electronics

MDPI

Article

Feature-Enhanced Document-Level Relation Extraction in Threat Intelligence with Knowledge Distillation

Yongfei Li, Yuanbo Guo *, Chen Fang, Yongjin Hu, Yingze Liu and Qingli Chen

School of Cryptographic Engineering, PLA Information Engineering University, Zhengzhou 450001, China
* Correspondence: yuanbo_g@hotmail.com

Abstract: Relation extraction in the threat intelligence domain plays an important role in mining the internal association between crucial threat elements and constructing a knowledge graph (KG). This study designed a novel document-level relation extraction model, FEDRE-KD, integrating additional features to take full advantage of the information in documents. The study also introduced a teacher–student model, realizing knowledge distillation, to further improve performance. Additionally, a threat intelligence ontology was constructed to standardize the entities and their relationships. To solve the problem of lack of publicly available datasets for threat intelligence, manual annotation was carried out on the documents collected from social blogs, vendor bulletins, and hacking forums. After training the model, we constructed a threat intelligence knowledge graph in Neo4j. Experimental results indicate the effectiveness of additional features and knowledge distillation. Compared to mainstream models SSAN, GAIN, and ATLOP, FEDRE-KD improved the F1score by 22.07, 20.06, and 22.38, respectively.

Keywords: threat intelligence; document-level relation extraction; knowledge distillation; knowledge graph

Citation: Li, Y.; Guo, Y.; Fang, C.; Hu, Y.; Liu, Y.; Chen, Q. Feature-Enhanced Document-Level Relation Extraction in Threat Intelligence with Knowledge Distillation. *Electronics* **2022**, *11*, 3715. https://doi.org/10.3390/electronics11223715

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 11 October 2022
Accepted: 9 November 2022
Published: 13 November 2022

1. Introduction

Today, the Internet of Things (IoT) impacts almost every aspect of societal needs [1]. With the rapid development of network and information technology, new cyber threats (e.g., session hijacking, masquerade attack, and interruption) [2] are showing a gradual rising trend. Increasing complexity of attack strategy and the ever-changing attack scenarios make traditional network defense, such as firewalls, hard to resist. In 2019, more than 10,000 new types of cybercrime were committed in Russia [3]. In February 2022, Ukrainian government agencies and banking websites were targeted by large-scale distributed denial-of-service (DDoS) attacks, resulting in the offlining of at least 10 websites [4]. To achieve better command of threat situations and coordinate the response to unknown threats, security experts proposed cyber threat intelligence (CTI) for network defense. Gartner [5] first put forward that CTI is knowledge of existing or emerging threats against assets, including scenarios, mechanisms, indicators, and actionable recommendations, which can provide the subject with countermeasures.

Knowledge of threat intelligence originates from security analysis reports, blogs, social media, etc., which provides powerful data support for situational awareness and active network defense [6]. However, threat intelligence is mainly in the form of natural language, containing a large amount of unstructured data. Thus, it is difficult to visualize the internal relations of crucial elements. To help researchers understand the semantic association of elements quickly, it is necessary to design corresponding algorithms for mining entities and relations between them from large-scale threat intelligence documents to construct a knowledge graph.

Relation extraction aims to identify relations between entities from a given text [7]. As shown in Figure 1, the head entity Attacker "*Mealybug*" and the tail entity Trojan

"*Trojan.Emotet*" can express the relation of "*Use*". Although relation extraction in the general domain has achieved satisfactory results, the mainstream models present the following limitations in the cybersecurity domain: (1) the lack of open-source datasets about threat intelligence; (2) threat intelligence contains plenty of terms such as vulnerability number, malware name, advanced persistent threat (APT) group, etc., with serious out-of-vocabulary (OOV) problem; (3) threat intelligence documents are complex in structure. The frequency of entities in a sentence is extremely low, leading to the serious imbalance in the distribution of data labels. In addition, the current work mainly focusses on text mining at the sentence level. However, in practical scenarios, there may be multiple mentions for an entity and the relations between entities usually depend on at least two sentences for inference [8].

