International Journal of Innovative
Computing, Information and Control
Volume **17**, Number **6**, December **2021**

 ISSN 1349-4198
pp. **1973–1988**

SELF-ATTENTION BASED DARKNET NAMED ENTITY RECOGNITION WITH BERT METHODS

Yuxuan Chen, Yubin Guo, Hong Jiang, Jianwei Ding and Zhouguo Chen

The 30th Research Institute of China Electronics Technology Group Corporation
No. 8, Chuangye Road, Chengdu 610093, P. R. China
Ceeall@outlook.com; ggyybb@foxmail.com; { 12134; mathe_007; czgexcel }@163.com

Received June 2021; revised September 2021

Abstract. *Darknet has become the focus and difficulty of cyberspace security research due to its anonymous services and a large number of illegal activities. In order to support cyberspace security, tracing the source of illegal activities and characters has become a new requirement for current cyberspace security. Darknet Named Entity Recognition (DarkNER) is the basic and critical task of source tracing to identify entities including person, organization, location and black words, etc. This paper introduces BERT language representation model as pretrained model with Iterated Dilated Convolutional Neural Networks (ID-CNNs) and Conditional Random Field (CRF) on the top to extract character and text features and predict sequence labels. Aiming at the text characteristics of Darknet, we build a self-attention layer on top of the ID-CNNs network to improve the effect of Darknet named entity recognition. In order to solve the problem of the low recognition accuracy of black words entities, we add 3 types of Darknet black words dictionaries features into the model to improve the recognition performance of black words. We not only compare several models based on BERT with models based on word2vec methods, but also compare the recognition effect of BERT-ID-CNNs-CRF model with the self-attention based BERT-ID-CNNs-CRF model. The results showed that our self-attention-based BERT model with dictionary features significantly outperforms the state-of-the-art methods.*

Keywords: Darknet, Self-attention, BERT, Named entity recognition, Deep learning, Iterated dilated convolutional neural networks, Conditional random field

1. **Introduction.** With the deepening of network information and data, criminals use network technology to hide personal identities, anonymously publish and disseminate illegal information. The ZeroNet network itself does not have anonymity, but it supports Tor service and can use Tor to provide service deployment and resource acquisition functions. Therefore, ZeroNet, as a new type of Darknet, also has problems such as difficulty in node discovery, difficulty in service location, difficulty in user monitoring, and difficulty in confirming communication relationships.

By constructing a knowledge graph that integrates the ZeroNet network layer and the information layer, not only can the information of ZeroNet sites and nodes and their relationships be retrieved intuitively and efficiently, but also a large amount of reliable information can be provided for the detection and root tracing of offenders and incidents. It can further realize the tracking and positioning of illegal characters and information, and Darknet Named Entity Recognition (DarkNER) is the most basic and critical step in constructing the Darknet knowledge graph. Darknet named entity recognition aims to automatically extract valuable information carriers in the text from Darknet sites, such as person names, locations, organizations, and black words.

DOI: 10.24507/ijicic.17.06.1973

As a classic task of Natural Language Processing (NLP), named entity recognition has wide usage in tasks such as automatic question answering, machine translation, information retrieval, and public opinion monitoring. Named entity recognition was originally based on rules and dictionaries. The rule-based and dictionary-based methods have to be written manually, using the rules written by experts to match in the text to extract the corresponding entities. In 1991, Rau [1] implicitly proposed the NER task. He used a combination of heuristic algorithms and rules to automatically extract company name entities from text. In the earliest days, the text entities extracted by named entity recognition were concentrated in three basic categories: person, organization, geographic location. At the same time, there are also studies on the recognition of currency, time, and percentage expression. The former belongs to the entity type and the latter belongs to the numeric type. Mariyah et al. [2] extracted important entities stated in SMS text, which help reveal the customer behavior and get the new insight of business.

Pattern matching methods based on dictionary and rules, also known as Export System (ES), have high accuracy in extracting texts with simple and regular language patterns, such as common person, organization, location [3], and time, date, price, phone number [4]. However, the pattern matching methods have poor scalability, since new entities with different patterns cannot fit these rule-based and dictionary-based methods.

Machine learning models first design feature templates to learn the contextual features of entities. Machine learning algorithms, such as Hidden Markov Models (HMM) [5], Support Vector Machines (SVM) [6], and Conditional Random Fields (CRF) [7] are widely studied and applied. These methods require a lot of engineering skills and domain knowledge to build feature templates which are critical to machine learning methods.

Compared with the machine learning methods, the deep learning methods discard the artificially defined feature template. The Neural Network Language Model (NNLM) [8] is used to train word vectors, with the purpose of obtaining reasonable sentences through unsupervised training methods. Word2vec [9] and glove [10] are the two most outstanding algorithms in word vector training algorithms, and they are widely used in named entity recognition models.

After the text word vector is represented, the neural network is used to extract vectorized text features. Several popular deep learning models are proposed for NER. Huang et al. [11] proposed a Bidirectional-Long Short-Term Memory (Bi-LSTM) model as the encoding layer. This method obtained 88.8% F1-scores on the CoNLL-2003 English shared task. Strubell et al. [12] proposed the use of ID-CNNs for named entity recognition, while improving the training speed. While word2vec and glove are called static word vectors and cannot solve the polysemy phenomenon of a word, Devlin et al. [13] proposed a language representation model called Bidirectional Encoder Representations from Transformers (BERT), which can learn the contextual features of the text, and infer the meaning of the words according to the context representation, and thus learn the polysemy of the words. Tran et al. [14] used BERT method and applied the focal loss instead of cross entropy to solving the imbalance between positive and negative samples, which improved the performance of multi-grained named entity recognition.

