocks, the number of filters per residual block, kernel size of dilated convolution, and the dilation factor) are determined by the experimental results in Section 5.6. For the limitation of GPU capacity, we only conducted experiments with batch sizes of 16, 32, and 64, finding 64 was the best. After careful consideration and experimentation, we chose 0.00001, 0.01, and 0.0001 for the learning rate of PLM models, CRF, and the rest models, such as RDCNN, BiLSTM, and GRU. As observed, the models converged within 30 epochs. To ensure convergence, for each of the following models and parameter settings, we perform 30 training epochs and keep the model with the highest F1-score in the validation set as the final set.

Table 4: Model parameter settings

Parameters	Values
Size of feature embedding	128
Max sequence length	250
Number of residual blocks	4
Number of filters per residual block	512
Kernel size of dilated convolution	3
Dilation factor of the residual block	3
Batch size in training	64
Train epochs	30

To run the deep neural network models, we used the PyTorch library, all running on Python 3.6.13, GeForce RTX 3090 GPU, 80 GB RAM, 14 cores Intel (R) Xeon (R) Gold 6330 CPU @ 2.00 GHz Linux Server.

5.3 Comparison with SOTA Models

We choose BERT + BiLSTM + CRF (BBC) and BERT + BiLSTM + GRU + CRF (BBGC), which perform better in the NER task, as the baseline models. In addition, we chose the BERT + RDCNN + CRF (BRC) model that currently has the best performance in Chinese CTI NER [9], which is compared with the proposed RoNERTa + RDCNN + CRF (RRC) model, and the experimental results are shown in Table 5.

As seen from Table 5 above, the RRC model improves the F1-score by 19.52% and 19.31%, respectively, compared with the baseline models BBC and BBGC, confirming the effectiveness of our proposed model. More specifically, when the distributed representation PLM of the baseline model is replaced from BERT to RoBERTa, the improvement on the above two models is 4.90% and 1.89%, respectively. It indicates that the RoBERTa model is more effective than the BERT model when combined with the recurrent neural network (RNN)-based BiLSTM and GRU models, which

is more effective and better able to capture the semantic information of the vocabulary in the domain of cybersecurity with RNN-based encoders.

Table 5: Model performances comparison

Model	Precision-micro	Recall-micro	F1-micro
BBC	0.5984	0.6614	0.6283
BBGC	0.6118	0.6502	0.6304
RoBERTa + BiLSTM + CRF (RBC)	0.6093	0.7623	0.6773
RoBERTa + BiLSTM + GRU + CRF (RBGC)	0.6295	0.6704	0.6493
RRC	0.8311	0.8161	0.8235

RRC improves by 14.62% and 17.42% over RBC and RBGC, respectively, indicating that the CNN-based RDCNN model performs better than the RNN-based model. As shown in Table 6, the mean difference is the average of the differences between the F1-micro scores of RBGC and RRC and the F1-micro score difference between RBC and RRC. Among them, the difference between the Campaign category is 0.6079, which is the largest, mainly because the F1-micro score of RBGC on it is 0, i.e., it does not learn any effective semantic information related to the Campaign type entity; the scores of the Tools, Industry, and Attacker categories follow closely. The Region withholds the smallest difference, which is 0.0815. After observing on Fig. 7a, we find that the mean difference ranking is consistent with the statistical value of the distribution of this dataset, i.e., it has a positive correlation with its average length. Among them, the average length of Industry is lower than that of Attacker, but its length distribution is more dispersed, as shown by the scatter plots in Fig. 7. In summary, we speculate that the better performance of RDCNN is because it can better capture the association between remote semantic information and has a greater tolerance of entity length variance.

