CyberRel: Joint Entity and Relation Extraction for Cybersecurity Concepts

Yongyan Guo, Zhengyu Liu, Cheng Huang(✉), Jiayong Liu, Wangyuan Jing, Ziwang Wang, and Yanghao Wang

School of Cyber Science and Engineering, Sichuan University, Chengdu, China
codesec@scu.edu.cn

Abstract. Cyber threats are becoming increasingly sophisticated, while new attack techniques are emerging, causing serious harm to businesses and even countries. Therefore, how to analyze attack incidents and trace the attack groups behind them becomes extremely important. Threat intelligence provides a new technical solution for attack traceability by constructing Cybersecurity Knowledge Graph (CKG). The CKG cannot be constructed without a large number of entity-relation triples, and the existing entity and relation extraction for cybersecurity concepts uses the traditional pipeline model that suffers from error propagation and ignores the connection between the two subtasks. To solve the above problem, we propose CyberRel, a joint entity and relation extraction model for cybersecurity concepts. We model the joint extraction problem as a multiple sequence labeling problem, generating separate label sequences for different relations containing information about the involved entities and the subject and object of that relation. CyberRel introduces the latest pre-trained model BERT to generate word vectors, then uses BiGRU neural network and the attention mechanism to extract features, and finally decodes them by BiGRU combined with CRF. Experimental results on Open Source Intelligence (OSINT) data show that the F1 value of CyberRel is 80.98%, which is better than the previous pipeline model.

Keywords: Relation extraction · Joint model · Threat intelligence · Knowledge graph

1 Introduction

Nowadays, the damage and impact caused by malicious behavior in cyberspace such as hacker attacks, frauds, and rumors have become more serious. Therefore, how to effectively and accurately detect cyber attacks as early as possible, analyze attack incidents, and trace the source of attackers and groups has become a severe problem for enterprises and countries.

The concept of Cyber Threat Intelligence (CTI) was developed supplying new theoretic support for cyber-attack source tracing, making it possible to trace the source of a wide range of attacks. Therefore, many researchers extract and analyze different threat intelligence to generate the Cybersecurity Knowledge Graph

D. Gao et al. (Eds.): ICICS 2021, LNCS 12918, pp. 447–463, 2021.
https://doi.org/10.1007/978-3-030-86890-1_25

(CKG). The CKG has the characteristic of strong timeliness and high accuracy, which can timely and easily detect, respond and defend against specific targets providing a new measure for attack source tracking, and can even effectively deal with sophisticated cyberattacks (e.g., zero-day attacks, advanced persistent threat).

The key step in constructing CKG is cyber threat intelligence information extraction, which involves subtasks such as entity recognition, relation extraction, and event extraction. Currently, many research groups have conducted research on the automated construction and analysis of CKG [3–9]. In terms of CTI information extraction, previous studies are dedicated to extracting cybersecurity concepts [10–12] and entities [13–15] from unstructured data.

However, the construction of CKG is inseparable from a large number of cybersecurity entity-relation triples. The CKG consists of a number of nodes and edges, where the nodes represent entities and the edges represent the relations between entities. Because that information comes from a large scale of unstructured data through various sources like system logs, vulnerability databases, cybersecurity reports, hacker forums, and social media, it has the characteristics of multisource, heterogeneous, polysemy, and highly dependent on domain knowledge. Therefore, relation extraction of cybersecurity is still a great challenge. Existing researches on cybersecurity relation extraction [16,17] uses the traditional pipeline model, named entity recognition first and then relation extraction, which leads to error propagation and losses sight of the relevance between entity recognition and relation extraction.

To solve the above problem, we propose CyberRel, a joint entity and relation extraction model for cybersecurity concepts, which extracts both cybersecurity entities and relations and generates the semantic triples. Specifically, we use a tagging scheme to convert the joint extraction problem into a multiple sequence labeling problem by generating separate label sequences for different relations containing information about the related entities and the subject and object of that relation. CyberRel applies the pre-trained model, BERT, to generate word vectors. After extracting semantic features by BiGRU, the model assigns higher weights to relation-related words in the sentences by an attention mechanism. Finally, BiGRU combined with CRF is used to decode and construct cybersecurity triples.

In summary, the main contribution of this paper are as follows:

- We propose a joint entity and relation extraction model for cybersecurity concepts, named CyberRel. The model employs deep learning techniques to extract entities and relations in sentences simultaneously, avoiding the error propagation of traditional pipeline models.
- We model the joint extraction problem as a multiple sequence labeling problem by generating separate label sequences for different relations. Each label sequence contains information about the related entities and the subject and object of that relation. This method can effectively solve the entity overlapping problem commonly found in the cybersecurity corpus.

– To validate the effectiveness of CyberRel, we collected and manually labeled OSINT data including vulnerability databases, security bulletins, and APT reports. The experimental results show that CyberRel outperforms the traditional pipeline model with an F1 value of 80.98%.