In this paper, we propose multiple novel Darknet Named Entity Recognition (DarkNER) models with BERT methods and use the self-attention mechanism to improve the semantic and grammatical clutter of Darknet text, which leads to the phenomenon of low named entity recognition. BERT pretrained the mixed representations on a large unlabeled dataset which contains 2.5G words. The token-level output of pretrained BERT model is passed to a linear layer or encoding layers to encode semantic information. Then, we construct a conditional random field layer to predict the current tags using the adjacent tag information. We compare these models with state-of-the-art DarkNER models

on the same dataset. The experimental results show that our models can achieve high accuracy in few steps, and our proposed models perform much better than state-of-the-art methods in all metrics.

The main contributions of this paper can be summarized as follows.

1) Instead of using word2vec and feature template, we introduce pretrained BERT model to obtain distributed representations of words, which, as a result, greatly improves the performance of the NER system of Darknet.

2) We propose 4 models for Darknet named entity recognition, including BERT-fine-tuning model, BERT-CRF model, BERT-ID-CNN-CRF model, BERT-ATT-IDCNNs-CRF model. What is more, aiming at the grammatical and semantic characteristics of Darknet texts, we build the self-attention mechanism to improve the effect of named entity recognition on Darknet texts.

3) We collect texts from the ZeroNet site and semi-automatically label the dataset. We evaluate the performance of different models on our Darknet dataset, and the results show that our model can surpass the performance of the state-of-the-art models in DarkNER.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the architectures of BERT, several encoding layers, CRF decoding layer and the self-attention mechanism. Darknet dataset and its statistics are described in Section 4. The experimental results are reported in Section 5. Section 6 draws conclusions.

2. **Related Work.**

Recently, there have been some researches on the Darknet and named entity recognition algorithm. This chapter will review these works in detail.

2.1. **Darknet analysis.**

Guo et al. [15] proposed a Darknet text crawling system based on simulated login, and used the semi-supervised LDA topic model to classify the topics of the ZeroNet Darknet, and found that there is a lot of content that endangers network and social security. Nunes et al. [16] proposed an operating system that collects cyber threat intelligence from Darknet and deep web sites, and constructed a co-training SVM machine learning algorithm to identify cyber threats, including newly developed malware and information on vulnerabilities that have not yet been deployed in cyber-attacks. Han et al. [17] proposed a traffic analysis method based on the graphical Gaussian model and the graphical lasso algorithm, which can monitor the malware traffic in the communication traffic in real time. The work of Han et al. [17] makes it possible to monitor malicious traffic in real time.

The above-mentioned studies analyzed the Darknet from the content level and the network level, and improved the ability to monitor the Darknet. These approaches were used to identify the nature and characteristics of Darknet's website and the users' activities on the Darknet. These methods, however, cannot help us understand the information in Darknet's websites. Darknet named entity recognition extracts the main information of Darknet's websites and provides some insights into whether users make illegal statements or conduct a series of criminal activities on Darknet's websites. It aims to identify the main information existing in the Darknet's website and determine whether the website has potential illegal incidents that pose a threat to individuals, organizations, and society.

2.2. **Named entity recognition for Darknet.**

Al-Nabki et al. [18] first proposed the concept of DarkNER and introduced the BiLSTM-CRF model to recognize the Tor domain entities. They did not construct a Darknet dataset, but adopted a public dataset on the Internet that is more similar to the Darknet data format. Al-Nabki et al. [18] also added a gazetteer to help recognize entities. Then, Al-Nabki et al. [19] presented a novel feature, called local distance neighbor, which substitutes gazetteers. They added new types of entities of Darknet, which is related to weapons and drug selling. Fan et al. [20] proposed

a Darknet market named entity recognition system based on the CNN-BiLSTM-CRF model, which can improve the recognition effect of entity types in special fields.

In terms of deep learning models, these DarkNER methods are all based on word vector embedding methods, but the word embedding method can only encode the same word into only one embedding, and thus it is impossible to distinguish the different semantics of polysemy. What is more, the ability of CNN and LSTM to extract features is weaker than Transformer that is used by BERT. Based on the above considerations, we use the BERT model to further improve the recognition effect of Darknet named entities. In terms of the data used in Darknet named entity recognition, previous studies used datasets in surface web which are similar in structure to Darknet texts, or Darknet trading market texts. However, there is also a large amount of important information in the posts and comments of Darknet websites, including illegal and criminal information. Therefore, we collect Darknet posts and comments' texts to construct a Darknet named entity recognition dataset, including the common entity types in the surface web and Darknet black words. Our research expands the scope of Darknet named entity recognition, studies the named entity recognition of black words from Darknet posts and comments, and designs an algorithm to improve the recognition effect of Darknet named entity recognition.

3. **Model Architecture.** In this section, we describe all architectures used in our models. We build a pre-trained BERT deep learning model, using dilated convolutional neural network ID-CNNs and self-attention-based ID-CNNs to encode the output of BERT, and the final output layer is decoded by the CRF layer. In this paper, three types of Darknet black words dictionaries are integrated into the model, which greatly improves the recognition effect of the named entity recognition model on Darknet sites' black words. The overall system architecture of the proposed models is shown in Figure 1 [13]. In the BERT network, the special classification token ([CLS]) is the first token of each sequence. We denote the embedding of the input token i as E_i, the final hidden vector of the input token i as $T_i \in \mathbb{R}^H$ and the final hidden vector of the special [CLS] token as $C \in \mathbb{R}^H$.