Table 6: F1-micro scores for each entity type of the model based on BiLSTM and RDCNN

	RBGC	RBC	RRC	Mean difference
Campaign	0.0000	0.6667	0.9412	0.6079
Tools	0.5366	0.5781	0.8177	0.2604
Industry	0.5463	0.5665	0.7282	0.1718
Attacker	0.6627	0.7283	0.8199	0.1244
Region	0.8063	0.8086	0.8889	0.0815

5.4 Comparison of the Effects of Different PLMs

Using RDCNN as the encoder and CRF as the decoder, we compare it with the BERT model [25] and many other PLMs to verify the effectiveness of the RoBERTa-wwm model. As shown in Table 7, the ALBERT model [10] has the worst effect, with only 76.46%. We believe that it is caused by the small size of its network layer, which improves the speed but reduces the accuracy. Moreover, in addition to the RoBERTa model, the LERT [38] model performs the best. However, its performance is worse than our proposed RoBERTa [26] and still has a difference of 0.96% in F1-micro scores.

Table 7: Performances comparison of different PLMs

Algorithm	Precision-micro	Recall-micro	F1-micro
AlBERT + RDCNN + CRF	0.7835	0.7466	0.7646
ELECTRA + RDCNN + CRF	0.8044	0.7377	0.7696
BERT + RDCNN + CRF	0.7681	0.8094	0.7882
MacBERT + RDCNN + CRF	0.8053	0.8161	0.8107
LERT + RDCNN + CRF	0.8202	0.8184	0.8193
RoBERTa + RDCNN + CRF	0.8311	0.8161	0.8235

As the best model BERT + RDCNN + CRF in the field of Chinese CTI NER at present [9], our proposed method improves by 6.30% and 0.67% compared to its precision and recall, respectively, for a total of 3.53% improvement in F1-micro score. Their specific enhancement information is shown in Table 8 below, where the Attacker category has the most enhancement of 5.34%. Industry, Tools, and Region have 3.59%, 2.39%, and 2.71%, respectively. The campaign category is 94.12%, with no enhancement compared to BERT. We believe that for most categories, the improvement brought by RoBERTa-wwm stems from its pre-training on larger-scale data compared to BERT models, including Wikipedia, news corpus, and Internet texts, which can help RoBERTa-wwm learn a wider range of linguistic knowledge and patterns to better understand and process texts in the Chinese cybersecurity domain. In addition, RoBERTa-wwm employs a series of pre-training tasks and techniques, such as MLM and continuous text classification. These tasks help RoBERTa learn to predict missing words from context and better understand the semantics and structure of the language, which is crucial for NER since contextual information about entity names is important for accurate entity recognition.

Table 8: F1-micro scores for each entity type based on BERT and RoBERTa

	BRC	RRC	Difference
Attacker	0.7665	0.8199	0.0534
Industry	0.6923	0.7282	0.0359
Tools	0.7938	0.8177	0.0239
Region	0.8618	0.8889	0.0271
Campaign	0.9412	0.9412	0.0000

As shown in Table 8, it can be seen that the number of Campaign category entities is relatively small, and the F1-micro of BERT itself has reached 94.12%, so it is difficult for RoBERTa to discover more semantic information. As a result, both of their precision is 100%, and recall is 88.89%. Their false negative error cases are "TrickyMouse". We assume the model fails to understand the latent relationship between "行动代号(action code)" and "Campaign" within the sentence. In addition, it should be noted that the Industry, Attacker, and Tools scores are relatively low in both models. We presume it is because the average length of their entities, shown in Fig. 6 in Section 5.1, is large, and the number of outliers makes it difficult for the model to discover the rules.

5.5 Effectiveness of the RDCNN Module

As the encoder in the RoBERTa-RDCNN-CRF model, RDCNN is of great importance. The effectiveness of the dilated convolution and residual connectivity in RDCNN is discussed in Sections 5.5.1 and 5.5.2, respectively.

