Distant Supervision for Relations Extraction via Deep Residual Learning and Multi-instance Attention in Cybersecurity

Guowei Shen[1], Ya Qin[1], Wanling Wang[1], Miao Yu[2(✉)], and Chun Guo[1]

[1] College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
gwshen@gzu.edu.cn, qyamail@163.com, 1733348173@qq.com, gc_gzedu@163.com

[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
yumiao@iie.ac.cn

Abstract. A large number of open source threat intelligence resources provide regularly updated threat sources that can be applied to a variety of security analysis solutions. Fragmented security news, security forums, and vulnerability information are important sources of cyber threat intelligence, but it is difficult to correlate these multiple-source data. Cybersecurity knowledge graph is a powerful tool for data-driven thread intelligence computing. Relation extraction is a very important task in construction of cybersecurity knowledge graph from unstructured data. In order to reduce the influence of noisy data in deep learning model, we propose a distant supervised relation extraction model ResPCNN-ATT based on deep residual convolutional neural network and attention mechanism. This method takes word vector and position vector of the word as input of the model, extracts semantic features of texts through the piecewise convolutional neural network model PCNN, achieves the learning effect of less noisy data and better extracts deep semantic features in sentenses by using deep residuals Compared with other models, the model proposed in this paper achieves higher accuracy than other models.

Keywords: Thread intelligence · Cybersecurity knowledge graph · Relation extraction · Residual learning

1 Introduction

Currently, there are numerous open source threat intelligence sources providing periodically updated threat feeds fed into various analytical solutions. Security news, security forums, and vulnerability information are important data sources for cyber threat intelligence. However, the above data is fragmented and it is difficult to correlate such multi-source data.

Cybersecurity knowledge graph is a powerful tool for data-driven thread intelligence computing. Through cyber security knowledge graph, researchers can intuitively know network security entities and relations between the entities, such as utilization relation

D. Wang et al. (Eds.): SPNCE 2020, LNICST 344, pp. 151–161, 2021.
https://doi.org/10.1007/978-3-030-66922-5_10

between malware and vulnerabilities, employment relation between attackers and organizations, and ownership between software and vulnerabilities. Relation extraction is a very important task in construction of cybersecurity knowledge graph from unstructured data.

In relation extraction, the lack of labeled data for training is a challenge when constructing a network security knowledge graph. A common technique for coping with this difficulty is distant supervision in natural language processing. Distant supervision strategy is an effective method of automatically labeling training data. However, the assumption in distant supervision method is too strong, leading to the wrong label problem.

In this paper, we propose a novel cybersecurity relation extraction model ResPCNN-ATT combined **Res**idual Learning, **P**iecewise **C**onvolutional **N**eural **N**etworks (**PCNN**) and multi-instance **ATT**ention. The following list details the main contributions of the article:

- In order to reduce the impact of noise data in open source threat intelligence data sources, we propose a distant supervised cybersecurity relation extraction model based on ResPCNN-ATT. The model first uses the pre-trained word vector and the position vector between cybersecurity entity pairs as the model input, and then uses PCNN to extract the semantic features.
- Deep residual learning is used to solve the problem of gradient disappearance caused by noise data, so as to extract more effective semantic features.
- In order to better capture the more important semantic features in sentences, a multi-instance attention mechanism is used to calculate the correlation between instance and the corresponding relation to reduce the impact of noise data.

The rest of the paper is organized as follows. We describe related works in Sect. 2. The cybersecurity relation extraction model and details are shown in Sect. 3. Experiment is in Sect. 4. Section 5 draws conclusions.

2 Related Work

Data-driven cyber security event prediction and analysis are hot topics in current cyber security research [1]. Xiaokui Shu introduces a new methodology that models threat discovery as a graph computation problem for threat intelligence [2]. As a semantic knowledge base, knowledge graph is a powerful tool for managing large-scale knowledge consists with entities and relations between them. Haoze Yu proposed a relation extraction method for the construction of knowledge graph in food field [3].

Natural language processing technology [4–6] tends to only consider the domain name and IP address when analyzing the relation between malicious entities, both of which have very simple relation definitions. Aditya Pingle propose the RelExt [7] system, which strives to improve various cyber threat representation schemes, especially cybersecurity knowledge graphs (CKG), by predicting the relations between cybersecurity entities identified by cybersecurity named entity recognizer. VIEM [8] analyzes a large number of inconsistencies by extracting software names and software versions in Public Security Vulnerability Reports, so the extraction of relations is more complicated.

