2021 Ninth International Conference on Advanced Cloud and Big Data (CBD) | 978-1-6654-0745-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/CBD54617.2021.00026

Vulnerability knowledge extraction method based on joint extraction model*

Zhen Liu[1,2], XiaoQiang Di[1,2,3], Wei Song[1,2], WeiWu Ren[1,2,(✉)]

1 Jilin Key Laboratory of Network and Information Security,Changchun University of Science and Tecchnology,Changchun,China

2 School of Computer Science and Technology,Changchun University of Science of Technology,Changchun,China

3 Information Center,Changchun University of Science and Technology,Changchun,China

renww@cust.edu.cn

***Abstract*—Information extraction is an important semantic processing task to construct network security knowledge graph. Extracting entities and relationships in vulnerability description from public data sets will inevitably lead to waste of manpower and difficulty in accurate positioning. Another challenge is that there are multiple relationships among vulnerable descriptors. This paper proposes a framework for the common vulnerabilities and exposures (CVE) analysis, which consists of entity annotation algorithm and relational classification model. In particular, we apply the model to CVE dataset to solve the problem of information extraction and relationship classification in the CVE vulnerability analysis. Moreover, the predicted relationship is used to construct vulnerability security knowledge graph. The experimental results show that the framework can deal with the CVE vulnerability description effectively, and has good relationship classification performance.**

***Index Terms*—relation classification, vulnerability security, CVE, information extraction, knowledge graph**

I. Introduction

In computer and network systems, more and more security vulnerabilities are found, which seriously threaten the network security of users, organizations and even the whole country. Strong loopholes often lead to social unrest and insecurity. For example, the attack range covers 400 web sites and servers of OpenSSL tennis dungeon loopholes and global blackmail attacks WannaCry, resulting in more than 150 injured countries, and governments, enterprises, medical institutions, universities and other industries all have IT devices. Therefore, CVE program emerges as the times require, which is used to discover and repair vulnerabilities and avoid being exploited by attackers. CVE is a global database maintained by security providers, developers and researchers, which use the power of the crowd to index common vulnerabilities. So far, more than 200,000 vulnerabilities have been indexed, which is inseparable from the contribution of the above three types of people. Therefore, CVE plays an important role in guiding the increase of security. Moreover, CVE data stream has cooperated with hundreds of security providers around the world.

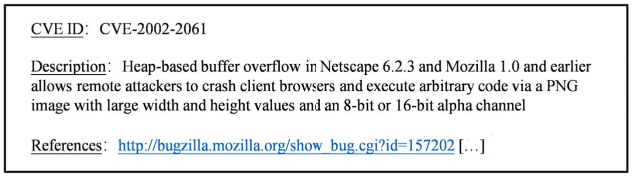
Although CVE database has a large number of vulnerabilities with data collected every day, and the vulnerability description contains the root causes, consequences and attack mechanisms of vulnerabilities. Unfortunately, identifying vulnerability information described by CVE contributors is a manual, labor-intensive and asynchronous process, and the description needs to be reviewed under the condition of sufficient security knowledge background and familiarity with entities. However, people prefer to get comprehensive and high-quality information without relying on human and expert identification. Specifically, vulnerability information on CVE may be scattered, wrong and outdated, which may make it difficult for researchers to accurately locate the vulnerability information, and then lead to the extension of patch development time. What's worse, there are many entities and relationships in the vulnerability description, which are not clearly pointed out, which makes it difficult for machine learning algorithm or neural network to directly find the software or software platform in which the vulnerability takes effect, the causes and consequences of the vulnerability, the way of vulnerability attack and who the attacker is. Therefore, the traditional supervised machine learning technology and Natural Language Processing method is applied to CVE data sets, and the performance of entity recognition and relational classification is often lower than expected.

Figure 1 shows a CVE data. From the description of this data, we can easily see that the name of the vulnerability is heap based buffer overflow, the software that the vulnerability takes effect is Netscape 6.2.3 and Mozilla 1.0, the attacker is remote attackers, the attack mode is PNG image, and the result is crash client browsers and execute array code. But the computer cannot accurately identify these entities and their relationship with the vulnerability. Our goal is to extract the entity in the description and determine the relationship between the entity and CVE ID using our proposed method.