5.5.1 Validity of the Dilated Convolution

As described in Section 4.2.2, the dilated CNN is the original CNN [39] that increases the perceptual field by no longer restricting the continuity of the filter. This section aims to verify the effectiveness of the dilated convolution on the Chinese CTI NER and the enhancement it brings by comparing the effects of dilated convolution with normal convolutional as model encoders. We conducted experiments with different block numbers and dilation factors under the filter number of 512 and the kernel size of 3, each with three dilated convolutional layers within each block. As shown in Table 9 below, the dilation factor represents the dilation factor of each convolutional layer in one block. For example, a dilation factor of 2, 2, 2 represents the dilation factor of 2 for each of its three layers. Specifically, when the dilation factor is 1, the dilation convolution degenerates to normal convolution. We put the highest score in bold, add an underline under the second-highest score, and italicize the third-highest score.

Table 9: F1-micro scores for different block numbers and dilation factors

Block number	Dilation factor			
	1, 1, 1	2, 2, 2	3, 3, 3	4, 4, 4
1	0.7654	0.8027	0.7987	0.7821
2	0.7871	0.7938	0.8080	0.7955
3	0.7938	0.7775	0.7608	0.7811
4	*0.8073*	0.8000	**0.8235**	0.7908

As shown in Table 9 above, when the dilated convolution factors are 1, 1, 1, the results are generally worse than after adding the expansion convolution, except when the block number is 3. Overall, it shows the effectiveness of introducing dilated convolution in the present NER problem.

5.5.2 Validity of the Residuals

This section aims to verify the effectiveness of the residuals by comparing the effect of NER after adding residual connections to the dilated convolution.

As shown in Table 10 below, either with BERT or RoBERTa as the PLM, using RDCNN as the encoder is better than the iterated dilated CNN (IDCNN), i.e., as shown in Fig. 5a, no residuals are added, improving 7.98% and 7.11% on BERT and RoBERTa, respectively, which proves the effectiveness of adding residuals.

Table 10: Residual validation table

Algorithm	Precision-micro	Recall-micro	F1-micro
BERT + IDCNN + CRF	0.7060	0.7108	0.7084
BERT + RDCNN + CRF	0.7681	0.8094	0.7882
RoBERTa + IDCNN + CRF	0.7097	0.8005	0.7524
RoBERTa + RDCNN + CRF	0.8311	0.8161	0.8235

5.6 Impact of RDCNN Parameter Changes on Chinese CTI NER

In this section, we will experimentally investigate the effect of the variation of important parameters of RDCNN on its recognition effect. These four important parameters are the number of filters f_d, the size of convolutional kernel k_d, the dilation factor d, and the number of dilated convolutional blocks b_d.

5.6.1 Impact of Different Filters and Kernel Size

By setting the block number to 4 and the dilation factor to 3, 3, 3, we investigate the effect of different number of filters and sizes of kernel on NER by conducting experiments with kernel sizes of 3, 5, 7, 9, and 11 and filter numbers of 128, 256, 384 and 512, respectively. The F1-micro scores of the model are shown in Table 11. The highest score is shown in bold, the second-highest score is underlined, and the third-highest score is italicized. The parameters here are the same as those in Table 4 except for the filter number and kernel size.

Table 11: F1-micro scores for different number of filters and kernel sizes

Filter number	Kernel size				
	3	5	7	9	11
128	0.6469	0.7825	0.6054	0.7610	0.7841
256	0.7836	0.8093	0.8149	0.7942	0.7969
384	0.7912	0.7872	*0.8145*	0.7996	0.8140
512	**0.8235**	0.8144	0.8159	0.7912	0.8084

Generally, when the number of filters is small, the recognition effect gradually increases as the kernel size becomes larger. But the recognition effect decreases when the kernel size increases to a certain degree. When the number of filters is large, such as when the filter number is 512, the recognition effect gradually decreases with the increase of kernel size. When the kernel size is small, the recognition effect gradually increases as the number of filters increases; and when the kernel size is large, such as when the kernel size is 11, the recognition effect first increases and then decreases as the number of filters increases. We believe this is due to overfitting as the number of filters and kernel size increase. After experiments, we found that the recognition effect of this model is best when the kernel size is 3, and the number of filters is 512.