Relation Extraction (RE) is one of the most important topics in NLP. Many relation extraction methods have been proposed [9–11], such as bootstrapping, unsupervised relation discovery and supervised classification. Most existing supervised RE methods require a large amount of labelled relation-specific training data, which is very time consuming and labor intensive.

Distant supervision is proposed to automatically generate training data. Under the framework of distance supervised learning, some recent work [12–15] attempts to use deep neural networks in relation prediction. Although distant supervision is an effective strategy to automatically label training data, it always suffers from the wrong label problem.

3 The Proposed Model

In this section, we describe the overall architecture of our model. We introduce the components of our security entity relation extraction model one by one.

Under the framework of distant supervised learning, the problem of insufficient label data in deep learning can be solved, but at the same time it also brings some problems, such as the low-quality label data and the wrong label data. This would have a great impact on subsequent tasks of entity relation extraction. In view of the above problems, we propose a distant supervised relation extraction model ResPCNN-ATT based on deep residual neural network and attention mechanism. The framework is shown in Fig. 1. The model is mainly composed of a vector representation layer, a deep residual convolutional network layer, and a multi-instance attention layer.

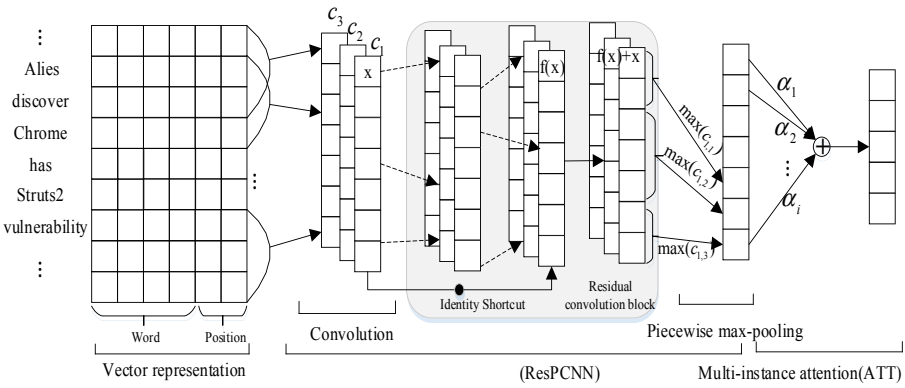

Fig. 1. Cybersecurity relation extraction model based on ResPCNN-ATT

The model first uses the pre-trained word vector and the position vector between entity pairs as input, which can highlight the role of the two entities, and then uses the piecewise convolutional neural networks to extract semantic features. At the same time, deep residual learning is introduced to solve the problem of gradient disappearance caused by noise data, so as to extract more effective semantic features. Finally, in order

to better capture the more important semantic features in sentences, the multi-instance attention mechanism is used to calculate correlation between instances and corresponding relation, so as to reduce the impact of noise data and improve the performance of relation extraction.

3.1 Vector Representation

The vector representation layer in the model mainly includes word embedding and position embedding.

Word Embedding. Before training the relation extraction model, the text data needs to be vectorized so that the model can read the data. Compared with traditional one-hot coding, word vector mapping can represent more semantic and syntactic information. Word vector mapping is to map each word in the text to a k-dimensional real-valued vector. It is a distributed representation of words. When training a neural network model, the most common method is to randomly initialize all parameters and then use an optimization algorithm to optimize the parameters. Research shows that when a neural network is initialized with a pre-trained word vector, the parameters can be converged to a better local minimum.

For a given sentence $X = \{x_1, x_2, \ldots x_n\}$ consisting of n words, use word2vec to map each word to a low-dimensional real-valued vector space, then perform word vector processing on the sentence, and finally get a vector representation of each word in the sentence, to form a word vector query matrix $\boldsymbol{D^c}$. Each input training sequence can be mapped by the word vector query matrix $\boldsymbol{D^c}$ to obtain the corresponding real-valued vector $\boldsymbol{x}_t = \{w_1, w_2, \ldots w_n\}$.