In this paper, we conduct an experiment to solve the above problems. In this experiment, we first use information retrieval and natural language processing (NLP) technology to analyze the entities and relationships contained in each CVE vulnerability report according to its description. Then we use

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2018YFB1800303, the Science and Technology Planning Project of Jilin Province under Grant No. 20190302070GX, the Science and Technology Projects of Jilin Provincial Education Department under Grant No. JJKH20200795KJ, the Science and Technology Projects of Jilin Provincial Education Department under Grant No. JJKH20200793KJ.

978-1-6654-0745-8/21/$31.00 ©2021 IEEE
DOI 10.1109/CBD54617.2021.00026



CVE ID： CVE-2002-2061

Description： Heap-based buffer overflow in Netscape 6.2.3 and Mozilla 1.0 and earlier allows remote attackers to crash client browsers and execute arbitrary code via a PNG image with large width and height values and an 8-bit or 16-bit alpha channel

References： http://bugzilla.mozilla.org/show_bug.cgi?id=157202 [...]

Fig. 1. A piece of CVE data

natural language processing technology, or more specifically, relationship classification technology, The semantic relationship between noun pairs can be predicted automatically from the description text. We build a model which is composed of a bi-directional long short-term memory (bi-LSTM) network and sentence level attention mechanism. We use word embedding and location embedding to process entity, relationship and text as the input of the model, supplemented by attention mechanism, which can effectively mine the hidden information in sentences. At the same time, the model itself is based on reinforcement learning, which can effectively deal with the noise problems in the data set. Using this NLP algorithm, we have achieved unprecedented performance. Finally, we construct the CVE vulnerability knowledge graph in the form of ”entity relationship entity” triplet, and analyze the similarity between label relationship and prediction relationship. We conclude from our experiments that it is feasible to deal with CVE vulnerability description information in our way. This automatic method can provide valuable vulnerability information for developers without security background, so that they can better analyze the attack source, attack method and attack path. In addition, the construction of the vulnerability knowledge graph can effectively and intuitively display the attributes and correlations of the CVE security vulnerability data, and deeply explore the intrinsic value of the vulnerability data.

Our contribution to this paper:

- This paper presents an algorithm that can extract entities and relationships in CVE data description effectively. It can label entities and relationships of CVE data description without manual identification, and build CVE vulnerability security knowledge graph.
- Extracted CVE data sets are used to train the joint model of reinforcement learning and bi-LSTM network, and the experimental results show that the performance of the model is better.
- The relationship of model prediction and labeled entities is used to build vulnerability security knowledge graph. Comparative experimental results show that the relationship of model prediction has good accuracy and integrity in building security knowledge graph.

II. Related work

In recent years, in the field of network security knowledge graph research, most scholars mainly from the network security knowledge representation [1], network security knowledge graph ontology construction [2], information extraction and other aspects of research, and then build network security knowledge graph. The network knowledge representation and reasoning framework proposed in reference [3] combines the semantic based and rule-based reasoning methods to transform the original data into knowledge. In literature [4], the semantic information model of cyberspace and social media data is established, and the knowledge representation and management are carried out. Network security knowledge graph ontology construction is based on knowledge representation, analyzes the structure and relationship of knowledge, and visualizes knowledge in a graphical way. Literature [5] integrates various structured and unstructured data, including 15 entities and 115 attributes, and constructs an ontology in the field of network security. The most common understanding is the [6], whose knowledge graph of Cyberspace Security integrates the public knowledge of CWE, CVE and CAPEC.

Information extraction technology is also very important to realize the construction of network security knowledge graph. The current network security knowledge graph only focuses on the association of existing public knowledge, and does not take into account the use of other open-source data, especially unstructured data, to enrich the nodes and relationships of the knowledge graph, or to dig deeper into the hidden information in the existing knowledge. However, there are three main problems in extracting information from unstructured network security texts. First, extracting network security knowledge requires professional domain-related theories, so information extraction tasks lack a large amount of labeled training data. At the same time, network attack incidents are often composed of one or more attack patterns, which include multiple heuristics and completed attacks. Each type of attack can be treated as a separate network security event, which makes network security analysis more difficult. In addition, there are many hidden entities and relationships in unstructured data that cannot be discovered, and the hidden information may not contain semantic information.