The rest of the paper is organized as follows: Sect. 2 discusses related work, and Sect. 3 presents the details of the joint entity and relation extraction model for cybersecurity concepts (CyberRel) which we proposed in this paper. Section 4 provides the experiments and analysis related to this work. Section 5 summarizes conclusion and proposes future works.

2 Related Work

In this section, we first review the methods for automated construction and analysis of CKG. Secondly, since the pivotal step of CKG construction is threat intelligence information extraction, we review the work related to CTI extraction including entity recognition, relation extraction, and event extraction subtasks. Finally, we present the related research on relation extraction.

2.1 Cybersecurity Knowledge Graph

The Knowledge Graph (KG) was originally proposed by Google. It is a knowledge base that integrates information from multiple sources, links real-world entities or concepts, and provides search services through semantic retrieval. In the field of cybersecurity, correlating and fusing threat intelligence data from different sources to generate the CKG can provide new technical means for situational awareness and attack traceability.

Building a CKG first requires abstracting a myriad of concepts and complex relations in the cybersecurity domain into a semantic network. Iannacone et al. [1] proposed STUCCO, an ontology for building CKGs, integrating 13 different formats of cybersecurity data sources. Building on this foundation, Syed et al. [2] proposed a Unified Cybersecurity Ontology (UCO). The UCO ontology provides a general understanding of the cybersecurity domain and, in addition to mapping to STIX, UCO extends several related cybersecurity standards, vocabularies, and ontologies such as CVE, CCE, CVSS, CAPEC, CYBOX, KillChain, and STUCCO.

In the area of automated construction and analysis of CKG, researchers have also proposed several ideas and approaches in recent years [3–9]. Jia et al. [3] introduced a cybersecurity knowledge base and deduction rules based on a quintuple model. Gao et al. [4] proposed EFFHUNTER, a system that facilitates threat hunting in computer systems using OSINT. The system uses an unsupervised, lightweight, and accurate NLP pipeline to extract structured threat behaviors from unstructured OSINT text. Piplai et al. [5] described a system that extracts information from After Action Reports (AARs) and represents the extracted information in a CKG. Zhao et al. [6] demonstrated a threat intelligence framework (HINTI). HINTI first recognizes IOCs and models the interdependent relations between IOCs using heterogeneous information networks

(HINs), and then proposes a threat intelligence computing framework based on graph convolutional networks to explore complex security knowledge. Although these approaches have made initial attempts and achieved good results in CKG construction, further research is needed in the key steps of knowledge graph construction.

2.2 Threat Intelligence Information Extraction

The construction of a knowledge graph can be divided into three steps, including information extraction, knowledge fusion, and knowledge reasoning. Among them, information extraction plays a decisive role in the quality of the generated knowledge graph. Information extraction for threat intelligence is divided into several subtasks, including entity recognition [10–15], relation extraction [16,17] and event extraction [18].

In terms of cybersecurity entity and concept recognition, Mittal et al. [10] proposed a framework for extracting threat intelligence from Twitter, CyberTwitter, which automates the extraction of security vulnerability concepts. Liao et al. [11] introduced iACE for automatically extracting IOCs and their context in the sentences of technical articles. Zhu et al. [12] designed Chainsmith, an IOC extraction system that collects IOCs from security articles and classifies them according to the stages of the Kill Chain. Ghazi et al. [13] used natural language processing to extract threat sources from unstructured web threat information sources and provided comprehensive threat reports in the STIX standard.

Due to the lack of a well-labeled corpus for training, relatively few studies have been conducted on cybersecurity relation extraction and event extraction compared to entity recognition. Pingle et al. [16] proposed RelExt, a deep learning-based cybersecurity relation extraction method for constructing CKGs. The model uses a pipeline approach, first identifying entities in the text by an entity recognizer then classifying the relations by a deep learning model. Jones et al. [17] implemented a semi-supervised cybersecurity relation extraction method based on a bootstrapping algorithm to extract relations. Satyapanich et al. [18] proposed CASIE, a security event extraction system that uses deep neural networks and can incorporate rich linguistic features and word embeddings for extracting security events related to cyber-attacks and vulnerabilities.

2.3 Relation Extraction

As a subtask of information extraction, relation extraction has a long research history. The main approaches to relation extraction can be broadly divided into three categories, including early rule-based approaches [19,20]; traditional machine learning-based approaches [21,22]; and deep learning-based approaches [23–27]. In recent years, the latest research results in the field of relation extraction have focused on deep learning models [28–31]. The advantage of deep learning methods is that they do not require manual extraction of features nor a large amount of domain knowledge.