Position Embedding. In the relation extraction task, we focus on finding the relation of entity pairs. Words that are often close to the entity are more able to highlight the relation between the two entities, such as some verbs: attack, use, etc. Therefore, in order to make full use of the information in the sentence, the position of each word in the sentence for two entities is an important feature in the relation extraction task. This paper uses the position vector (Position Embeddings, PE) mapping representation method proposed by zeng [14] et al., that is, the relative distance between the current word,entity e_1 and entity e_2 is stitched and converted into a vector representation through embedding. In sentence position vectorization, if the dimension of the word vector is d^c and the dimension of the position vector is d^p, then the dimension of the sentence vector is:

$$d^s = d^c + d^p * 2 \tag{1}$$

For example, the vectorized representation of "Alies discover Chrome has XSS vulnerabilities" is shown in Fig. 2, "Chrome" and "XSS" in the sentence correspond to entities e_1 and entities e_2 respectively. Then the distance from "Alies" to "Chrome" is 2, the distance from "Alies" to "XSS" is 4, and the distance from "vulnerability" to "Chrome" is -3, the distance from "vulnerability" to "XSS" is -1.

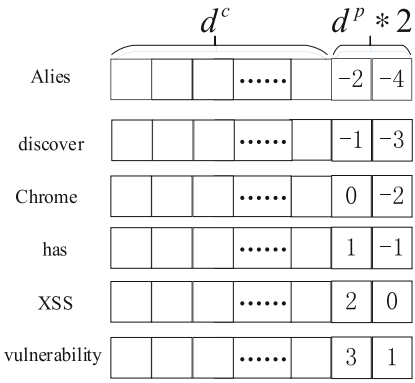

Fig. 2. Position embedding

3.2 Deep Residual Neural Network

In cyber security relation extraction tasks, the main challenge is that the length of the input sentence is variable and not fixed, and important feature information may appear in any area of the sentence. Therefore, in order to be able to use all local features and predict relations globally, this paper uses a piecewise convolutional neural network PCNN model to extract semantic features in sentences.

In this paper, a residual convolution block is designed for residual learning. Each residual convolution block is a sequence composed of two convolution layers. After each convolution layer, the activation function ReLU is used for nonlinear mapping, and features are then extracted using a local maximum pool. The kernel size of all convolution operations in the residual convolution module is w, and the newly generated features are guaranteed to be the same size as the original ones through the border padding operation. The convolution kernels of the two-layer convolution are $\boldsymbol{W}_1, \boldsymbol{W}_2 \in R^{w*1}$. The first layer of the residual convolution block is:

$$\boldsymbol{c}_{i,1} = f(\boldsymbol{W}_1 \bullet \boldsymbol{c}_{i,i+\mathrm{w}-1} + \boldsymbol{b}_1) \tag{2}$$

The second layer is:

$$\boldsymbol{c}_{i,2} = f(\boldsymbol{W}_2 \bullet \boldsymbol{c}_{i,i+\mathrm{w}-1} + \boldsymbol{b}_2) \tag{3}$$

Where $\boldsymbol{b}_1$,$\boldsymbol{b}_2$ are bias vector. In this paper, we optimize the residual learning to get the output vector $\boldsymbol{c}$ of the residual convolution block [16, 17].

After the semantic feature is acquired by convolution layer, the most representative local feature is further extracted by pooling layer. In order to capture characteristic information of different sentence structures, a Piecewise Max Pooling process is used.

3.3 Multi-instance Attention

In the relational extraction model, sentence-level attention is built on multiple instances, dynamically reducing the weight of noisy instances, and making full use of semantic information in these sentences to obtain final sentence vector representation.

For the instance set $\boldsymbol{S} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \cdots \boldsymbol{g}_n)$ describing the same entity pair $< e_i, e_j >$, $\boldsymbol{g}_i$ is the instance vector output by the convolution layer, n is the number of instances contained in the set $\boldsymbol{S}$. This paper will calculate the correlation degree between the instance vector $\boldsymbol{g}_i$ and the relation r. In order to reduce the impact of noise data and make full use of the semantic information contained in each instance in the set, the calculation of instance set vector $\boldsymbol{S}$ will depend on each instance $\boldsymbol{g}_i$ in the set:

$$\boldsymbol{S} = \sum_i \alpha_i \boldsymbol{g}_i \tag{4}$$

Where a_i is the weight of the input instance vector $\boldsymbol{g}_i$, which measures the correlation of the corresponding relation r. The calculation formula of α_i is as follows:

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)} \tag{5}$$

e_i is a query-based function, which indicates the degree of matching between the input instance vector $\boldsymbol{g}_i$ and the prediction relation r.