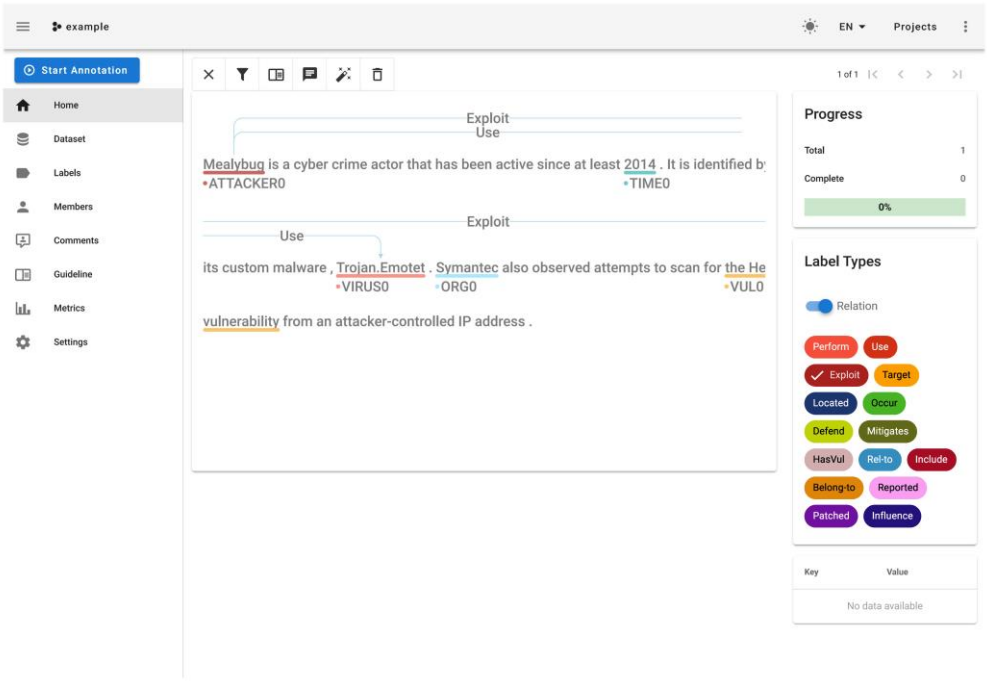

Figure 1. A threat intelligence text containing entities and relations.

To this end, this paper proposes a novel **f**eature-**e**nhanced **d**ocument-level **re**lation extraction model (FEDRE) to improve the in-domain performance of threat intelligence, which integrates new features. Then we introduce a teacher–student model to achieve **k**nowledge **d**istillation (FEDRE-KD). In summary, we present a practical model to convert threat intelligence documents into structured data and construct a knowledge graph. It can be further utilized in threat hunting and decision making. Our contribution can be summarized as follows:

(1) We captured part-of-speech (POS) of entity, width of mention, distance between entity pair, and type of entity as new features in document-level threat intelligence relation extraction. Pre-training model bidirectional encoder representation from transformers (BERT) was applied as encoder to alleviate the OOV problem.

(2) We introduced a teacher–student model, gathering effective information from texts by soft labels, which retains the association between classes and eliminates some invalid redundant information. We achieved knowledge distillation and further improved performance.

(3) We collected 227 threat intelligence documents and manually annotated them based on an ontology we defined. We systematically compared the performance of our model with the mainstream neural network models on the document-level relation extraction task. Experimental results demonstrate the effectiveness of our model. The extraction results were integrated to construct a threat intelligence knowledge graph, realizing the visualization of correlation of key elements.

2. Related Work

2.1. Document-Level Relation Extraction

As an important task of information extraction, relation extraction refers to extracting pre-defined relations from unstructured text based on named entity recognition. Relations between entity pairs can be formally described as triples $< e_1, r, e_2 >$, where e_1 and e_2 are entities and r belongs to relation set $R = \{r_1, r_2, \ldots, r_n\}$.

Early studies [9–12] mainly focused on the relations between entities within a single sentence. However, in practical scenarios, an increasing number of relations need to be inferred through multiple sentences. Recently, research has shifted to document-level relation extraction, which needs to integrate intra-sentence and inter-sentence information and capture interactions between entity mentions. Existing methods are mainly divided into two categories: document graph methods and transformer-based methods. Specifically, Wang et al. [13] encoded the document into global entity representation, local entity representation, and contextual relation representation, and constructed a global heterogeneous graph. Zhang et al. [14] proposed an entity-pair-level document graph and collected contextual information horizontally and vertically to enhance entity pair representation. They captured features of single-hop and multi-hop logical reasoning by using criss-cross attention. For the transformer-based methods, Xu et al. [15] integrated the unique dependence between mentions into a standard self-attention mechanism which ran through the whole encoding module. Yuan et al. [16] captured critical sentence feature using an inter-sentence attention mechanism and designed gating function to combine sentence-level and document-level features. Xie et al. [17] presented EIDER, an evidence-enhanced document-level relation extraction framework consisting of three stages, including joint relation and evidence extraction, evidence-centered relation extraction, and fusion of extraction results, which achieved a significant improvement.