5.6.2 Effect of Different Block Number and Dilation Factor

As shown in Table 9 above, when the block number is 4, and the dilation factor is 4, 4, 4, the recognition effect is the best, with an 82.35% F1-score. In addition, when the dilation factor is kept constant, the recognition effect gradually improves as the block number increases, which illustrates the effectiveness of increasing its perceptual field by increasing the block in this problem. However, the effect is slightly worse when the block number is 3. When the block number is kept constant, it is observed that the better effect generally occurs when the dilation is 3, 3, 3. Either too large or too small dilation is detrimental to the model's performance.

5.6.3 Computational Efficiency of the Present Investigation

In our analysis of RBC, BBC, RRC, and BRC, we found that RDCNN-based models required an average of 364 sec/epoch, while BiLSTM-based models consumed only 53 sec/epoch. This discrepancy demonstrates that while RDCNN-based models offer superior performance, they come at the cost of increased computational time. The challenge lies in improving the computational efficiency of RDCNN models.

In contrast, the computational efficiency of PLMs for BiLSTM-based models becomes a crucial consideration, given their lower time consumption. For example, the significant 16 sec/epoch difference between BRC (45 sec/epoch) and RBC (61 sec/epoch) accounts for almost one-third of BRC's total time consumption. Conversely, the time cost becomes less significant in the case of RDCNN-based models, as evidenced by BRC (376 sec/epoch) and RRC (353 sec/epoch). Therefore, when considering time consumption, it becomes important for BiLSTM-based models to choose a faster pre-trained language model (PLM).

5.7 Effectiveness of Text Augmentation

As shown in Fig. 8a below, we validate the effectiveness of our proposed data augmentation method on several models. Experiments show that the data augmentation method is more effective on the RBGC model, with an improvement of 8.38% F1-score. It has a good improvement on the BiLSTM encoder-based models. Specifically, its effectiveness on the BBGC, BBC, and RBC improve by 4.92%, 2.88%, and 0.65%, respectively. However, this data augmentation method has a limited effect on the RDCNN encoder-based model, which only improves by 0.09% F1-score on the BRC and even reduces by 2.63% F1-score on RRC. As can be seen from Figs. 8b and 8c, the effectiveness of the above models mainly stems from the decrease in their recall values, i.e., the decrease in the ability to identify all the entities needed to be identified. However, at the same time, there is a significant improvement in the precision, i.e., the entities identified by the model are more accurate, reducing the false positive rate and ultimately leading to an increase in the F1-score of most models. This data augmentation method is more suitable for NER models based on BiLSTM as an encoder.