Conditional probability of prediction relation $p(\boldsymbol{R}|\boldsymbol{S})$ is calculated by softmax function:

$$p(\boldsymbol{R}|\boldsymbol{S}) = soft\max(\tilde{\boldsymbol{r}}\boldsymbol{S} + \boldsymbol{b}) \tag{6}$$

Where $\tilde{\mathbf{r}}$ is the relation matrix and $\boldsymbol{b}$ represents the bias vector. $p(\boldsymbol{R}|\boldsymbol{S})$ is used to predict the relation between pairs of cyber security entities.

$$\tilde{R} = \arg\max p(\boldsymbol{R}|\boldsymbol{S}) \tag{7}$$

4 Performance Evaluation

In this section, we empirically demonstrate the performance of the proposed method on dataset CSER.

4.1 Datasets

In order to verify the performance of our proposed model, we build a Cyber Security Entity Relation dataset CSER. 10 types of relation were labeled. The dataset CSER is clawed from Freebuf website and wooyun vulnerability database, which includes network text data such as technology sharing, network security, and vulnerability information.

Commonly used Precision-Recall (P-R) curve, AUC value and average accuracy (P@N) are used to evaluate the model. The P-R curve is a curve drawn with the recall rate R as the abscissa and the accuracy rate P as the ordinate, using P and R at different confidence levels. The AUC value is the area included under the P-R curve. Generally, the larger the AUC value is, the better the model perform. P@N is the accuracy rate calculated by comparing the first N relation instances.

The set of dimensions of the word vector is {50, 60, · · · , 300}. The set of dimensions of the position vector is {1, 2, · · · , 10}. During the training process, the Adam optimizer performs optimization training. The value set of the learning rate is {0.01,0.001,0.0001}. The set of batch size processed in one iteration is {40, 160, 640, 1280}. In order to prevent the model from overfitting, the dropout method is used in CNN. Other parameters are shown in Table 1.

Table 1. Parameters

Parameters	Value
CNN window size	3
CNN hidden size	230
Learning rate	0.01
Batch size	160
Epoch	60
Dimension of the position vector	5
Dropout rate	0.5
Dimension of the word vector	50

4.2 Results

The experimental comparison in this paper mainly compares two aspects of the models.

On the one hand, it uses CNN algorithm with different performance to encode the training data and extract the semantic features in the sentence, mainly including the traditional models: CNN, PCNN, and ResPCNN.

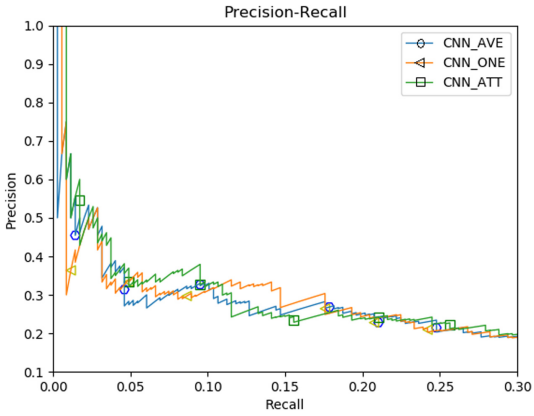

Fig. 3. The results of different bag methods AVE/ONE/ATT based on CNN

The second aspect is based on how CNN/PCNN/ResPCNN uses the information in the packaging bag for experimental comparison. Three different methods were used to process the information in the bag, namely AVE, ONE, and ATT. AVE assigns the same weight to all the sentences in the packet as the entity pair, that is $\alpha_i = 1/n$. ONE means to take the instance vector with the highest confidence, and find a sentence with the highest score from each bag to represent the entire bag. All models in this paper have been trained and tested on the dataset CSER. Figures 3, 4 and 5 show the P-R curves of the result on different bag models.