2.2. Threat Intelligence Information Extraction

Indicators of compromise (IOC) can be observed at the early stage of a cyber-attack, which is vital for identifying whether a computer has been hacked. However, existing IOC extraction methods rely heavily on expert knowledge. To tackle this problem, Long et al. [18] proposed an end-to-end sequence labeling model, which can automatically extract IOC from cybersecurity texts using a multi-head self-attention mechanism and contextual features, exploring entities and relations between them has significant applications in cybersecurity. Because of the fast-growing volume of the documents, it is time-consuming and laborious to develop feature templates. Therefore, Gasmi et al. [19] proposed three relation extraction models based on LSTM, reducing troubles caused by feature engineering. Wang et al. [20] constructed an automatic IOC extraction method, iAES, for cybersecurity blogs based on regular expression matching and deep learning models. Satyapanich et al. [21] classified the types of cyber events into five categories and combined semantic features with deep learning to propose a new cybersecurity event extraction system CASIE, which provides data support for building a knowledge graph.

2.3. Knowledge Distillation

In general, large models have good performance and generalization ability because of their complex structure, followed by long training time and high computational overhead. Small models, on the other hand, have limited expressiveness due to their scale. Therefore, it is reasonable to guide small models with the knowledge from large models, so that the former can have comparable or similar performance and significantly reduce training time, thus realizing compression and acceleration.

Hinton et al. [22] first proposed the concept of knowledge distillation. They suggested training a complex network (called the teacher model) and using the output of this model and ground truth to train a simpler network (called the student model). Remero et al. [23] extended Hinton's approach, utilizing the output of the teacher model and the feathers of the middle layer as hints to improve the training process and performance. In addition

to applications in computer vision, knowledge distillation is gradually being applied to natural language processing (NLP). Zhang et al. [24] designed a bipartite graph to discover type constraints between entities and relations based on the entire corpus. Then, they combined such type constraints with neural networks to achieve a knowledgeable model. Furthermore, this model was regarded as a teacher to generate well-informed soft labels and guide the optimization of a student network via knowledge distillation.

2.4. Threat Intelligence Knowledge Graph

Knowledge graph is a semantic network proposed by Google, revealing relations between entities. It has been widely applied in many fields, such as data storage [25], knowledge reasoning [26] and intelligent answering [27]. Threat intelligence knowledge graphs can store massive amounts of information. Experts are capable of constructing corresponding systems based on the interrelated information. Then, they make determinations using logical reasoning based on pre-defined rules, providing support for user decisions.

Gao et al. [28] proposed SECURITYKG, collecting open-source cyber threat intelligence (OSCTI) reports and using a combination of artificial intelligence (AI) and NLP techniques to construct a security knowledge graph. Piplai et al. [29] suggested that the cybersecurity knowledge graph based on information extracted from OSCTI was limited, as the author may focus on only specific aspects of malware. In addition, not all authors were trustworthy. Conflicting information could be detected on the basis of ontology, constraints, and rules after inserting behavior knowledge into the knowledge graph. Mittal et al. [30] presented an end-to-end system for knowledge extraction in the cybersecurity informatics domain. It combined vector space and the knowledge graph (called VKG structure) to represent threat intelligence. In this structure, vectors contained implicit information about entities. On the other hand, the knowledge graph possessed explicit information about entities and their relations. Meanwhile, they created a search engine and a warning system by actively updating the knowledge graph.

3. Framework Architecture

This paper proposes a novel document-level relation extraction model, FEDRE, integrating global and local information. It captures part of speech of entity, width of mention, distance between entity pair, and type of entity as new features. The framework of FEDRE is shown in Figure 2.

As shown in Figure 3, we introduce a teacher–student model, gathering effective information from texts using soft labels.