The main tasks of information extraction are named entity recognition (NRE) and relation extraction (RE). Named entity recognition method [7] is mainly divided into three categories: rule based, statistical machine learning and deep learning. The relationship between entities is the key part of building a knowledge graph. How to identify the relationship between entities of unstructured text has become one of the key tasks of building knowledge graph [8]. In recent years, scholars mainly study the relationship extraction method based on distance supervised learning [9]. The distance supervised learning method will add noise to the training set, and scholars further propose relationship extraction methods such as multi-instance learning, sentence level attention mechanism, confrontation training, and reinforcement learning [10]–[14]. The results of named entity recognition will affect the performance of relation extraction model and lead to error propagation. Therefore, in order to improve the learning ability of the model, scholars further improved the series process of entity extraction and relationship extraction, and proposed entity and relationship joint extraction [15]. At present, the main joint extraction methods are entity relationship extraction method based on parameter sharing [16], entity relationship extraction method based on sequence annotation [17], [18] and entity relationship extraction method based on graph [19], [20].

III. Methods

The overall architecture of the method proposed in this paper consists of two parts: CVE vulnerability entity relative data set and joint neural network model. We use the entity relation annotation algorithm to process the original CVE data set and label the relationship and entity in the CVE description to generate a new CVE data set. The joint neural network model is trained with the generated data set, and a large number of experiments are carried out on the test set.

A. Dataset

In order to evaluate our proposed method and test its performance, we collected public CVE datasets for nearly 20 years. In the dataset, we only focus on the description field in the CVE entry. We sample these vulnerabilities reports marking the entities and relationships in the CVE description, and use the market data to evaluate the performance of our model.

For each CVE data, we first clean the data, delete the status, reference, phase, votes and comments fields of the data, and only keep the CVE ID and description fields of the data, because CVE ID is the unique identifier of the publicly disclosed vulnerability, and more entities and relationships can be found in the description, which provides three tuples for the next step of building vulnerability security knowledge graph. At the same time, we found that there were unpublished, merged or revoked CVE entries in the extracted CVE data. We filtered the data out. Prevent this data from affecting the performance and accuracy of the model. A total of 30000 CVE entries were extracted, covering 82986 entities and 5 relationships between entities and vulnerability IDs. Among the two entities marked by each sentence, one is the CVE identifier, and another entity is the vulnerability attack category. Vulnerability attack categories include vulnerability attack environment, vulnerability attack object, vulnerability attack method and vulnerability attack result. Among them, the attack environment includes the operating system and software version; the target of the attack refers to the permission of the vulnerability to attack and the target of the vulnerability attack; the attack method refers to the vulnerability category, including SQL Injection, Buffer Overflow and Gain Privileges, etc. the result of the attack are often different in different attacks. As shown in Table 1, the relationship we mark also corresponds to the marked entity. We mark the entity pairs that have no relationship as NA.

For CVE description, we extract text information, delete all network links, and use nltk toolkit [21] to mark the part of speech of words to facilitate the judgment in the algorithm. We do not delete any symbols from unstructured text information, because they are usually part of the above marked entities. We select the keywords associated with different relationships, the keywords of attack environment are In, On and Under, the key words of attack object are Allow, the key words of attack method are Via and By, and the key words of attack result are To. Through our observation and research on the CVE description, we find that there is more than one entity related to vulnerability information near these keywords, so we choose these words as the keywords of entity annotation.

TABLE I
The meaning of various relationships

Relations	Values	relational meaning
NA	0	No relation
/cve/environment/need	1	Vulnerability attack environment
/cve/user/allow	2	Target of vulnerability attack
/cve/attack/method	3	Vulnerability attack method
/cve/result/cause	4	Vulnerability attack results