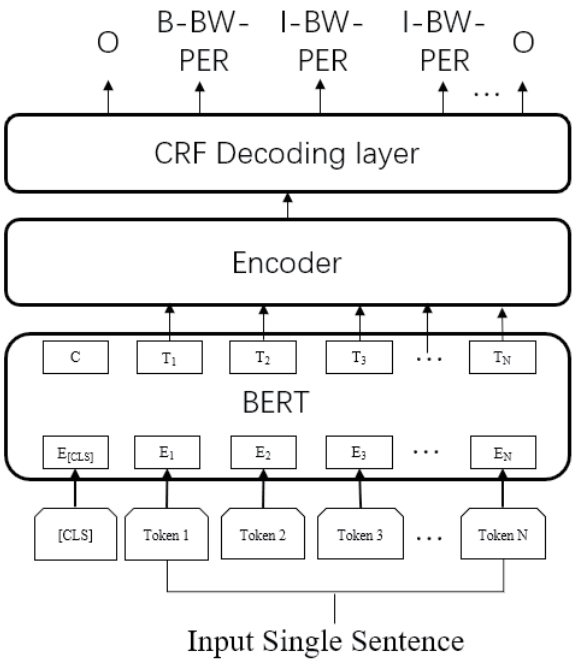


FIGURE 1. Model architecture

3.1. **BERT embedder.** BERT pretrained deep bidirectional representations from unlabeled text. Since the original vocabulary of BERT does not include some black words, we have added some important black words to the basic vocabulary. The BERT was pretrained using two unsupervised tasks, including masked language model and next sentence prediction.

The masked language model masked 15 percent of the input tokens as object words at random and predicted the masked words omnidirectionally based on the context. The task of the masked language model is to correctly predict these masked words. The model preliminarily trained the parameters of the Transformer model by predicting the masked vocabulary omnidirectionally.

The next sentence prediction is a binarized task which chooses sentences A and B from the corpus to predict whether B is the next sentence of A. Sentence B has a 50% probability of being the correct next sentence of sentence A, and a 50% probability of being a random other sentence. Then Transformer updates its own parameters by identifying the continuity of sentences A and B, judging which are continuous and which are not.

The data sent to the BERT embedding layers are preprocessed according to the paper of Devlin et al. [13]:

1) A [CLS] token is added at the beginning of the first sentence, and a [SEP] token is added between the two sentences and at the end of the last sentence.

2) Sentences are filled with [PAD] at the end until they reach the maximum sequence length.

3) Convert each word in the sentences into a one-dimensional vector by querying the word vector table with original 21,000 token vocabulary and additional 80 token security vocabulary as the model input.

For a given token, the input representation is obtained by adding three parts together:

$$I = \sum_{j=0}^{m-1} E_j \tag{1}$$

where

- I is the third dimension vector of input representation with shape (1, n, 768) that is passed to BERT's encoder layer;
- E_0 is word embedding (generated based on WordPiece);
- E_1 is segment embedding (indicating the sentence to which the token belongs);
- E_2 is position embedding (indicating the position information of the token in the sequence).

We get the sequence output of BERT model, summed by transformers:

$$out_{seq} = \alpha \sum_{i=0}^{A-1} \beta_i T_i \tag{2}$$

where

- α is a parameter vector of output layer, β_i is the parameter of the i-th transformers from $l-1$ layer;
- A is the number of self-attention heads which is set to 12;
- T_i is the output of the i-th transformers from $l-1$ layers.

3.2. **BERT baseline.** After getting the BERT sequence output, in order to get the baseline of BERT model on our own Darknet dataset, we simply add a fully connected linear layer and a log-SoftMax layer to calculate the log probabilities:

$$prob = softmax_{log}(W * out_{seq} + b) \tag{3}$$

where W is the parameter matrix of the fully connected linear layer and b is the bias parameter of the fully connected linear layer. Then, the predicted tag is the label with the highest probability on a single character.

3.3. **Self-attention.** Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence [21]. Self-attention uses the attention mechanism to calculate the association between each word and all other words, and words with a high degree of correlation with the current word have a higher attention score. For example, if we use the sentence, "China and Japan are geographically separated by the East China Sea", for named entity recognition task, when deciding the tag of the word "China", the word "Japan" will have a high attention score. Attention score is a weighted representation of the correlation strength of the current two words, and then a new representation of the word in the feedforward neural network is obtained by calculation. This means that by learning the dependencies through the self-attention mechanism, contextual information can be well considered. The encoder loads the input data, and uses the self-attention mechanism that is superimposed to obtain a new representation of each word that takes account of the contextual information. At the same time, the decoder in transformer also uses a similar self-attention mechanism. Compared with the encoder, it not only depends on the previous output, but also depends on the output of the encoder.

3.4. **Encoder.** The simple linear encoder cannot well-finetune tasks on specific domain like DarkNER. So, we apply 2 different models to encoding the sequence output from BERT, namely ID-CNNs, Att-IDCNNs.

3.4.1. *ID-CNNs layer.* The original CNN has a disadvantage that after convolution, the last layer of neurons only gets a small piece of information in the original input data. The dilated CNN adds a dilation width to this filter. When it acts on the input matrix, it skips all the input data in the middle of the dilation width, and the size of the filter matrix itself remains unchanged, so that the filter obtains the data on a wider input matrix. The ID-CNNs layer is shown in Figure 2 [12].

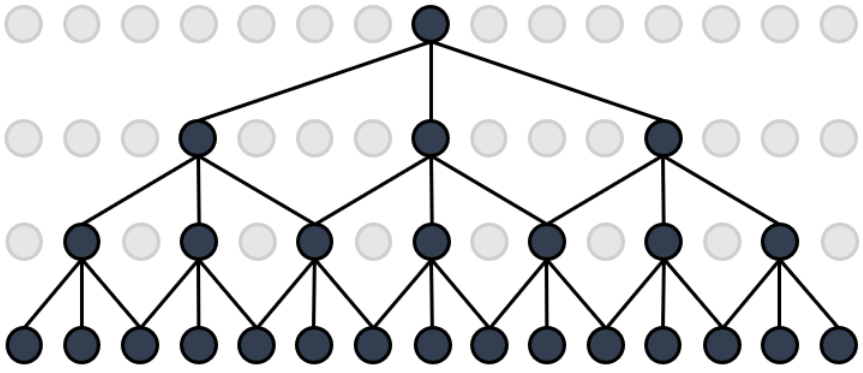


FIGURE 2. Layer of ID-CNNs

The ID-CNN layer shown in Figure 2 is divided into three dilated convolutional blocks, and the dilated widths from bottom to top are 1, 2, and 4. ID-CNN captures the long-distance information of a long sequence of texts when local information is lost, which is suitable for the current long text dataset. This method has better context and structured prediction capabilities than traditional CNN. And different from LSTM, ID-CNN only

needs $O(n)$ time complexity for the processing sequence of sentences of length N even in the case of parallelism.