3.1. Encode Layer

In a given a document, $D = [x_t]_{t=1}^{l}$, x_t represents the word at position t. We marked entity mentions by inserting a special symbol "*" at the start and the end of mentions. We used the pre-trained model BERT as our encoder, obtaining contextual embedding H:

$$H = Bert([x_1, \ldots, x_l]) = [h_1, \ldots, h_l] \tag{1}$$

where $H \in R^{l \times d_1}$, and d_1 is the dimension of the hidden layer in the pre-trained model.

3.2. Representation Layer

We used NLTK, a Python library, to generate POS tags of the input sentence. Then we created a POS embedding matrix P:

$$P = Pos([x_1, \ldots, x_l]) = [p_1, \ldots, p_l] \tag{2}$$

where $P \in R^{l \times d_2}$, and d_2 is the dimension of the POS embedding.

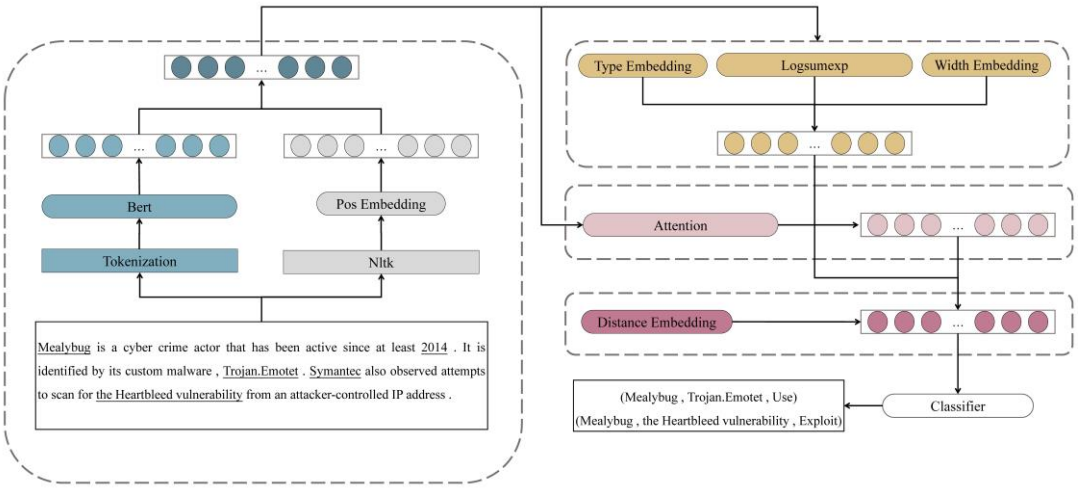

Figure 2. The framework of the proposed model FEDRE. The input will be preprocessed to obtain tokenization and part-of-speech. FEDRE converts them into vectors as mention-level representation. To solve the problem of multiple mentions, the model adopts log-sum-exp pooling to acquire entity-level embedding, combined with type embedding and width embedding. Then FEDRE obtains local contextual embedding using the multi-head attention mechanism. Distance embedding is presented as an additional feature for entity pair. Representations for a specific entity pair are encoded with the embeddings above. Finally, the model feeds them into a classifier and infers the relations in the original input.

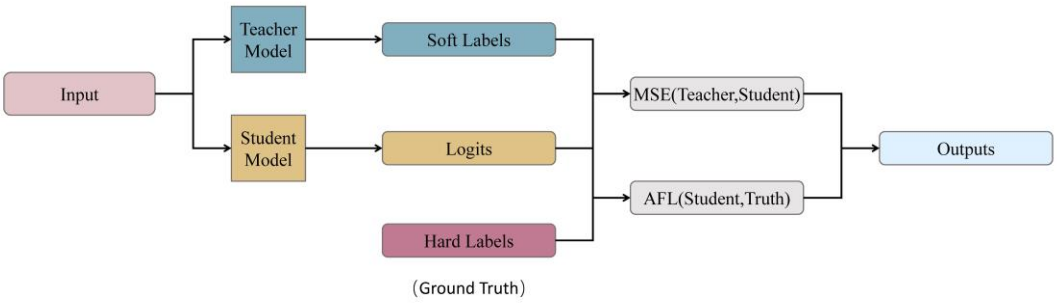

Figure 3. The teacher–student model. FEDRE was trained on the annotated data as the teacher model and generated soft labels. Then the student model was trained on both the annotated data and the predicted soft labels to eliminate invalid redundant information. The total loss was computed by combining the mean squared error loss with the adaptive focal loss.