In the method described in this paper, it is very important to mark and build data sets. The process of entity relationship annotation is shown in Figure 2. Firstly, we clean the data, extract the vulnerability attack environment, vulnerability attack object, vulnerability attack method and vulnerability attack result in CVE description, and then form entity pairs with CVE ID, and mark them as corresponding relationships. Finally, we add noise data into the dataset by randomly adding NA relation and swapping entities. Taking extraction vulnerability attack as an example, we describe our algorithm of extracting entity in algorithm 1. As described in line 2, traverse each word in the sentence, and mark the position of attack method according to whether the word is the corresponding keyword. The 5 line intercepts part of the sentence according to the position of the marked entity, which is used to judge the following keywords and label the entity, such as line 6 and line 7. In line 9, we delete the entity whose length is empty.

```
Algorithm 1 extraction vulnerability attack algorithm
Input: A sentence of CVE description
Output: All the vulnerability attack methods and corresponding sentences
1: for  word in X do
2:     Mark the location of the vulnerability attack method X = $\{x_1, x_2, \ldots, x_n\}$
3: for $x_i \in X$ do
4:     According to the index position, the sentence is intercepted
5:     Judge word character and keyword
6:     Intercepting words to form entities
7:     Replace words in sentences with entities
8:     Delete empty entities
```

B. Model

As shown in Figure 3, the joint neural network model proposed in this paper consists of two parts: a reinforcement learning model and a neural network model. We model the noise problem in the data set as a reinforcement learning process. With the training, the reinforcement learning model

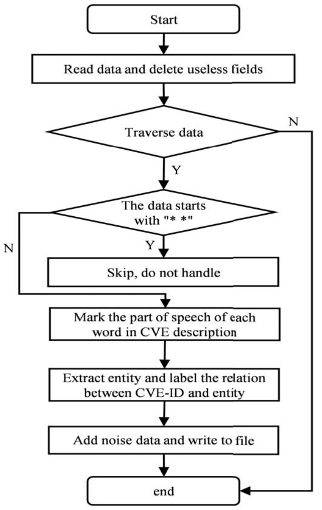

Start
Read data and delete useless fields
Traverse data
N
Y
The data starts with "* *"
N
Y
Skip, do not handle
Mark the part of speech of each word in CVE description
Extract entity and label the relation between CVE-ID and entity
Add noise data and write to file
end

Fig. 2. Flow chart of entity relation annotation

has stronger ability to recognize and distinguish high-quality sentences from the data set, and can update its own strategic network according to the reward obtained in the neural network model. Neural network model is an end-to-end model composed of attention mechanism and bi-LSTM network. In the training process, the model analyzes the semantic information of sentences and uses an attention mechanism to select the words that have an influence on classification, and then predicts the tags of relationships in sentences.

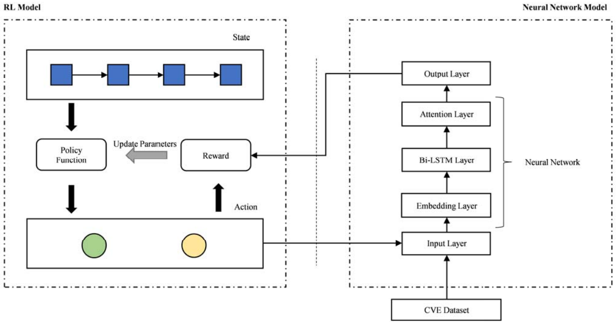


Fig. 3. Joint neural network model

Specifically, in the training process, we transform each sentence in CVE dataset into a vector composed of the position vector and the word vector. First, we deal with the vector representation of each word in the sentence. The word vector is the vector form of the word obtained by looking up the word2vec embedding matrix, and the position vector is the vector representation of the relative distance between the current word and two entities in the sentence. Then, we connect the word vector and the position vector of each word to form a new vector, which is finally input into the bi-LSTM layer. In the bi-LSTM layer and the attention layer, our goal is the same, which is to analyze the semantic information in sentences and discover the hidden entities and information. Bi-LSTM mines the hidden information in sentences through forward and backward LSTM. The attention mechanism is to analyze the weight of the input sentence vector through trained context vector parameters to confirm that the word or entity needs more attention, so as to increase the processing ability of bi-LSTM layer of sentence semantic information.