Currently, there are two main approaches to relation extraction based on deep learning: the pipeline approach and the joint approach. The pipeline approach performs relation classification after extracting all the entities. Zeng et al. [23] first applied CNN to relation extraction to automatically extract lexical and sentence-level features. Wei et al. [24] proposed a novel cascaded binary annotation framework (CASREL) that models relations as functions that map subjects to objects in a sentence, which naturally handles the overlapping triple problems. Although these methods achieve promising results, the pipeline architectures suffer from the problem of error propagation. In addition, neglecting the relationship between the two tasks of entity recognition and relation extraction for training can also affect the effectiveness of relation extraction. Therefore, to construct the bridge between the two subtasks, building a joint model that extracts entities together with relations simultaneously has attracted much attention. Miwa et al. [25] proposed a joint relation extraction model based on shared parameters, which captures both word sequences and dependency tree substructure information for end-to-end relation extraction via LSTM. Bekoulis et al. [26] propose a joint model that uses a CRF layer to model the entity recognition task and the relation extraction task as a multi-headed selection problem. Zheng et al. [27] proposed a new tagging scheme that can convert the joint extraction task to a sequence labeling problem. Yuan et al. [30] proposed a relation-based attention network (RSAN) to jointly extract entities and relations using a relation-aware attention mechanism.

In the construction of CKG, a lot of research has been conducted on the extraction of cybersecurity entities and concepts, while research on cybersecurity relation extraction is still in its infancy. Existing approaches use traditional pipeline methods, which leads to error propagation and loses sight of the relevance between entity recognition and relation extraction. Different from these above works, this paper proposes a joint entity and relation extraction model for cybersecurity concepts, which extracts entities and relations simultaneously, effectively avoiding the shortcomings of the traditional pipeline model.

3 Methodology

In this section, we introduce CyberRel, a joint entity and relation extraction model for cybersecurity concepts. We briefly outline the overall strategy here before discussing details in the following subsections. The overall architecture of CyberRel is shown in Fig. 1. CyberRel takes threat intelligence data collected from multiple sources as raw input. Then the data undergoes a pre-processing process including data cleaning, sentence segmentation, and tokenization to obtain the training corpus, which will be fed into the joint extraction model subsequently (see Sect. 3.1 for details). We adopt the cybersecurity entities and relations defined in the UCO 2.0 [2] ontology and model the joint entity and relation extraction problem as a multiple sequence labeling problem by generating a sequence of labels for each relation through a specific tagging schema (see Sect. 3.2 for details). Each relation label sequence contains information about the

entities involved and the subject and object of the relation. Our proposed multiple sequence labeling model is structured into an embedding layer, an encoding layer, an attention layer, and a decoding layer (see Sect. 3.3 for details). Finally, CyberRel constructs cybersecurity triples based on the label sequences predicted by the model, and these triples will eventually be used to construct CKGs.

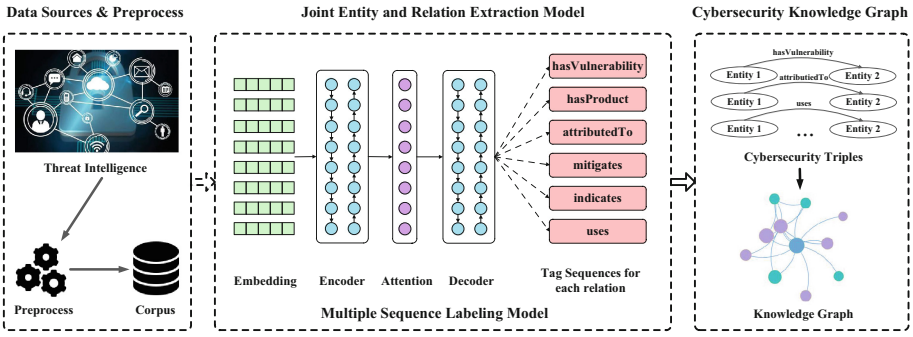

Fig. 1. CyberRel architecture.

3.1 Data Preprocess

CyberRel can extract cybersecurity triples from massive amounts of heterogeneous threat intelligence data. Threat intelligence data can be sourced from vulnerability databases, security bulletins, APT reports, security or technology blogs, hacking forums. This data is usually stored in rich text documents such as PDF, HTML/XML, JSON, and other formats. First, we use various text parsing tools (e.g. HTMLParser, PDFLib) to extract the raw text from these documents. But the extracted raw text is not well-formatted. Therefore, we devised some data pre-processing procedures as follows.

The first step in preprocessing is data cleaning, where we remove non-ASCII characters from the text and whitespace characters at the beginning and end of each sentence. It is worth noting that in some threat intelligence data, special types of entities are often rewritten to prevent readers from clicking on them by mistake. For example, the IP address "136.244.119.85" is rewritten as "136. 244.119[.]85"; the URL "http://www.test.com" is rewritten to "http://www.test[.]com"; the email address "hacker@test.com" is rewritten as "hacker[at]test.com". We revert this rewritten form to its original form.

The next step in preprocessing is special entity substitution. In the field of cybersecurity, some entities are very different in form from the normal natural language, such as IP, MAC, Hash, URL, Email, domain name, file name, and file path. We build regular expressions to match these entities from text and replace them with natural language strings in the form of "sub type", where "type" is the type of the special entity. For example, we would replace the IP address "136.244.119.85" with "sub ip".

The last step in the preprocessing process is text segmentation, which is the process of converting text into sequences. We use the NLTK library for sentence segmentation and WordPiece for word tokenization.