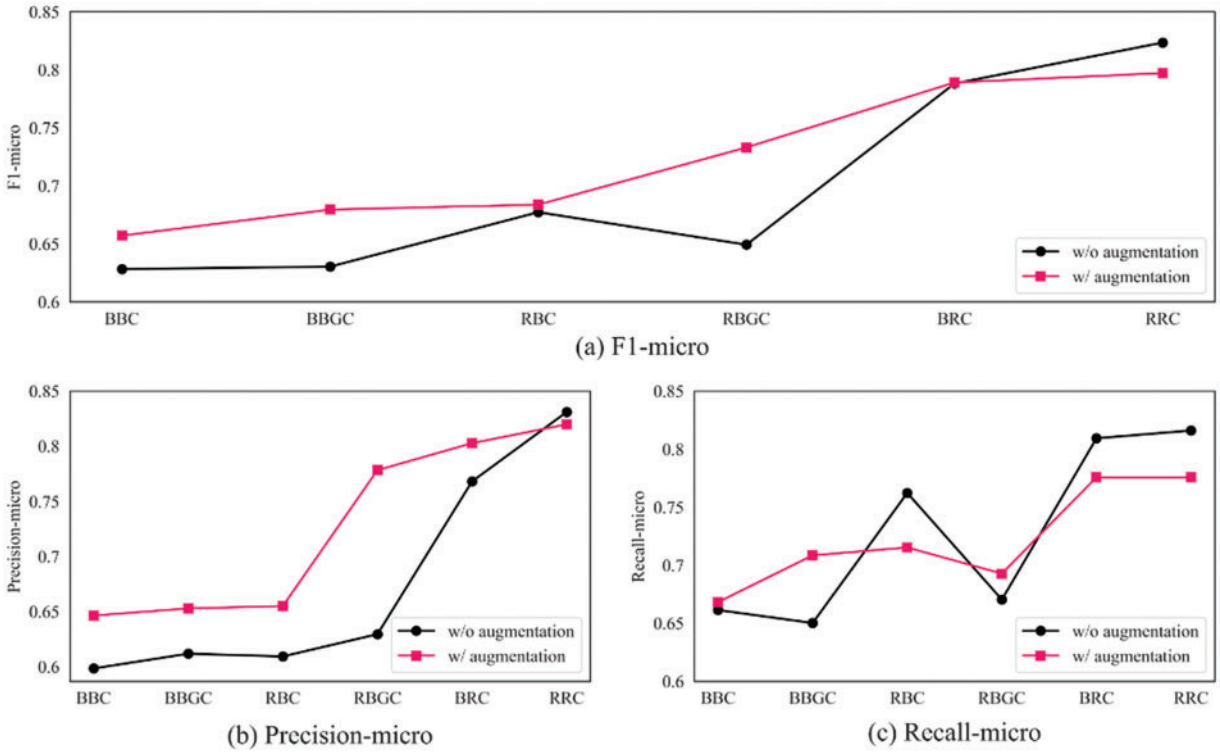

Figure 8: Comparison of data augmentation effects

6 Conclusion and Future Work

Due to the non-standardized and technical characteristics of Chinese CTI, its NER is a novel and challenging task. Firstly, in this study, we propose the RoBERTa-wwm-RDCNN-CRF model, which employs RoBERTa-wwm which is a powerful PLM, an efficient and accurate encoder RDCNN, and an effective decoder CRF. We conduct comprehensive experiments on a real-world dataset. Its F1-score reaches 82.35%, which surpasses the current best model BRC [9] by about 3.53% and exceeds the commonly used baseline model in this field, BBC, by approximately 19.52%. Secondly, we keep up with the times to explore the effectiveness of various contemporary pre-trained languages with better results in Chinese CTI NER and find that RoBERTa-wwm has the best results. Thirdly, in addition to demonstrating the effectiveness of dilated residual convolution, which has not been done by previous studies, we also explore the effects of the number of modules, dilation factors, the number of filters, and kernel size on their performance. Fourthly, we propose a data augmentation method, verified to have significant effect enhancement in models based on BiLSTM as an encoder. Finally, we publish our pre-processing method for other researchers to reference and reproduce. We hope that the above findings can provide a reference for other researchers.

In the future, we will continue to explore the encoder part of RoBERTa-wwm-RDCNN-CRF, hoping to improve the computational efficiency of RDCNN and the effectiveness of NER further. In addition, we will also embark on the study of relationship extraction in Chinese CTI and combine

it with this study to build a knowledge graph in the Chinese CTI domain, which can provide favorable support for downstream applications in cybersecurity, such as situational awareness, intrusion detection, and ATP detection.

Acknowledgement: We would like to express our sincere gratitude to the previous studies and researchers in the field of Chinese CTI for laying the foundation of our study.

Funding Statement: This research was funded by the Double Top-Class Innovation Research Project in Cyberspace Security Enforcement Technology of People's Public Security University of China (No. 2023SYL07).

Author Contributions: Study conception and design: Z.Z., J.G.; data collection: Z.Z.; analysis and interpretation of results: Z.Z., J.G.; draft manuscript preparation: Z.Z., J.G. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in [37] can be reached at https://github.com/MuYu-z/CDTier.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. D. Wagner, K. Mahbub, E. Palomar and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," *Computers & Security*, vol. 87, pp. 101589–101602, 2019.