IV. Experiment

A. Data set and evaluation indicators

To evaluate our model, we randomly sampled 30,000 pieces of data from the CVE original data set, and each piece of data was cleaned and annotated, and contained one or more triples. In addition, in order to evaluate the effectiveness of the model, the widely used indicators Precision (P), Recall (R) and F1-scores (F1) are used to measure the performance of the model. When the relationship type and the two entities are correct, the extracted triple is considered correct.

B. model training

In order to better train our model, we divide the data set, 90% of which is the training set of the model and 10% is the test set of the model. The data set consists of four positive relation classes and one negative relation class. We add more than 20% noise data into the training set and the test set. These noise data are only labeled with entities and the relationship is labeled with NA. In the test set, we switch the two entities of each sentence to increase the noise data in the test set and make the test conform to the reality.

In our experiment, we pre-trained neural network model and RL model respectively. The number of training rounds of the neural network model is 3, and that of RL model and joint training is 10. The experimental parameters are shown in Table 2. The word embedding and position embedding is 50 and 5 dimensions. The batch size is 128 and the number of extractions is 3. We use dropout algorithm for sentence representation in embedding layer. Bi LSTM layer and attention layer, and the value of dropout is 0.5. In the Bi LSTM layer, the number of hidden units set for Bi LSTM is 128. We set the learning rate to 0.02 and the update rate to 0.01.

TABLE II
The experimental parameters

Parameters	Values
Word embedding dim	50
dropout	0.5
Position embedding dim	5
Bi-LSTM hidden size	128
Batch size	128
Learning rate	0.02
Update rate	0.01
Sentence length	70
Sample times	3
dropout	0.5

C. Experimental results and discussion

It can be seen from Table 3 that the accuracy of the Union-LSTM and origin-LSTM models on the test set is relatively high, but there is not much difference. The recall and F1 values of Union-LSTM are significantly higher than those of origin-LSTM, which shows Our model performs better than the original model, which proves the effectiveness of joint training. The experimental results of origin-LSTM under the conditional random field show that the attention mechanism and joint training can further improve the performance of relationship classification and meet our expectations of the experiment. Therefore, the attention mechanism and reinforcement learning mechanism can better deal with noisy data, and the joint method we proposed is better than most methods.

TABLE III
COMPARISON OF EXPERIMENTAL RESULTS OF EACH GROUP

Method	Precision	Recall	F1
origin-LSTM-CRF	0.797	0.644	0.712
LSTM	0.762	0.622	0.685
Bi-LSTM	0.775	0.637	0.699
origin-LSTM	0.813	0.663	0.73
Union-LSTM	**0.816**	**0.676**	**0.739**

In order to verify that our proposed method can facilitate the relationship classification of multi statement entity pairs, we show the P@N value of different methods in Table 4. Each method is given P@100, P@300, P@500 And its average. We can see that in the P@100 column, the score of the original model is higher than that of the joint model, which indicates that in the early stage of training, the joint model did not get a good training model because of the problems of reinforcement learning and small data scale. With the increase of the amount of data, the accuracy of the model gradually improves and exceeds the original model, which indicates that the performance of the joint model is slightly improving. For the case of the CNN method, the above points can also be verified, and the average accuracy of our proposed model is higher. At the same time, we also conducted a performance comparison experiment between the origin-LSTM-CRF joint model and the Union-LSTM model. The experimental results prove that the model we proposed has a significant improvement on the CVE data set.These results show that the accuracy of the proposed model is very high in different number of test sets, and the average accuracy is 81%. At the same time, reinforcement learning mechanism can improve the noise processing effect of the neural network model.

TABLE IV
P@N COMPARISON BETWEEN DIFFERENT METHODS

P@N(%)	100	300	500	Mean
origin-LSTM-CRF	78.2	78.8	79.5	78.8
origin-CNN	78.9	79.6	80.1	79.5
Union-CNN	79.7	80.4	81.6	80.6
origin-LSTM	**81**	80.5	80.8	80.7
Union-LSTM	80	**81**	**82**	**81**

In order to verify the effectiveness of the joint training of the neural network model and reinforcement learning, we have carried out experiments with original neural network model and union model respectively. From Figure 4, we can see that combining RL mechanism can improve the performance of the original neural network model. For this model, when the recall rate is less than 0.05, the performance of the combined model is worse than that of the original model. The reason for this situation is that there are few training data in the initial stage of training, the strategy network of reinforcement learning is not perfect, and the ability to deal with noise is not strong; When the recall rate is greater than 0.05, the performance of the model using reinforcement learning mechanism is better than the original model. Therefore, the noise data removal method based on reinforcement learning is effective. And the collaborative learning of the two modules can reduce the noise of the data set and effectively improve the classification performance. And it can be obviously seen in the figure that the accuracy of the model in processing our data set is generally higher than 80%.