For each token, we concatenate the contextual embedding and its POS embedding to generate POS-enhanced token representation:

$$C = [h_1|p_1, \ldots, h_l|p_l] = [c_1, \ldots, c_l] \tag{3}$$

where $C \in R^{l\times(d_1+d_2)}$, and [|] denotes the concat operation.

Span width was an important feature for the entity, so we trained a width embedding matrix:

$$W = Width(l_i) = [w_1, \ldots, w_{n_1}] \tag{4}$$

where $w_i \in R^{d_3}$, and d_3 is the dimension of the width embedding.

We took the embedding of "*" at the start of mention and concatenated it with width embedding to obtain width-enhanced mention embedding:

$$mention_{m_j} = c_{m_j} \Big| w_{m_j} \tag{5}$$

For an entity e_i with N_{e_i} mentions $\left\{m_j^i\right\}_{j=1}^{N_{e_i}}$, a log-sum-exp pooling was applied to produce entity embedding:

$$h_{e_i} = log \sum_{j=1}^{N_{e_i}} \exp\left(mention_{m_j}\right) \tag{6}$$

Experimental results showed that entity types contained information required for relation extraction. Therefore, an entity-type embedding matrix was generated to merge type information:

$$T = Type(t_i) = [t_1, \ldots, t_{n_3}] \tag{7}$$

where $t_i \in R^{d_5}$, and d_5 is the dimension of the type embedding.

Given a pre-trained multi-head attention matrix $A \in R^{HD \times l \times l}$, A_{ijk} denotes the attention score from *token j* to *token k* in the *ith* attention head. We first took the attention from the "*" symbol as the mention-level attention, then averaged the attention over mentions of the same entity to obtain entity-level attention $A_i^E \in R^{HD \times l}$, denoting the attention scores from the *ith* entity to all tokens. Then we located important context for the given entity pair (e_s, e_o) by attention matrix, calculating local contextual embeddings.

$$A^{(s,o)} = A_s^E \cdot A_o^E \tag{8}$$

$$q^{(s,o)} = \sum_{i=1}^{HD} A_i^{(s,o)} \tag{9}$$

$$a^{(s,o)} = q^{(s,o)} / 1^T q^{(s,o)} \tag{10}$$

$$c^{(s,o)} = Ha^{(s,o)} \tag{11}$$

Next, representation for a specific entity pair (e_s, e_o) was encoded as:

$$z_s^{(s,o)} = \tan h\left[W_S h_{e_s} + W_{C_1} c^{(s,o)} + W_{D_1} D(d_{so}) + W_{T_1} T(e_s)\right] \tag{12}$$

$$z_o^{(s,o)} = \tan h\left[W_O h_{e_o} + W_{C_2} c^{(s,o)} + W_{D_2} D(d_{os}) + W_{T_2} T(e_o)\right] \tag{13}$$

where d_{so} is the distance between the first mention of entity s and entity o.

To reduce the number of parameters, we exploited the group bilinear, which effectively lowered the computational overhead. Specifically, we divided the entity representation into k equal-sized groups and fused the features to obtain representation of the given entity pair.

$$\left[z_s^1; \ldots; z_s^k\right] = z_s \tag{14}$$

$$\left[z_o^1; \ldots; z_o^k\right] = z_o \tag{15}$$

$$g^{(s,o)} = (\sum_{i=1}^{k} z_s^i W_r^i z_o^i + b_r) \tag{16}$$

3.3. Relation Classification

We calculated the logit of relation r of the given entity pair using a non-linear activation:

$$P(r|e_s, e_o) = \sigma\left(g^{(s,o)}\right) = \sigma(\sum_{i=1}^{k} z_s^i W_r^i z_o^i + b_r) \tag{17}$$

Relation extraction can be regarded as a multi-label classification task. Traditional baselines usually use standard binary cross-entropy loss to tackle this problem, which specify a global threshold as the criterion for whether a relation label exists. However,

models have different confidence in threshold for different entity pairs. In addition, the distribution of entities and relations was extremely unbalanced in this task. Therefore, we adopted adaptive focal loss (AFL) as our loss function. Specifically, we set a learnable dynamic threshold combined with focal loss.

$$L_{AFL} = \sum_{r_i \in P_T} (1 - P(r_i))^{\gamma} log(P(r_i)) + log(P(r_{TH})) \quad (18)$$