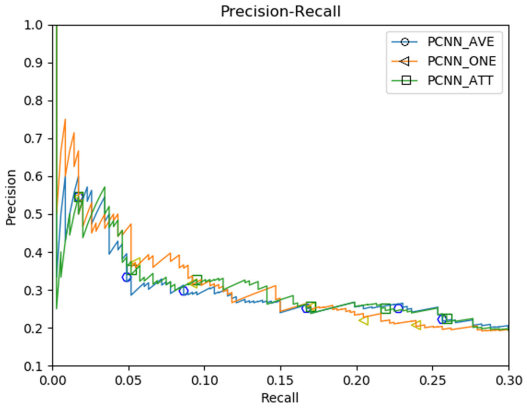

Fig. 4. The results of different bag methods AVE/ONE/ATT based on PCNN

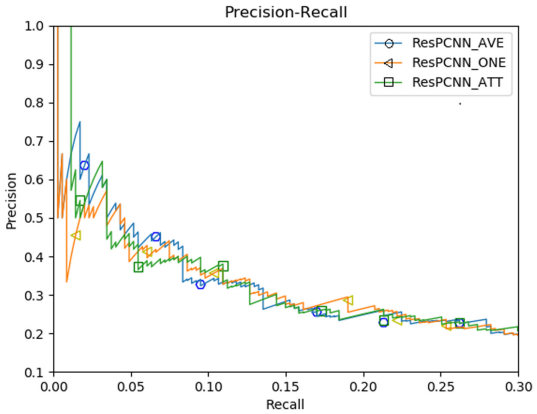

Fig. 5. The results of different bag methods AVE/ONE/ATT based on ResPCNN

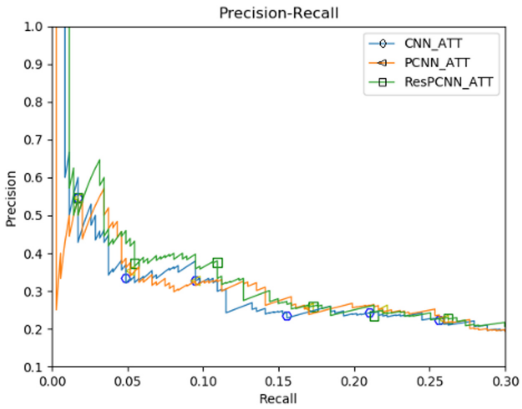

Fig. 6. The results of different sentence semantic feature extraction models CNN/PCNN/ResPCNN

From Fig. 6, the AUC value of the model ResPCNN-ATT is the highest value on the dataset CSER, which reachs 12.68%. The model ResPCNN-ATT proposed in this paper can better extract the deep semantic information of sentences, indicating that the introduction of the ATT method can effectively reduce the redundant data in distant supervised learning.

Table 2. Results for the first 100, 200, and 300 extracted relation instances upon manual evaluation.

Models	P@100	P@200	P@300	Mean	AUC
CNN+AVE	0.3267	0.2537	0.2452	0.2743	0.1062
CNN+ONE	0.2971	0.3035	0.2392	0.2799	0.1096
CNN+ATT	0.3267	0.2437	0.2425	0.2710	0.1121
PCNN+AVE	0.2971	0.2587	**0.2645**	0.2727	0.1096
PCNN+ONE	0.3168	0.2587	0.2358	0.2705	0.1109
PCNN+ATT	0.3267	0.2736	0.2525	0.2842	0.1121
ResPCNN+AVE	0.3267	0.2686	0.2458	0.2804	0.1205
ResPCNN+ONE	0.3564	0.2786	0.2558	0.2969	0.1184
ResPCNN+ATT	**0.4158**	**0.3084**	0.2558	**0.3267**	**0.1268**

As can be seen from Table 2, comparing the accuracy of the first 100, 200, and 300 relation instances on the dataset CSER, the relation extraction accuracy of ResPCNN-ATT is the highest, which reaches 32.67%. However, the accuracy of the CSER dataset is lower than other datasets. This is because the sentences in the CSER dataset are mixed with Chinese and English, the more complicated the sentence structure is, the less obvious the entity relation characteristics are, and the less the corpus data is.

In order to further analyze the relation extraction model proposed in this paper, by adding the depth of the ResPCNN-ATT model to verify the effectiveness of the introduction of residual learning, comparative experiments of convolutional layers with different depth is designed. In this paper, the number of convolutional layers is increased by increasing the number of residual convolution blocks, and the experimental comparison is performed on the CSER dataset. Figure 7 shows the P-R curves on models with different depth.