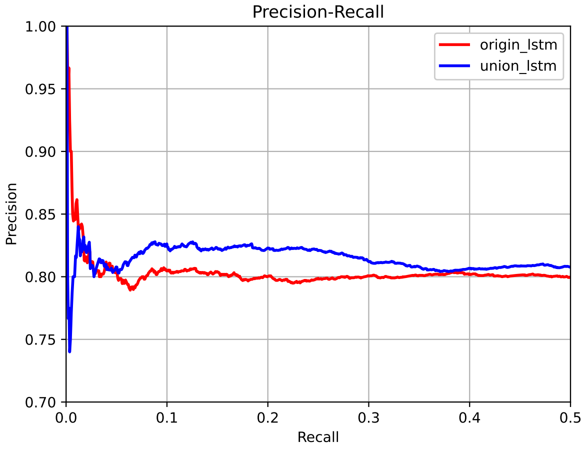


Fig. 4. Performance comparison of joint network and origin network

Vulnerability security knowledge graph has not only achieved corresponding results in the prediction of the hidden relationship of vulnerabilities and the prediction of breaches of vulnerabilities, but also can provide strong support for network security analysis. At the same time, knowledge graphs also play an important role in assisting threat detection. Therefore, in this paper, entities and relationships are graphed, and the Neo4j graph database is used to construct a knowledge graph to further enrich the network security knowledge graph. We use the extracted entity pairs and the relationship predicted by the model to form a triple, which is inserted into neo4j to improve the vulnerability knowledge graph. We select two CVE items from the test set for comparison, as shown in Table 5. And the knowledge graph composed of these two pieces of data is drawn. As shown in Figure 5, the experimental results show that we can transform each data into a knowledge graph. For the relationships contained in the data, only when the probability of model prediction is greater than 60%, we choose to add relationships of the knowledge graph. Otherwise we do not. By comparing with the data in the table, we can see that the relationship predicted by the model is the same as the actual relationship. Although some relationships are not predicted, such as the relationship between the attack user entity ’remote attackers’ and its CVE ID, it does not affect the construction of the overall vulnerability security knowledge graph. Therefore, the above experiments can prove that the proposed model is feasible in extracting the existing relationships in CVE data, and the performance of relationship classification and extraction is better.

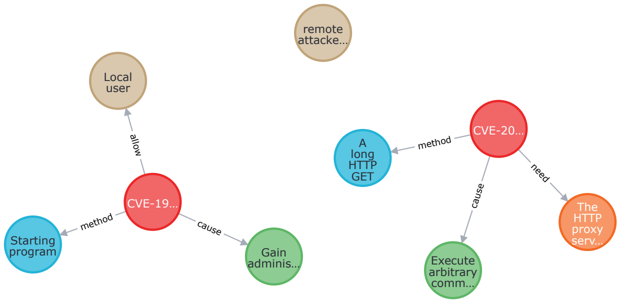


Fig. 5. Knowledge graph composed of two data

TABLE V
EXAMPLES OF CVE ITEMS

CVE Item	Entity	Relation	Probability
CVE-1999-1414	Local user	2	0.831
	Starting program	3	0.847
	Gain administrator privilege	4	0.82
CVE-2000-0376	The HTTP proxy server	1	0.781
	remote attackers	2	0
	A long HTTP GET request	3	0.829
	Execute arbitrary command	4	0.811

V. Conclusion and Future work

This paper proposes a framework to deal with the CVE description, which is composed of entity relation annotation algorithm and relation classification model. In this framework, for each new CVE to describe data, we first use entity annotation algorithm to annotate the entities in the sentence, then we can mark the relationship in the data through the trained model, then transform the entity and relationship into three tuples, and build the knowledge graph of vulnerability. And with the increase of the amount of data, the model can update its own parameters, making the relationship classification better. A large number of experiments show that our model can better filter the noisy CVE description sentences, and can better classify the new CVE data at the sentence level.