In this experiment, we put together 4 dilated CNN blocks of the same structure, and inside each block are three dilated convolution layers with dilation widths of 1, 1, 2. The implementation of ID-CNNs model is described as follows:

$$I_t = r\left(D_1^0 x_t\right) \tag{4}$$

$$c_t^0 = r(I_t) \tag{5}$$

$$c_t^l = r\left(D_i^{l-1} c_t^{l-1} + b_t\right) \tag{6}$$

$$a_t = concat\left(c_t^1, c_t^2, \ldots, c_t^l\right) \tag{7}$$

where

- D_i^{l-1} represents the l-th dilated convolutional layer of dilation width i;
- x_t, I_t, c_t^l represent the input vector which is the output of BERT, representation of ID-CNNs model and output vector of l-th convolution layer at time t;
- b_t is a bias vector at time t and $r(\cdot)$ denotes the ReLU activation function;
- a_t is the output of ID-CNNs model.

3.4.2. *Self-attention based ID-CNNs encoder.* Since ZeroNet Darknet sites have high text semantic sparseness and short sentences, while ID-CNNs coding layer has increased the expansion width, ignoring some semantic details, we need attention mechanism to solve this kind of problems. Based on the above considerations, we construct a self-attention layer above the ID-CNNs layer. Scaled dot-product attention (Figure 3 [21]) is faster and more space efficient in practice because it can be implemented using highly optimized matrix multiplication codes, which is consistent with the ID-CNNs layer. Therefore, the ID-CNNs coding layer based on the self-attention mechanism has fast calculation speed and low time complexity. After deep consideration, we implement a self-attention mechanism based on scaled dot-product attention, which help the deep learning model more focus on the main features of the current word, and can better improve the problem of ID-CNNs ignoring some local features.

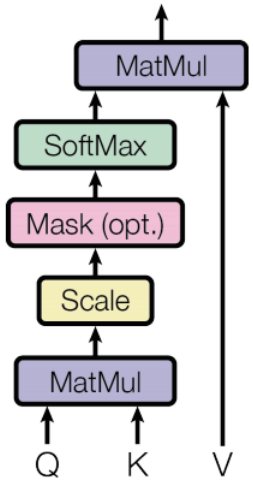


FIGURE 3. Scaled dot-product attention

The realization of the scaled dot-product attention mechanism is as follows:

$$Z_i = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{8}$$

$$Attention(Q, K, V) = concat(Z_i) \tag{9}$$

$$Q = W^Q * I \tag{10}$$

$$K = W^K * I \tag{11}$$

$$V = W^V * I \tag{12}$$

where d_k is the dimension of Q and K. The input of attention layer consists of queries and keys of dimension d_k, and values of dimension d_v. The self-attention layer computes the dot products of the query with all keys, and divides each by $\sqrt{d_k}$, then a SoftMax function is applied to calculating the attention score of the values. In practice, we calculate the attention function of a set of queries at the same time and pack them into a matrix Q. The keys and values are also packed together into matrices K and V.

3.5. **CRF decoder.** After encoding the BERT output sequence, just add a SoftMax layer to calculate the probability of each label that will ignore the dependencies across output labels in NER. For example, B-BW-PER is the start of a single person name with black words' format sequence and B-BW-PER must be followed by I-BW-PER instead of I-ORG or any other labels. Therefore, CRF layer is introduced as a decoder to reorder tags according to the tag information of adjacent data.

The feature functions used in CRF can be expressed as node feature function on Y node $S_j(y_i, x, i)$ and local context feature function of Y node $T_j(y_{i-1}, y_i, x, i)$, where x represents the input sequence, i is the current position, y_i is the current state, and y_{i-1} is the previous state. We use P to express the probability distribution function of CRF

$$P(Y|x, \lambda, \mu) = \frac{1}{Z(x)} \exp\left(\sum_i^n \sum_j^k \lambda_j T_j\left(y_{i-1}, y_i, x, i\right) + \sum_i^n \sum_j^l \mu_j S_j\left(y_i, x, i\right)\right) \tag{13}$$

where

$$Z(x) = \sum_{y \in Y} \exp\left(\sum_i^n \sum_j^k \lambda_j T_j\left(y_{i-1}, y_i, x, i\right) + \sum_i^n \sum_j^l \mu_j S_j\left(y_i, x, i\right)\right) \tag{14}$$

where $Z(x)$ is the normalization factor, and the value of S and T can only be 0 or 1. k is the total number of local context feature functions defined at the node, while l is the total number of node feature functions defined at the node and n is the length of the input sequence. λ_j and μ_j are weights of feature functions.

For a single output batch of encoding layer: $a = [(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)]$, in which m is the number of sequences in a single output batch of encoder, we use maximum likelihood estimation to estimate parameters λ and μ:

$$\lambda, \mu = \text{argmax}\left(\log\left(\prod_{k=1}^m P\left(y^k|x^k, \lambda, \mu\right)\right)\right) \tag{15}$$

In conclusion, the purpose of these models used in training stage is to maximize the log-probability of the correct tag sequence.

3.6. **BERT-ATT-IDCNNs-CRF model.** We combine the BERT model, an ID-CNNs network with self-attention layer and a CRF network to form a BERT-ATT-IDCNNs-CRF model. At the same time, we add three black words dictionaries as external knowledge to the model. The structure of the BERT-ATT-IDCNNs-CRF model with dictionary features proposed in this paper is shown in Figure 4.