3.2 Tagging Scheme

In this section, we will introduce the tagging schema for the joint entity and relation extraction. The entities and relations applied by CyberRel are derived from UCO 2.0 [2], which provides a general understanding of the cybersecurity domain.

- The main entity types in UCO 2.0 include: Indicator, Threat Actor, Attack Pattern, Malware, Tool, Campaign, Course of Action, Vulnerability.
- The main relation types in UCO 2.0 include: hasProduct, hasVulnerability, uses, attributedTo, mitigates, indicates.

In the field of relation extraction, there has been related work [27,30,31] on the joint entity and relation extraction through the construction of a specific tagging schema. For cybersecurity concepts, the extracted relation usually suffers from the entity overlapping problem that different types of relations sharing the same entities, so the tagging scheme has to overcome this issue. CyberRel generates a sequence of labels for each relation in UCO 2.0. In each tag sequence, we use the typical "BIO" signs to locate the entities in the sentence, where "B" represents the starting part of the entity, "I" represents the middle part of the entity, and "O" is the non-entity part. At the same time, we also label the entity as subject or object in the relation, with "1" representing the subject in the triple and "2" representing the object in the triple.

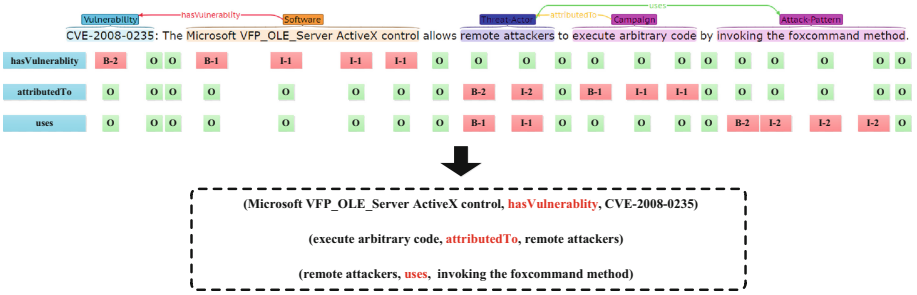

Fig. 2. An example for tagging scheme.

Figure 2 shows an example of our tagging scheme. The first label sequence describes the "*hasVulnerablity*" relation, where "*Microsoft VFP_OLE _Server ActiveX control*" is an entity of type "*Software*", as the subject of the "*hasVulnerablity*" relation; "*CVE-2008-0235*" is an entity of type "*Vulnerability*", as the object of the "*hasVulnerablity*" relation. Through the label sequence, we can generate triple ("*Microsoft VFP_OLE_Server ActiveX control*", "*hasVulnerability*", "*CVE-2008-0235*"). Likewise, other label sequences can be used to

generate triples of corresponding relations. If a relation does not exist in a sentence, the label sequence for that relation will be all "O". Besides, as we can see, the "*attributedTo*" and "*uses*" relations have the over-lapped entity "*remote attackers*", and they can be extracted without conflict based on the separate label sequences.

3.3 Multiple Sequence Labeling Model

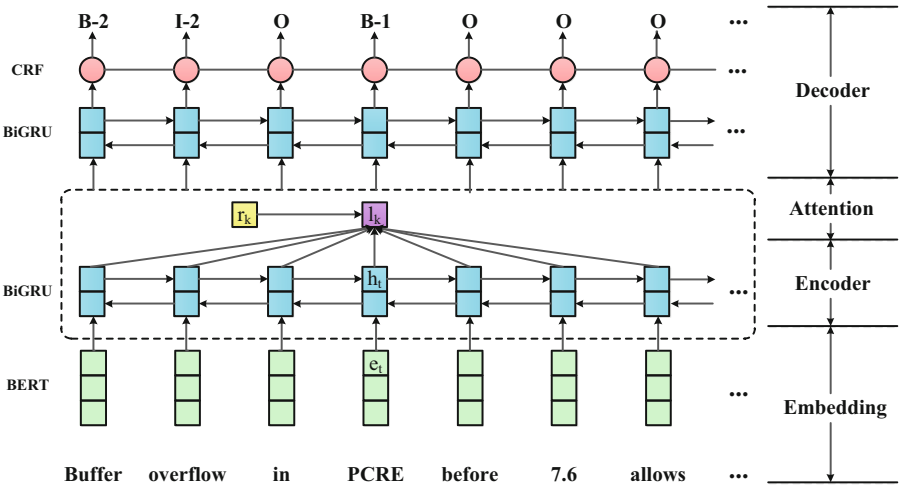

Fig. 3. The multiple sequence labeling model for joint entity and relation extraction. It receives the same sentence input and different relation r_k to extract all triples in the sentence. e_t is the BERT embedding of the word, h_t is the hidden vector of time step t, r_k is the trainable embedding of the k-th relation, l_k is the attention weights under relation type r_k. Under the given relation r_k (Take *hasVulnerability* for example), the decoder extracts the corresponding entities of r_k to generate triples (*PCRE, hasVulnerability, Buffer overflow*).