In the future work, we will study how to improve and enrich the vulnerability security knowledge graph. At the same time, we will use graph neural network to analyze the knowledge graph to find the hidden attack sources, attack methods and attack paths in the knowledge graph, so as to provide auxiliary decision-making for the scientific defense of Cyberspace.

Acknowledgment

The authors are very grateful to three anonymous referees for their valuable comments on the paper, which have considerably improved the presentation of this paper.

References

[1] Scharei K, Heidecker F, Bieshaar M. Knowledge Representations in Technical Systems–A Taxonomy[J]. arXiv preprint arXiv:2001.04835, 2020.

[2] Razzaq A, Anwar Z, Ahmad H F, et al. Ontology for attack detection: An intelligent approach to web application security[J]. computers & security, 2014, 45: 124-146.

[3] Petnga L, Austin M. An ontological framework for knowledge modeling and decision support in cyber-physical systems[J]. Advanced Engineering Informatics, 2016, 30(1): 77-94.

[4] Camastra F, Ciaramella A, Maratea A, et al. Semantic maps for knowledge management of web and social information[M]//Computational Intelligence for Semantic Knowledge Management. Springer, Cham, 2020: 39-51.

[5] Iannacone M, Bohn S, Nakamura G, et al. Developing an ontology for cyber security knowledge graphs[C]//Proceedings of the 10th Annual Cyber and Information Security Research Conference. 2015: 1-4.

[6] Kiesling E, Ekelhart A, Kurniawan K, et al. The SEPSES knowledge graph: an integrated resource for cybersecurity[C]//International Semantic Web Conference. Springer, Cham, 2019: 198-214.

[7] Li J, Sun A, Han J, et al. A survey on deep learning for named entity recognition[J]. IEEE Transactions on Knowledge and Data Engineering, 2020.

[8] Fisher J, Vlachos A. Merge and label: A novel neural network architecture for nested NER[J]. arXiv preprint arXiv:1907.00464, 2019.

[9] Mintz M, Bills S, Snow R, et al. Distant supervision for relation extraction without labeled data[C]//Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. 2009: 1003-1011.

[10] Zeng X, He S, Liu K, et al. Large scaled relation extraction with reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).

[11] G. Ji, K. Liu, S. He, and J. Zhao, “Distant supervision for relation extraction with sentence-level attention and entity descriptions,” in Proc. AAAI,2017, pp. 3060–3066.

[12] Feng J, Huang M, Zhao L, et al. Reinforcement learning for relation classification from noisy data[J]. arXiv preprint arXiv:1808.08013, 2018.

[13] Wu Y, Bamman D, Russell S. Adversarial training for relation extraction[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017: 1778-1783.

[14] Han X, Zhu H, Yu P, et al. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation[J]. arXiv preprint arXiv:1810.10147, 2018.

[15] Fu T J, Li P H, Ma W Y. GraphRel: Modeling text as relational graphs for joint entity and relation extraction[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 1409-1418.

[16] Zheng S, Hao Y, Lu D, et al. Joint entity and relation extraction based on a hybrid neural network[J]. Neurocomputing, 2017, 257: 59-66.

[17] Zheng S, Wang F, Bao H, et al. Joint extraction of entities and relations based on a novel tagging scheme[J]. arXiv preprint arXiv:1706.05075, 2017.

[18] Bekoulis G, Deleu J, Demeester T, et al. Joint entity recognition and relation extraction as a multi-head selection problem[J]. Expert Systems with Applications, 2018, 114: 34-45.

[19] Sahu S K, Christopoulou F, Miwa M, et al. Inter-sentence relation extraction with document-level graph convolutional neural network[J]. arXiv preprint arXiv:1906.04684, 2019.

[20] Christopoulou F, Miwa M, Ananiadou S. A walk-based model on entity graphs for relation extraction[J]. arXiv preprint arXiv:1902.07023, 2019.

[21] Loper E, Bird S. Nltk: The natural language toolkit[J]. arXiv preprint cs/0205028, 2002.