There are a large number of black words in the text of the ZeroNet site, which are closely related to the text topic and are representative. Therefore, only a model pre-trained in regular sentences is not perfectly suitable for the text information of ZeroNet

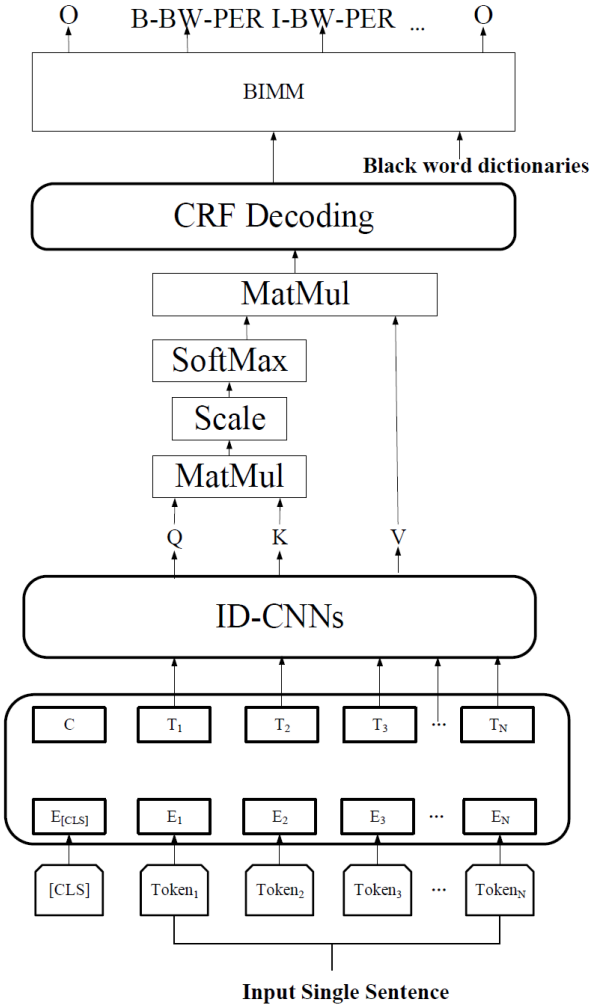


FIGURE 4. BERT-ATT-IDCNNs-CRF model with dictionary features

Darknet. Based on this characteristic, we construct three types of black words dictionaries: personal name black words dictionary, place name black words dictionary and organization/institution black words dictionary. After the CRF layer of the deep learning model, the bi-directional maximum matching method is used to segment single sentences, and each phrase is matched with phrases in various black words dictionaries to improve the recognition effect of the named entity of the ZeroNet Darknet black words. In order to further disambiguate the matching results, the word segmentation rules are defined in this paper as follows.

1) If the forward and backward word segmentation results are different, the result with a smaller number of word segmentation will be selected.

2) If the number of words in the word segmentation result is the same, a) If the word segmentation results are the same, it means that the word segmentation effect is in line with expectations, and the word segmentation methods do not have ambiguity, and the first one is returned as the word segmentation result; b) If the word segmentation results are different, indicating that the forward and backward segmentation methods have the problem of ambiguity, and the one with more words is returned as the result of word segmentation.

3) If there are matching results in the multi-class black word dictionary, take the longest matching type as the modified classification result.

4. Evaluation.

4.1. **Dataset construction.** This paper constructs the ZeroNet Darknet named entity recognition dataset in a semi-automatic way. This paper uses regular expressions to filter a large amount of label information in the texts of Darknet. After that, we load the processed content data file in json format and train an annotation model through the Wikipedia dataset and automatically build the initial Darknet dataset. Since there are many labeling errors in the automatically constructed dataset, we use YEDDA to refine and manually correct the entity tags in the initial Darknet dataset. We relabel the collected text in a two-column format, where the first column is the text character, and the second column is its related comments. This paper uses the BIO annotation mode, where the "B" header tag represents the starting position of the entity, the "I" header tag represents the internal and end positions of the entity, and the "O" tag represents the entity that is not of interest in this paper. We have labeled 18 different types of entities which are almost the same in the surface Internet, namely location (LOC), organization (ORG), person (PER), religious or political groups (NORP), buildings (FAC), countries and regions (GPE), products (PRODUCT), date (DATE), specific time (TIME), event (EVENT), books or pictures (WORK_OF_ART), law (LAW), language (LANGUAGE), percentage (PERCENT), currency (MONEY), measurement (QUANTITY), ordinal (ORDINAL), quantifier (CARDINAL). Aiming at the peculiarities of the Darknet, another three types of black words entities have been added, namely, person black words (BW-PER), location black words (BW-LOC), and organization black words (BW-ORG). The Darknet dataset has a total of 21 entity types. Finally, the annotated dataset (32228 sentences in total) is divided into training set, validation set and test set at a ratio of 5.5:1:1. The training set, validation set, and test set are randomly shuffled, so they are similar in distribution. The details of the DarkNER dataset are shown in Table 1.

4.2. **Baselines.** We compare our several BERT-based methods with LSTM, Bi-LSTM and CNN baselines with word2vec embedding methods: a Bi-LSTM with CRF decoding (Bi-LSTM-CRF [18]), CNN for feature extraction and Bi-LSTM combined encoding with CRF decoding (CNN-Bi-LSTM-CRF [20]). We also introduce the BERT finetuning method [13] as one of the baselines and compare our ID-CNNs-CRF and ATT-ID-CNNs-CRF model against it.

For baselines, we set the hyperparameters of the baselines to be the same as the recommended settings of the corresponding papers.

4.3. **Evaluation metrics.** In this paper, for multi-classification problems, we use multi-class evaluation metrics to evaluate the recognition performance of all Darknet named entity recognition models, including accuracy (Acc), precision (P), recall (R) and F1-score (F1), which are defined as below:

$$Accuracy = (TP + TN)/(TP + FP + FN + TN) \tag{16}$$

$$Precision = TP/(TP + FP) \tag{17}$$

$$Recall = TP/(TP + FN) \tag{18}$$

$$F1\text{-}score = 2 * (Recall * Precision)/(Recall + Precision) \tag{19}$$

where the terms true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) can be formulated in a $2 * 2$ confusion matrix, as Table 2.