Based on the tagging scheme above, we propose an end-to-end multiple sequence labeling model to jointly extract cybersecurity entities and relations. We take the sentence and a type of relation as input to the model, and the output sequence holds information about the subject and object entities involved in that relation. Thus, for a sentence, when we traverse all the relation types, the model generates a label sequence for each type of relation, resulting in a joint extraction of entities and relations. Figure 3 gives an overall structure of the model, which is divided into four parts. The embedding layer generates a word vector e_t for each word x_t in sentence X. In the encoding layer, the embedding sentence is fed into the bi-directional Gated Recurrent Units (BiGRU) neural network to generate a hidden state representation h_t. Then we apply the attention mechanism to assign different weights to the context words under different relations and constructs a

relation-specific sentence representation l_k. Finally, in the decoding layer, we use another BiGRU neural network and joined it with CRF for decoding to obtain the label sequence and extract corresponding entities under the specific relation.

Embedding. Given a sentence as a sequence of tokens, the word embedding layer is responsible to map each token to a word vector. In this paper, we propose to use a pre-trained model to generate word vectors. The pre-trained word embedding model converts words in natural language into dense vectors, and semantically similar words will have similar vector representations. The latest pre-trained model BERT [35] can solve the problem of polysemy, generating different word vectors for the same word according to the context, which can better express the semantic features of the words. This situation often occurs in the cybersecurity corpus. For a piece of software, when describing the vulnerabilities that exist in that software, this entity should then be recognized as a "*Software*" type, and the triple ("*Software*", "*hasVulnerability*", "*Vulnerability*") can be extracted. In another context, the software may be exploited as a tool by an attacker, at which point the entity should be recognized as a "*Tool*" type, and the triple ("*Threat Actor*", "*uses*", "*Tool*") can be extracted. So, we use the BERT model to generate word embedding vectors in the embedding layer. For the input sentence $X = \{x_1, x_2, x_3, ..., x_n\}$, where x_t is the t-th word in the sentence. After the computation of the BERT pre-trained model, the word embedding vector $E = \{e_1, e_2, e_3, ..., e_n\}$ of the sentence is generated, where e_t is the word vector of the t-th word in the sentence.

Encoder. Compared with the traditional recurrent neural network (RNN), GRU consists of an update gate and a reset gate, which can alleviate the gradient disappearance or explosion problem that occurs during training. The GRU hidden state h_t is generated by the previous hidden state h_{t-1} and the input e_t of the current state together. The GRU only calculates the correlation between time step t and the previous time step. However, in the cybersecurity corpus, entities may constitute relations with the entities before or after. So, for the word vectors generated by the embedding layer, we further extract the semantic features of the sentences $H = \{h_1, h_2, h_3, ..., h_n\}$ using BiGRU and then concatenate the forward and backward GRU hidden states as the contextual word representation. The transformations are as follows:

$$h_t = \left[\overrightarrow{GRU}(e_t), \overleftarrow{GRU}(e_t)\right] \tag{1}$$

Attention Mechanism. In the cybersecurity corpus, a sentence usually contains many entities and complex relations. As shown in Fig. 2, the sentence contains five different entities ("*Vulnerability*", "*Software*", "*Threat Actor*", "*Campaign*", "*Attack Pattern*") and three different relations ("*hasVulnerability*", "*attributedTo*", "*uses*"). Therefore, it is necessary to assign different weights to the words in a sentence according to different types of relations. For example,

for the "*hasVulnerability*" relation, the words in the sentence indicating a software name or identify a specific vulnerability should be paid higher attention. Thus, we have referred to the relation-based attention mechanism proposed by Yuan et al. [30]. The attention mechanism can assign different weights to the words in a sentence under each relation, and the attention score can be calculated as follows:

$$h_g = avg\{h_1, h_2, h_3, \ldots, h_n\} \tag{2}$$

$$e_{tk} = v^T \tanh\left(W_r r_k + W_g h_g + W_h h_t\right) \tag{3}$$

$$a_{tk} = \frac{\exp\left(e_{tk}\right)}{\sum_{j=1}^{n} \exp\left(e_{jk}\right)} \tag{4}$$

where h_g indicates the global representation of the sentence, r_k is the embedding of the k-th relation. v, W_r, W_g, and W_h are all trainable parameters. The attention score generated reflects the importance of the sentence's words in the context as well as relational expression in the current relation. The sentence representation l_k under the r_k relation is generated by the weighted sum of the sentence words, which is calculated as shown in Eq. 5. The attention layer combines the generated l_k and the sentence representations output by the encoding layer as input to the decoding layer, as shown in Eq. 6.