[2] C. Gao, X. Zhang, M. Han and H. Liu, "A review on cyber security named entity recognition," *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 9, pp. 1153–1168, 2021.

[3] Z. Long, L. Z. Tan, S. P. Zhou, C. Y. He and X. Liu, "Collecting indicators of compromise from unstructured text of cybersecurity articles using neural-based sequence labelling," in *Int. Joint Conf. on Neural Networks*, Budapest, Hungary, pp. 1–8, 2019.

[4] Y. Qin, G. Shen, W. Zhao, Y. Chen, M. Yu *et al.,* "A network security entity recognition method based on feature template and CNN-BiLSTM-CRF," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 6, pp. 872–884, 2019.

[5] T. Li, Y. B. Guo and A. K. Ju, "Triple extraction of network security knowledge fused with adversarial active learning," *Journal on Communications*, vol. 41, no. 10, pp. 80–91, 2020.

[6] B. Xie, G. Shen, C. Guo and Y. Cui, "The named entity recognition of Chinese cybersecurity using an active learning strategy," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–11, 2021.

[7] Y. B. Guo, Y. F. Li, Q. L. Chen, C. Fang and Y. Y. Hu, "Cyber threat intelligence entity extraction based on focal loss," *Journal on Communications*, vol. 43, no. 7, pp. 85–92, 2022.

[8] X. Z. Yang, G. J. Peng, Z. C. Li, Y. Q. Lv, S. D. Liu *et al.,* "Research on APT attack entity recognition and alignment based on BERT and BiLSTM-CRF," *Journal on Communications*, vol. 43, no. 6, pp. 58–70, 2022.

[9] B. Xie, G. W. Shen, C. Guo, Y. Zhou and M. Yu, "Cybersecurity entity recognition method based on residual dilated convolution neural network," *Chinese Journal of Network and Information Security*, vol. 6, no. 5, pp. 126–138, 2020.

[10] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma *et al.,* "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. of ICLR*, Addis Ababa, Ethiopia, pp. 1–17, 2020.

[11] M. Anul Haq and M. Abdul Rahim Khan, "DNNBoT: Deep neural network-based botnet detection and classification," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1729–1750, 2022.