To reduce variance of the experimental results, we repeat each experiment for 6 times and calculate the average results.

TABLE 1. Dataset details

Datasets	**Darknet Named Entity Recognition (DarkNER) dataset**			
Tag name	**Count**			
	Train	**Dev**	**Test**	**Ratio**
Sentences	23617	4305	4306	5.5:1:1
Tokens	874015	155213	160412	5.63:1:1.03
Tags	42263	7414	7813	5.7:1:1.05
Location	866	226	230	3.83:1:1.02
Organization	6182	966	1023	6.4:1:1.06
Person	9135	1620	1695	5.64:1:1.05
Political groups	2922	507	542	5.76:1:1.07
Buildings	295	56	57	5.27:1:1
Regions	10104	1890	1986	5.35:1:1.05
Products	138	20	21	6.9:1:1.05
Date	4372	688	705	6.35:1:1.02
Specific time	217	51	56	4.25:1:1.1
Event	1143	269	302	4.25:1:1.12
Books & pictures	452	58	54	7.8:1:0.93
Law	389	43	36	9.05:1:0.84
Language	109	23	24	4.74:1:1.04
Percentage	262	40	40	6.55:1:1
Currency	257	69	86	3.72:1:1.25
Measurement	94	15	25	6.27:1:1.67
Ordinal	614	103	106	5.96:1:1.03
Quantifier	4010	686	701	5.85:1:1.02
Person black words	98	35	35	2.8:1:1
Location black words	104	31	34	3.35:1:1.1
Organization black words	762	89	99	8.56:1:1.11

TABLE 2. Confusion matrix of TP, TN, FP and FN

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

4.4. **Training details.** In this study, all of the models were implemented in TensorFlow 1.13.1 and were trained on a single GTX 1660Ti GPU.

For the hyperparameters of our models, we adapt Adam as the default optimizer, with a learning rate of $5e^{-5}$, a weight decay rate of 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$. According to BERT pretraining hyperparameters, the max sequence length is set to 128. During training process, the batch size is set to be 8, and the number of training epochs is equal to 24. Proportion of training to perform linear warm-up is 0.1. Also, the dropout rate and gradient clip are applied for training with $dropout = 0.5$, $clip = 5$. The LSTM layer used in this paper has a size of 128 while the ID-CNN used three-layer dilated convolutional layer with dilation width $= [1, 1, 2]$.

5. Experimental Results and Discussion.

5.1. Results and analysis.

The results obtained on our Darknet named entity recognition test dataset are shown in Table 3. All of the baselines use word2vec embedding methods while our joint models use BERT methods. The 'BERT-ATT-IDCNNs-CRF' model uses scaled dot-product attention to focus on the semantic incompleteness and short sentences of texts in ZeroNet Darknet. The main feature is to estimate the degree of correlation between the current vector and other vectors.

TABLE 3. Performance comparation of baselines and our models, evaluated by accuracy, precision, recall and F1 (%)

Team name	Model	Features	Acc	P	R	F1
Al-Nabki et al. [18]	Bi-LSTM-CRF	High accuracy with simple pattern but low accuracy in real situation	95.68	73.81	69.31	71.49
Fan et al. [20]	CNN-Bi-LSTM-CRF	Overall good accuracy but low accuracy in some types of entities	96.16	76.63	75.42	76.03
Our	BERT-finetuning	Low accuracy in Darknet entity types	95.85	73.46	72.24	72.84
	BERT-CRF	Cannot fit the data well	96.12	76.35	73.38	74.83
	BERT-ID-CNN-CRF	High accuracy and an acceptable accuracy in Darknet entity type	96.87	77.32	77.61	77.47
	BERT-ATT-IDCNNs-CRF		96.97	78.09	78.71	78.40

As can be seen from Table 3, the performance of the most advanced model based on the word2vec algorithm is similar to the BERT-finetuning model, but the joint model based on the BERT pre-training is significantly improved over the state in all indicators. Among them, in all models, the BERT-ATT-IDCNNs-CRF model is based on the BERT-ID-CNN-CRF, the precision is increased by 0.77%, the recall rate is increased by 1.1%, and the F1 value is increased by 0.93%. The BERT-ATT-IDCNNs-CRF model constructed in this paper has the best performance, including accuracy, precision, recall and F1-score.

Finally, the F1-score is used to verify the entity recognition effect of 21 entity types including 3 types of black words. The named entity recognition models first load the ZeroNet Darknet dataset marked by the annotation module. Then 6 deep learning models are trained after loading the BERT pre-training parameters. The results are shown in Figure 5.

From the results, we can see that the 'Bi-LSTM-CRF' and 'CNN-Bi-LSTM-CRF' models have obvious shortcomings in the named entity recognition effect compared to the BERT-based model, and their effect on most types of entity recognition is not as good as that based on the BERT pre-training joint model. For PRODUCT entities, the reason for the poor recognition effect is that there are too few product entities in the Darknet dataset, so that the model is seriously over-fitting, and the recognition effect of new product entities is very poor. It is obvious that the six deep learning models are all too poor in the recognition of the three types of black words, and the F1 value of the best type of black words does not even exceed 40%. Therefore, based on the above reasons, this paper adds three types of black language dictionaries to the model, and uses the bidirectional