$$l_k = \sum_{t=1}^{n} a_{tk} h_t \tag{5}$$

$$h_t^k = h_t \oplus l_k \tag{6}$$

Decoder. The decoding layer generates the label sequences of the sentences under the r_k relation and returns the relational triples through the tagging scheme described in Sect. 3.2. We first used another BiGRU to produce sentence representations $H^o = \{h_1^o, h_2^o, h_3^o, ..., h_n^o\}$ and generate sequence scores $Z = \{z_1, z_2, z_3, ..., z_n\}$using features from the encoding and attention layers. The calculation process is as follows, where W is the parameter:

$$h_t^o = \left[\overrightarrow{GRU}(h_t^k), \overleftarrow{GRU}(h_t^k)\right] \tag{7}$$

$$z_t = W h_t^o \tag{8}$$

Next, the sequence is decoded by the CRF layer, which is able to obtain constrained rules from the training data, to ensure that the predicted cybersecurity entity labels are valid. The decoding process is shown as follows:

$$score(Z, y) = \sum_{t=0}^{n} A_{y_t, y_{t+1}} + \sum_{t=1}^{n} Z_{t, y_t} \tag{9}$$

$$p(y \mid Z) = \frac{\exp(score(Z, y))}{\sum_{y' \in Y_Z} \exp\left(score\left(Z, y'\right)\right)} \tag{10}$$

$$y^* = \arg\max_{y \in Y_Z} score(Z, y) \tag{11}$$

where A is the transition matrix between labels, $score(Z, y)$ is the position score, and $p(y \mid Z)$ is the normalized probability function. Finally, the label sequence y^* is generated.

4 Experiments

4.1 Datasets

The datasets used in this paper are collected from publicly available OSINT data, including the CVE vulnerability database, security bulletins, and Advanced Persistent Threats (APT) reports. To train the CyberRel model, we invited five graduate students majoring in cybersecurity to annotate the dataset, using the BRAT annotation platform [34]. In total, we annotated 13,262 sentences containing 75,990 triples.

- **CVE vulnerability database:** CVE is the Common Vulnerabilities and Exposures, a list of various computer security vulnerabilities that have been publicly disclosed. The CVE Automation Working Group is piloting the use of git to share information about public vulnerabilities [32].
- **Security bulletins:** Many vendors (e.g. Microsoft, Adobe, Oracle, Vmware) regularly publish security bulletins that are intended to disclose security vulnerabilities in their software, describe remedies, and provide applicable updates for the affected software.
- **APT reports:** APT reports are publicly available papers and blogs related to malicious activities and associated with APT organizations or toolsets [33].

4.2 Evaluation Metrics

We use standard Precision (P), Recall (R), and F1-score to measure the performance of CyberRel. A triple is considered to be correctly extracted if and only if its relation type and both entities are correctly matched.

4.3 Experimental Settings

To evaluate the effectiveness of CyberRel, we design a set of experiments. Since the previous work used the traditional pipeline model, we compare CyberRel (joint model) with the previous work [16] (pipeline model) in the main experiment. As CyberRel is built with the word embedding model and neural networks, we designed two comparison experiments to analyze the effects of different word embedding models and different neural networks on the performance of CyberRel.

We use StratifiedKFold to create train/test splits and set k = 5. The size of the BERT embedding vector is 768 dimensions. The size of the BiGRU hidden layer and relational embedding vector are both set to 300. We choose RMSprop as our model optimizer, the learning rate is 0.0001, and the batch size is 64. We use the dropout mechanism to avoid overfitting with a rate of 0.5.

4.4 Experimental Result

Main Results. The CyberRel proposed in this paper is a joint entity and relation extraction model, so we compare it with the existing pipeline approach, RelExt [16]. From Table 1, we can see that CyberRel outperforms RelExt, significantly improving precision (83.00%), recall (79.09%) and F1-score (80.98%). This indicates that the joint model extracts both entities and relations, which avoids the error propagation between the two subtasks of the pipeline model, and effectively improves the performance of entity-relation triples extraction.

Table 1. Main results of the compared models.

Models	Precision	Recall	F1-score
RelExt [16]	57.48%	63.90%	60.52%
CyberRel	**83.00%**	**79.09%**	**80.98%**

Effect of Different Word Embeddings. As the word vectors generated by the word embedding model serve as the input to the downstream model, the quality of the word vectors has an important impact on the model performance. In this section, we experiment with two representative word embedding models, BERT [35] and Word2Vec [36], where the BERT model is the "cased_L-12_H-768_A-12" version, and the Word2Vec model is trained by Youngja et al. [36] through cybersecurity corpus. It can be seen from Table 2 that using BERT for word embedding has a certain improvement compared to Word2Vec. The F1 value is improved by 13.10% when GRU is used and by 14.03% when LSTM is used. This is attributed to the fact that BERT can generate different word vectors for the same word depending on the context thus making better use of the contextual information of the text, while Word2Vec can only generate a fixed word vector representation for each word.

Table 2. Results for different word embeddings and different neural networks in CyberRel.