[12] M. Anul Haq, M. Abdul Rahim Khan and T. AL-Harbi, "Development of PCCNN-based network intrusion detection system for EDGE computing," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1769–1788, 2022.
[13] M. Anul Haq, "DBoTPM: A deep neural network-based botnet prediction model," *Electronics*, vol. 12, no. 5, pp. 1159–1173, 2023.
[14] E. Balamurugan, A. Mehbodniya, E. Kariri, K. Yadav, A. Kumar *et al.,* "Network optimization using defender system in cloud computing security based intrusion detection system with game theory deep neural network (IDSGT-DNN)," *Pattern Recognition Letters*, vol. 156, pp. 142–151, 2022.
[15] M. Anul Haq, M. Abdul Rahim Khan and M. Alshehri, "Insider threat detection based on NLP word embedding and machine learning," *Intelligent Automation & Soft Computing*, vol. 33, no. 1, pp. 619–635, 2022.
[16] G. Aguilar, F. AlGhamdi, V. Soto, M. Diab, J. Hirschberg *et al.,* "Named entity recognition on code-switched data: Overview of the CALCS, 2018 shared task," in *Proc. of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, Melbourne, Australia, pp. 138–147, 2018.
[17] P. Liu, H. Li, Z. Wang, J. Liu, Y. Ren *et al.,* "Multi-features based semantic augmentation networks for named entity recognition in threat intelligence," in *26th Int. Conf. on Pattern Recognition (ICPR)*, Montreal, Canada, pp. 1557–1563, 2022.
[18] M. T. Alam, D. Bhusal, Y. Park and N. Rastogi, "Looking beyond IoCs: Automatically extracting attack patterns from external CTI," arXiv: 2211.01753, 2022.
[19] M. Bayer, P. Kuehn, R. Shanehsaz and C. Reuter, "CySecBERT: A domain-adapted language model for the cybersecurity domain," arXiv: 2212.02974, 2022.
[20] M. T. Alam, D. Bhusal, Y. Park and N. Rastogi, "CyNER: A python library for cybersecurity named entity recognition," arXiv: 2204.05754, 2022.
[21] M. Alsaedi, F. Ghaleb, F. Saeed, J. Ahmad and M. Alasli, "Cyber threat intelligence-based malicious URL detection model using ensemble learning," *Sensors*, vol. 22, no. 9, pp. 1–19, 2022.
[22] Z. Li, J. Zeng, Y. Chen and Z. Liang, "AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports," in *European Symp. on Research in Computer Security*, Darmstadt, Germany, pp. 589–609, 2022.
[23] S. Dasgupta, A. Piplai, A. Kotal and A. Joshi, "A comparative study of deep learning based named entity recognition algorithms for cybersecurity," in *2020 IEEE Int. Conf. on Big Data (Big Data)*, Atlanta, USA, pp. 2596–2604, 2020.
[24] C. E. Tsai, C. L. Yang and C. K. Chen, "CTI ANT: Hunting for Chinese threat intelligence," in *2020 IEEE Int. Conf. on Big Data (Big Data)*, Atlanta, USA, pp. 1847–1852, 2020.
[25] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv: 1810.04805, 2019.
[26] Y. H. Liu, M. Ott, N. Goyal, J. F. Du, M. Joshi *et al.,* "RoBERTa: A robustly optimized BERT pretraining approach," arXiv: 1907.11692, 2019.
[27] K. Clark, M. T. Luong and Q. V. Le, "ELECTRA: Pre-training text encoders as discriminators rather than generators," arXiv: 2003.10555, 2020.
[28] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang *et al.,* "Revisiting pre-trained models for Chinese natural language processing," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Punta Cana, Dominican Republic, pp. 657–668, 2020.
[29] E. F. T. K. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proc. of the Seventh Conf. on Natural Language Learning at HLT-NAACL 2003 (CoNLL-2003)*, Edmonton, Canada, pp. 142–147, 2003.
[30] M. Ding, C. Zhou, H. Yang and J. Tang, "CogLTX: Applying BERT to long texts," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, Virtual, pp. 12792–12804, 2020.
[31] Q. X. Wang and W. Yang, "Research on threat intelligence entity extraction based on STIX standard," *Cyberspace Security*, vol. 11, no. 8, pp. 86–91, 2020.

[32] Y. Ren, Y. Xiao, Y. Zhou, Z. Zhang and Z. Tian, "CSKG4APT: A cybersecurity knowledge graph for advanced persistent threat organization attribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5695–5709, 2022.
[33] K. Zhang, X. Chen, Y. Jing, S. Wang and L. Tang, "Survey of research on named entity recognition in cyber threat intelligence," in *2022 IEEE 7th Int. Conf. on Smart Cloud (SmartCloud)*, Shanghai, China, pp. 68–73, 2022.
[34] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 770–778, 2016.
[35] C. Szegedy, S. Ioffe, V. Vanhoucke and A. Alemi, "Inception-v4, inception-resNet and the impact of residual connections on learning," in *Proc. of the AAAI Conf. on Artificial Intelligence*, California, USA, pp. 4278–4284, 2017.
[36] X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. of the Fourteenth Int. Conf. on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, pp. 315–323, 2011.
[37] Y. Zhou, Y. T. Ren, M. Yi, Y. J. Xiao, Z. Y. Tan *et al.,* "CDTier: A Chinese dataset of threat intelligence entity relationships," *IEEE Transactions on Sustainable Computing*, pp. 1–13, 2023. https://doi.org/10.1109/TSUSC.2023.3240411
[38] Y. Cui, W. Che, S. Wang and T. Liu, "LERT: A linguistically-motivated pre-trained language model," arXiv: 2211.05344, 2022.
[39] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard *et al.,* "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.