3.4. Knowledge Distillation

In this module, a teacher-student model was introduced to realize knowledge distillation, so that model performance could be further improved. Specifically, we firstly obtained teacher model trained by the process mentioned above. Then we used the mean square error (MSE) loss to calculate the difference between logits generated by the student model and soft labels generated by the teacher model. Finally, we combined it with AFL in 3.3 as the overall loss function of the student model.

$$L_{KD} = MSE(Teacher,\ Student) \quad (19)$$

$$L_{RE} = \alpha_1 L_{AFL} + \alpha_2 L_{KD} \quad (20)$$

where α_1 and α_2 are hyperparameters used to balance the two loss functions.

4. Experiment

4.1. Dataset

We annotated 227 threat intelligence documents manually, 151 of which were selected as the training set and the remaining 76 as the test set. The training set contained 1610 entities and 949 relations. Definitions of entities and relations between them are shown in Tables 1 and 2.

Table 1. Distribution of entities.

Entity Type	Definition	Training Set	Test Set
ATTACK	Technique	298	135
ORG	Organization/Vendor	261	136
TIME	Time	233	97
VIRUS	Virus/Script	210	83
LOCATION	Country/Region	206	89
SOFTWARE	Legitimate Software	125	57
ATTACKER	Attacker	79	28
OS	Operation System	61	25
VULNERABILITY	Vulnerability	50	23
VERSION	Version	40	12
Course-of-action	Defense Strategy	31	24
EVENT	Attack Event	16	7
Total	/	1610	716

Table 2. Distribution of relations.

Relation Type	Definition	Training Set	Test Set
Target	Target of Attack/Attacker	239	116
Perform	Perform Attack	203	69
Rel-to	Associated	138	48
Use	Use Software/Virus	82	28
Occur	Occurred Time	59	22
Influence	Influence by Attack	59	34
HasVul	Contain Vulnerability	49	17
Mitigate	Defend Against an Attacker	35	21

Table 2. *Cont.*

Relation Type	Definition	Training Set	Test Set
Located	Located in	33	21
Exploit	Exploit Vulnerability	20	8
Reported	Reporting Time	11	4
Patched	Patching Time	7	5
Defend	Carry out Defense Strategy	7	16
Total	/	942	409

Threat intelligence ontology was constructed, as shown in Figure 4.

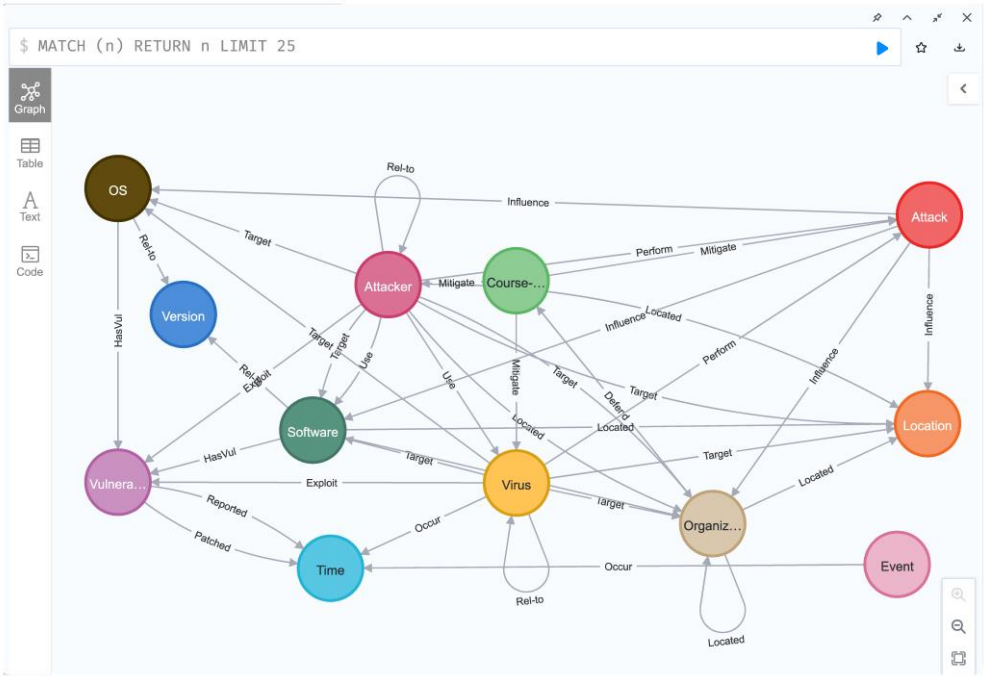

Figure 4. The threat intelligence ontology. Crucial elements and relationships between them are displayed above. For instance, entity "Attacker" is connected with entity "OS" by relation "Target".