Methods		Precision	Recall	F1-score
Embedding	Neural Network			
Word2Vec	LSTM	70.84%	62.55%	66.40%
	GRU	73.91%	62.81%	67.88%
BERT	LSTM	82.40%	78.63%	80.43%
	GRU	**83.00%**	**79.09%**	**80.98%**

Effect of Different Neural Networks. Since a neural network is used in our model for the sequence labeling task, we investigated the effect of different neural networks on the model performance, specifically, we experimented with the performance of LSTM and GRU neural networks, respectively. As shown in Table 2,

when using the same word embedding model, such as BERT, GRU (Precision: 83.00%, Recall: 79.09%, F1-score: 80.98%) performs slightly better than LSTM (Precision: 82.40%, Recall: 78.63%, F1-score: 80.43%). The experimental results show that GRU is more suitable for the problem of the joint entity and relation extraction for cybersecurity concepts. So we take BiGRU neural network in CyberRel.

4.5 Case Study

In this section, we illustrate the advantages of the joint model over the pipeline model by two examples, as shown in Appendix Table 3. In both examples, our proposed joint model predicts all the triples in the sentences correctly.

For Case 1, although the pipeline model correctly predicts all the "*Software*" entities in the entity recognition task, in the relation prediction between the three "*Software*" entities, the model predicts the three entities in two-by-two combinations and comes up with the wrong relations ("*mitigates*"). This indicates that the pipeline model does not take into account the connection between entity recognition and relation extraction tasks, while the joint model is able to predict the two "*hasProduct*" triples between the three "*Software*" entities well.

For Case 2, the pipeline model only recognizes the "*patches/Course-of-Action*" entity but not the "*CVE-2008-3138/Vulnerability*" entity, resulting in a null input to the relation extraction model that fails to predict the relation between them. This indicates that the pipeline model has the defect of error propagation, implying that if an entity is not predicted or is incorrectly predicted, it will affect the subsequent relation extraction task.

5 Conclusion

In this paper, we propose CyberRel, a joint entity and relation extraction model for cybersecurity concepts, which can extract both entities and relations in the cybersecurity corpus. Specifically, we use an tagging scheme to convert the joint extraction problem into a multiple sequence labeling problem by generating separate label sequences for different relations containing information about the related entities and the subject and object of that relation. In addition, CyberRel employs BERT model, BiGRU neural network, and attention mechanism to extract the features of sentences and generate label sequences under different relations. In the experimental part, our results on OSINT data demonstrate that CyberRel achieves better results compared to the traditional pipeline approach. To further improve the quality of CKG generation, our future research work will focus on document-level relation extraction and cybersecurity entity disambiguation.

Acknowledgment. This research is funded by the National Natural Science Foundation of China (No. 61902265), Sichuan Science and Technology Program (No. 2020YFG0047, No. 2020YFG0374).

A Appendix

Table 3. The examples of the triples to the given sentences extracted by joint model and pipeline model.

#Case 1	
Raw text	CVE-2008-3138: The (1) PANA and (2) KISMET dissectors in Wireshark (formerly Ethereal) 0.99.3 through 1.0.0 allow remote attackers to cause a denial of service (application stop) via unknown vectors.
Joint model	(PANA/Software, hasVulnerability, CVE-2008-3138/Vulnerability)
	(KISMET/Software, hasVulnerability, CVE-2008-3138/Vulnerability)
	(Wireshark/Software, hasProduct, PANA/Software)
	(Wireshark/Software, hasProduct, KISMET/Software)
	(remote attackers/Threat-Actor, uses, unknown vectors/Attack-Pattern)
	(denial of service/Campaign, attributedTo, remote attacker/Threat-Actor)
	(denial of service/Campaign, attributedTo, unknown vectors/Attack-Pattern)
Pipeline model	(PANA/Software, mitigates, Wireshark/Software)
	(KISMET/Software, mitigates, Wireshark/Software)
	(PANA/Software, mitigates, KISMET/Software)
	(KISMET/Software, mitigates, PANA/Software)
	(Wireshark/Software, mitigates, PANA/Software)
	(Wireshark/Software, mitigates, KISMET/Software)
	(remote attackers/Threat-Actor, uses, unknown vectors/Attack-Pattern)
	(denial of service/Campaign, attributedTo, remote attackers/Threat-Actor)
	(denial of service/Campaign, attributedTo, unknown vectors/Attack-Pattern)
#Case2	
Raw text	To remediate CVE-2020-3956 apply the patches listed in the 'Fixed Version' column of the 'Response Matrix' found below.
Joint model	(patches/Course-of-Action, mitigates, CVE-2008-3138/Vulnerability)
Pipeline model	Only "patches/Course-of-Action" found