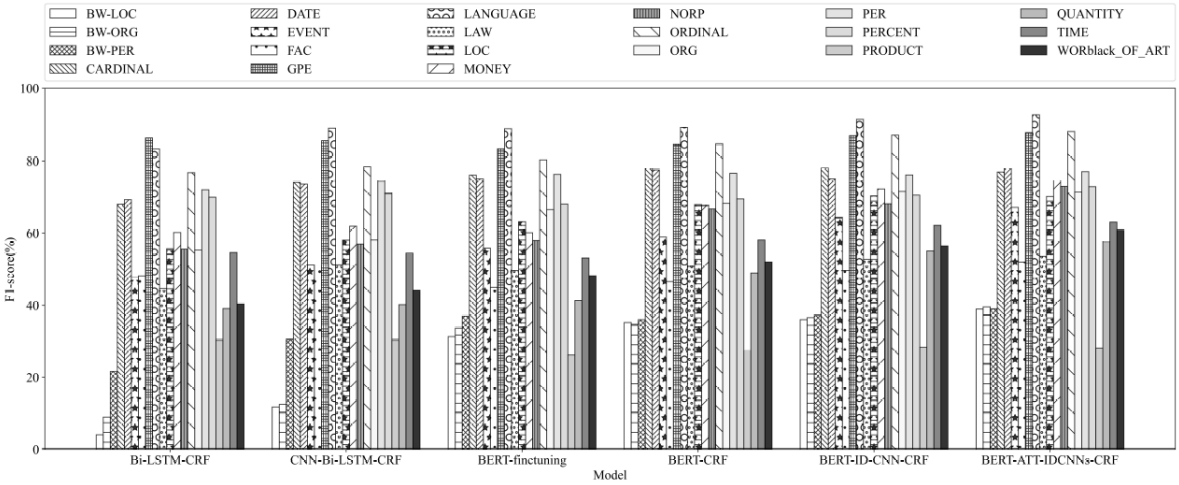


FIGURE 5. The recognition effect of all models on 21 types of Darknet entities with F1-score

TABLE 4. Performance on Darknet corpus with black words dictionaries features, evaluated by F1 (%)

Model	BW-PER	BW-ORG	BW-LOC
BERT-ATT-IDCNNs-CRF	39.03	39.58	38.97
BERT-ATT-IDCNNs-CRF-dictionaries	94.29	82.16	98.51

maximum matching algorithm (BIMM) to match the black words involved in the dictionary in the model to improve the recognition effect of the deep learning model on black words. The results are shown in Table 4.

It can be seen from Table 4 that the BERT-ATT-IDCNNs-CRF model with 3 types of dictionary features has greatly improved the recognition effect of black words in ZeroNet Darknet.

5.2. **Discussion.** Compared with the most advanced model in the field of Darknet named entity recognition, the BERT-ATT-IDCNNs-CRF model proposed in this paper, which integrates the features of three types of black words dictionaries, has achieved a certain degree of technological breakthroughs in the ZeroNet Darknet named entity recognition. The most important thing is that our model can better recognize multiple types of black words in the ZeroNet Darknet.

In the process of extracting entities from the texts of ZeroNet, the pre-trained "LOC", "PER" and "ORG" entity types have a high accuracy rate. Some entity types with simple models and simple semantics, such as country ("GPE"), and language ("LANGUAGE") have excellent recognition effects. However, the types of black words entities are mixed in Chinese and English, entity boundaries are blurred, and a large number of new black words types continue to appear, which makes it hard to accurately identify black words entities. At the same time, due to the blurring of the boundary between certain black words entities (such as strange person name) and other entity types, various black words entity types are often missed or incorrectly labeled as other entity types. The feature of the black words' dictionaries can significantly improve the recognition effect of black words in the Darknet dataset, which is a very important part for the recognition of Darknet entities.

6. **Conclusions.** This paper constructs 6 deep learning models to complete the named entity recognition task for ZeroNet Darknet, namely 'Bi-LSTM-CRF', 'CNN-Bi-LSTM-CRF', 'BERT-finetuning', 'BERT-CRF', 'BERT-ID-CNN-CRF', 'BERT-ATT-IDCNNs-CRF'. Among them, 'Bi-LSTM-CRF' and 'CNN-Bi-LSTM-CRF' two models are based on word2vec word vector embedding algorithm, while 'BERT-finetuning', 'BERT-CRF', 'BERT-ID-CNN-CRF', and 'BERT-ATT-IDCNNs-CRF' deep learning models are based on BERT pre-trained deep learning models. Metrics such as accuracy, precision, recall, and F1-score are used to evaluate the recognition effect of the model. Then, we evaluate the performance of the model used in the experiment on the ZeroNet Darknet named entity test set. From the experimental results, it can be seen that our four deep learning models based on BERT outperform the two deep learning models based on word2vec on the recognition effect of most entity types of the ZeroNet Darknet. The 'BERT-ATT-IDCNNs-CRF' model based on the self-attention mechanism constructed in this paper can achieve the best effect on the named entity recognition of most entity categories. However, all models have poor recognition effect of black words, so this paper proposes an algorithm that integrates three types of black words dictionaries features into a deep learning model, which greatly improves the effect of named entity recognition of black words for the ZeroNet Darknet.

In the future, we may consider pretrain the BERT model on a large unlabeled Darknet corpus. At the same time, we will increase the size of the ZeroNet Darknet named entity recognition dataset and manually correct wrong labeled tags to improve the accuracy of Darknet named entity recognition.

Acknowledgment. This work is partially supported by the National Key R&D Program of China (Grant Numbers: 2017YFC080700, 2016YFE0206700), and by Sichuan Science and Technology Program (Grant Number: 19ZDYF1987). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] L. F. Rau, Extracting company names from text, *Proc. of the 7th IEEE Conference on Artificial Intelligence Application*, vol.1, pp.29-32, 1991.

[2] S. Mariyah, I. Sutedja, R. Yulianto and S. Pramana, Reveal the customer behavior from business SMS text using named entity recognition, *ICIC Express Letters*, vol.13, no.8, pp.653-662, DOI: 10.24507/icicel.13.08.653, 2019.

[3] D. Farmakiotou, V. Karkaletsis, J. Koutsias, G. Sigletos, C. D. Spyropoulos and P. Stamatopoulos, Rule-based named entity recognition for Greek financial texts, *Proc. of the Workshop on Computational Lexicography and Multimedia Dictionaries*, pp.75-78, 2000.