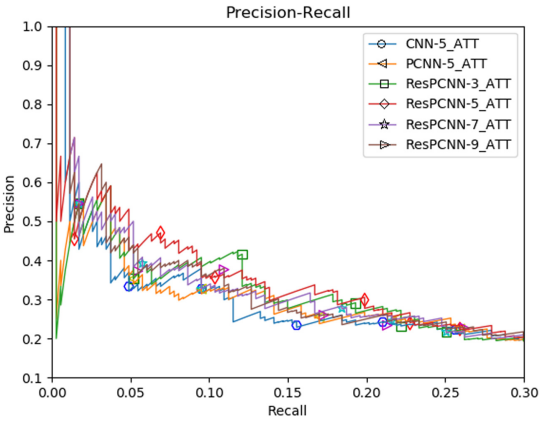

Fig. 7. The results on models with different depth

5 Conclusions

In this paper, we introduce a novel distant supervised cybersecurity relation extraction model ResPCNN-ATT. Algorithm ResPCNN is used to extract semantic features. Deep residual learning is introduced to solve the problem of gradient disappearance due to noise data. The mechanism calculates the correlation between the instance and the corresponding relation to reduce the impact of noisy data. The experimental results show that the model proposed in this paper has the highest accuracy of relation extraction compared with other model methods.

In the future, we intend to use reinforcement learning to further solve the problem of noise in the training data automatically generated by the distant supervised method.

Acknowledgement. Project supported by the National Natural Science Foundation Of China (No. 61802081).

References

1. Sun, N., Zhang, J., Rimba, P., et al.: Data-driven cybersecurity incident prediction: a survey. IEEE Commun. Surv. Tutor. **21**(2), 1744–1772 (2018)

2. Shu, X., Araujo, F., Schales, D.L., et al.: Threat intelligence computing. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1883–1898 (2018)
3. Yu, H., Li, H., Mao, D., et al.: A relationship extraction method for domain knowledge graph construction. World Wide Web **23**, 1–19 (2020). https://doi.org/10.1007/s11280-019-00765-y
4. Liao, X., Yuan, K., Wang, X.F., et al.: Acing the IOC game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 755–766 (2016)
5. Siracusano, G., Trevisan, M., Gonzalez, R., et al.: Poster: on the application of NLP to discover relationships between malicious network entities. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 2641–2643 (2019)
6. Zhu, Z., Dumitras, T.C.: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 458–472. IEEE (2018)
7. Pingle, A., Piplai, A., Mittal, S., et al.: RelExt: relation extraction using deep learning approaches for cybersecurity knowledge graph improvement. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 879–886 (2019)
8. Dong, Y., Guo, W., Chen, Y., et al.: Towards the detection of inconsistencies in public security vulnerability reports. In: 28th {USENIX} Security Symposium ({USENIX} Security 19), pp. 869–885 (2019)
9. Socher, R., Huval, B., Manning, C.D., et al.: Semantic compositionality through recursive matrix-vector spaces. In: Joint Conference on Empirical Methods in Natural Language Processing & Computational Natural Language Learning, pp. 1201–1211 (2012)
10. Daojian, Z., Kang, L., Siwei, L., et al.: Relation classification via convolutional deep neural network. In: Proceedings of COLING, pp. 2335–2344 (2014)
11. Zhou, P., Shi, W., Tian, J., et al.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 207–212 (2016)
12. Santos, C.N.D., Xiang, B., Zhou, B.: Classifying relations by ranking with convolutional neural networks. Comput. Sci. **86**(86), 132–137 (2015)
13. Lin, Y., Shen, S., Liu, Z., et al.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 2124–2133 (2016)
14. Zeng, D., Liu, K., Chen, Y., et al.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1753–1762 (2015)
15. Qin, P., Xu, W., Wang, W.Y.: Robust distant supervision relation extraction via deep reinforcement learning. arXiv preprint arXiv:1805.09927 (2018)
16. Huang, Y.Y., Wang, W.Y.: Deep residual learning for weakly-supervised relation extraction. arXiv preprint arXiv:1707.08866 (2017)
17. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. Comput. Vision Pattern Recognit. 770–778 (2015)