References

1. Iannacone, M., et al.: Developing an ontology for cyber security knowledge graphs. In: Proceedings of the 10th Annual Cyber and Information Security Research Conference, pp. 1–4 (2015)
2. Syed, Z., Padia, A., Mathews, M.L., Finin, T., Joshi, A., et al.: UCO: a unified cybersecurity ontology. In: Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security (2016)
3. Jia, Y., Qi, Y., Shang, H., Jiang, R., Li, A.: A practical approach to constructing a knowledge graph for cybersecurity. Engineering **4**(1), 53–60 (2018)
4. Gao, P., et al.: Enabling efficient cyber threat hunting with cyber threat intelligence. arXiv preprint arXiv:2010.13637 (2020)
5. Piplai, A., Mittal, S., Joshi, A., Finin, T., Holt, J., Zak, R.: Creating cybersecurity knowledge graphs from malware after action reports. IEEE Access **8**, 211:691–211:703 (2020)
6. Zhao, J., Yan, Q., Liu, X., Li, B., Zuo, G.: Cyber threat intelligence modeling based on heterogeneous graph convolutional network. In: 23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020), pp. 241–256 (2020)
7. Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., Niu, X.: TTPDrill: automatic and accurate extraction of threat actions from unstructured text of CTI sources. In: Proceedings of the 33rd Annual Computer Security Applications Conference, pp. 103–115 (2017)
8. Piplai, A., Mittal, S., Abdelsalam, M., Gupta, M., Joshi, A., Finin, T.: Knowledge enrichment by fusing representations for malware threat intelligence and behavior. In: 2020 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, pp. 1–6 (2020)
9. Milajerdi, S.M., Gjomemo, R., Eshete, B., Sekar, R., Venkatakrishnan, V.: Holmes: real-time apt detection through correlation of suspicious information flows. In: IEEE Symposium on Security and Privacy (SP), vol. 2019, pp. 1137–1152. IEEE (2019)
10. Mittal, S., Das, P.K., Mulwad, V., Joshi, A., Finin, T.: CyberTwitter: using Twitter to generate alerts for cybersecurity threats and vulnerabilities. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 860–867. IEEE (2016)
11. Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R.: 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)
12. Zhu, Z., Dumitras, T.: ChainSmith: automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: IEEE European Symposium on Security and Privacy (EuroS&P), vol. 2018, pp. 458–472. IEEE (2018)
13. Ghazi, Y., Anwar, Z., Mumtaz, R., Saleem, S., Tahir, A.: A supervised machine learning based approach for automatically extracting high-level threat intelligence from unstructured sources. In: 2018 International Conference on Frontiers of Information Technology (FIT), pp. 129–134. IEEE (2018)
14. Zhao, J., Yan, Q., Li, J., Shao, M., He, Z., Li, B.: TIMiner: automatically extracting and analyzing categorized cyber threat intelligence from social data. Comput. Secur. **95**, 101867 (2020)

15. Husari, G., Niu, X., Chu, B., Al-Shaer, E.: Using entropy and mutual information to extract threat actions from cyber threat intelligence. In: 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 1–6. IEEE (2018)
16. Pingle, A., Piplai, A., Mittal, S., Joshi, A., Holt, J., Zak, R.: 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)
17. Jones, C.L., Bridges, R.A., Huffer, K.M., Goodall, J.R.: Towards a relation extraction framework for cyber-security concepts. In: Proceedings of the 10th Annual Cyber and Information Security Research Conference, pp. 1–4 (2015)
18. Satyapanich, T., Ferraro, F., Finin, T.: CASIE: extracting cybersecurity event information from text. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 8749–8757 (2020)
19. Iria, J.: T-rex: a flexible relation extraction framework. In: Proceedings of the 8th Annual Colloquium for the UK Special Interest Group for Computational Linguistics (CLUK 2005), vol. 6, p. 9. Citeseer (2005)
20. McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y., White, P.: Simple algorithms for complex relation extraction with applications to biomedical IE. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 491–498 (2005)
21. Jiang, J., Zhai, C.: A systematic exploration of the feature space for relation extraction. In: Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics. Proceedings of the Main Conference, vol. 2007, pp. 113–120 (2007)
22. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pp. 423–429 (2004)
23. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335–2344 (2014)
24. Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y.: A novel cascade binary tagging framework for relational triple extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1476–1488 (2020)
25. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1105–1116 (2016)
26. Bekoulis, G., Deleu, J., Demeester, T., Develder, C.: Joint entity recognition and relation extraction as a multi-head selection problem. Expert Syst. Appl. **114**, 34–45 (2018)
27. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1227–1236 (2017)
28. Sun, C., et al.: Extracting entities and relations with joint minimum risk training. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2256–2265 (2018)
29. Fu, T.-J., Li, P.-H., Ma, W.-Y.: GraphRel: modeling text as relational graphs for joint entity and relation extraction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1409–1418 (2019)

30. Yuan, Y., Zhou, X., Pan, S., Zhu, Q., Song, Z., Guo, L.: A relation-specific attention network for joint entity and relation extraction. In: International Joint Conference on Artificial Intelligence 2020. Association for the Advancement of Artificial Intelligence (AAAI), pp. 4054–4060 (2020)
31. Dai, D., Xiao, X., Lyu, Y., Dou, S., She, Q., Wang, H.: Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 6300–6308 (2019)
32. MITRE: Cvelist project (2021). https://github.com/CVEProject/cvelist
33. CyberMonitor: Apt & cybercriminals campaign collection (2021). https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections
34. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: Brat: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 102–107 (2012)
35. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
36. Youngja, P.: Cybersecurity embeddings. https://ebiquity.umbc.edu/resource/html/id/379/Cybersecurity-embeddings (2018)