[4] K. Shaalan and H. Raza, Arabic named entity recognition from diverse text types, *International Conference on Natural Language Processing*, pp.440-451, 2008.

[5] D. M. Bikel, S. Miller, R. Schwartz and R. Weischedel, Nymble: A high-performance learning name-finder, *arXiv Preprint*, cmp-lg/9803003, 1998.

[6] C.-L. Goh, M. Asahara and Y. Matsumoto, Chinese unknown word identification using character-based tagging and chunking, *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*, pp.197-200, 2003.

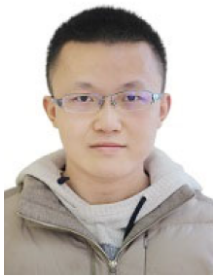
[7] A. McCallum and W. Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, *Proc. of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*, pp.188-191, 2003.

[8] Y. Bengio, R. Ducharme and P. Vincent, A neural probabilistic language model, *Proc. of the 13th International Conference on Neural Information Processing Systems*, vol.3, pp.893-899, 2000.

[9] T. Mikolov et al., Efficient estimation of word representations in vector space, *arXiv Preprint, arXiv: 1301.3781*, 2013.

[10] J. Pennington, R. Socher and C. D. Manning, Glove: Global vectors for word representation, *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp.1532-1543, 2014.
[11] Z. Huang, W. Xu and K. Yu, Bidirectional LSTM-CRF models for sequence tagging, *arXiv Preprint, arXiv: 1508.01991*, 2015.
[12] E. Strubell, P. Verga, D. Belanger and A. McCallum, Fast and accurate entity recognition with iterated dilated convolutions, *arXiv Preprint, arXiv: 1702.0209*, 2017.
[13] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *arXiv Preprint, arXiv: 1810.04805*, 2018.
[14] T. D. Tran, M. N. Ha, L. H. B. Nguyen and D. Dinh, Improving multi-grained named entity recognition with BERT and focal loss, *ICIC Express Letters, Part B: Applications*, vol.12, no.3, pp.291-299, DOI: 10.24507/icicelb.12.03.291, 2021.
[15] X. Guo, J. Ding, H. Jiang and Z. Chen, ZeroNet text content analysis based on semi-supervised LDA topic model, *Information Technology*, vol.44, no.3, pp.32-38, 2020 (in Chinese).
[16] E. Nunes et al., Darknet and deepnet mining for proactive cybersecurity threat intelligence, *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pp.7-12, 2016.
[17] C. Han, J. Shimamura, T. Takahashi, D. Inoue, M. Kawakita, J. I. Takeuchi and K. Nakao, Real-time detection of malware activities by analyzing Darknet traffic using graphical lasso, *2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/the 13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp.144-151, 2019.
[18] M. W. Al-Nabki, E. Fidalgo and J. V. Mata, DarkNER: A platform for named entity recognition in Tor Darknet, *Jornadas Nacionales de Investigación en Ciberseguridad (JNIC2019)*, vol.1, pp.279-280, 2019.
[19] M. W. Al-Nabki, F. Janez-Martino, R. A. Vasco-Carofilis, E. Fidalgo and J. Velasco-Mata, Improving named entity recognition in Tor Darknet with local distance neighbor feature, *arXiv Preprint, arXiv: 2005.08746*, 2020.
[20] X. Fan, A. Zhou, R. Zheng and M. Li, Research on named entity recognition in dark web market based on deep learning, *Information Security Research*, vol.7, no.1, pp.37-43, 2021 (in Chinese).
[21] A. Vaswani et al., Attention is all you need, *arXiv Preprint, arXiv: 1706.03762*, 2017.

Author Biography



Yuxuan Chen received the B.E. degree in Information Security from Sichuan University, China in 2018. He is currently working toward the M.E. degree in communication and information system with the Science and Technology on Communication Security Laboratory of The 30th Research Institute of China Electronics Technology Group Corporation. His research interests include natural language processing, knowledge graph, and intelligence security analysis.

Yubin Guo is an engineer in the Science and Technology on Communication Security Laboratory at The 30th Research Institute of China Electronics Technology Group Corporation. He received the M.E. degree in Electronic and Communication Engineering from University of Electronic Science and Technology of China, in 2014. His major research interests include dark web threat intelligence, malware analysis, intelligence security analysis and so on. He totally published 2 papers, 2 patents for invention, and participated in more than 10 projects as core staff including national key research and development project, National Natural Science Foundation of China and so on.

Hong Jiang received the B.E. and M.E. degrees from Sichuan University, Chengdu, China, in 1985, 1991, respectively. He is the deputy chief engineer of the Southwest Institute of Communications. His research interests include cyber security and network management.

Jianwei Ding is a senior engineer in the Science and Technology on Communication Security Laboratory at The 30th Research Institute of China Electronics Technology Group Corporation. He finished his Ph.D. in Computer Science and Technology Department at Tsinghua University, China in 2016. His major research interests include dark web threat intelligence, malware analysis, intelligence security analysis and so on. He totally published more than 10 papers, 3 patents for invention, and participated in more than 10 projects as core staff including national key research and development project, National Natural Science Foundation of China and so on.

Zhouguo Chen is a researcher-level senior engineer in the Science and Technology on Communication Security Laboratory at The 30th Research Institute of China Electronics Technology Group Corporation. He received B.E. and M.E. degrees from University of Electronic Science and Technology of China and he was employed as a visiting professor by University of Electronic Science and Technology of China. His major research interests include deep web and dark web monitoring, cognitive confrontation, traceback and so on. He totally published more than 20 papers, 5 patents for invention, more than 10 software Copyrights and participated in more than 20 projects as core staff including national key research and development project, National Natural Science Foundation of China and so on. He is responsible for or participated in more than 20 provincial and ministerial scientific research projects, such as deep web, dark web discovery and identification, identity correlation tracking, large-scale character portrait and other key research and development programs of the Ministry of Science and Technology, and the Natural Science Foundation.