4.2. Experiment Setup

Our model was trained on Nvidia Geforce RTX 3090 GPU based on Pytorch1.7.1. We used cased BERT-based as the pre-trained encoder for threat intelligence. We trained the model for 100 epochs with batch size 8, using the AdamW optimizer with warmup and early stop strategy (If the performance was not improved for 20 consecutive epochs, the training process would be stopped). The learning rate was set to 5×10^{-5} for BERT and 1×10^{-4} for other layers. The loss weight of the teacher model and student model was set to 1:1, i.e., $\alpha_1 = \alpha_2 = 1$. We chose 25 as the size of the POS embedding (p), type embedding (t), width embedding (w), and distance embedding (d).

To tackle the imbalance of the dataset, we adopted random oversampling to copy minority classes before training our model. Specifically, tokens were replaced by their synonyms to create new samples.

Following prior studies, we introduced commonly used metrics in the relation extraction task to evaluate our model, i.e., precision (P), recall (R), and F1-score (F1). Additionally, we used F1 as the main evaluation metric. Furthermore, we presented time overhead as another index. Meanwhile, we calculated performance for each kind of relation- analysing model at a more granular level.

We compared our model with three excellent works, including SSAN [15], GAIN [31], and ATLOP [8]. For fair comparisons, we use cased BERT-based as the base encoder for all methods.

4.3. Result and Analysis

4.3.1. Model Comparison

Table 3 presents the relation extraction results of our model and baseline models on our dataset. First, compared to ATLOP, FEDRE improved its performance significantly by 21.01/22.61/22.38 P/R/F1 on the test set. This demonstrates the usefulness of additional features during inference. In addition, we concluded that FEDRE-KD outperformed FEDRE by 4.51 in the F1, proving that knowledge distillation can effectively be promoted. The experimental results also show that FEDRE-KD performed better than all the baseline models. The F1 of our model was 21.07 higher than that of SSAN and 20.06 higher than that of GAIN.

Table 3. Performance comparison of different models.

Model	P (%)	R (%)	F1 (%)	Overhead (h)
SSAN [15]	50.57	46.86	48.64	2.85
GAIN [31]	50.77	48.58	49.65	2.80
ATLOP [8]	51.36	36.71	42.82	2.68
FEDRE	72.37	59.32	65.20	2.28
FEDRE-KD	**79.60**	**62.00**	**69.71**	**2.00**

4.3.2. Ablation Study

We conducted ablation studies to further analyze the utility of each module in FEDRE. The results are shown in Table 4.

Table 4. Ablation study of FEDRE.

Model	P (%)	R (%)	F1 (%)	Overhead (h)
FEDRE	**72.37**	**59.32**	**65.20**	2.28
NoPOS	63.19	49.06	55.24	1.81
NoWidth	57.74	48.50	52.72	1.81
NoType	62.63	47.01	53.71	1.96
NoDistance	60.40	48.38	53.73	**1.68**

We first removed the POS embeddings and width embeddings, which are denoted as NoPOS and NoWidth, respectively. It was obvious that performance would drop if any feature of them was removed, indicating that the information for POS and width is important for relation prediction. Specifically, we found that verbs and nouns were more likely to be associated to other tokens in threat intelligence documents. Meanwhile, integrating width embedding could enrich representation at the mention level.

Then, we removed the entity-type embeddings, which is denoted as NoType. The performance dropped sharply, by 11.49. There existed different relations between different kinds of entities. For instance, "Patched" would only appear when the head entity belonged to "Vulnerability" and the tail entity belonged to "Time". Therefore, integrating type embeddings can enrich representation at the entity level.

Finally, we removed the distance embeddings, which is denoted as NoDistance. The performance dropped by 11.47. This further demonstrates that the distance of two entities could enrich representation at the entity-pair level.

4.3.3. Fine-Grained Performance Comparison

To further observe the ability of introducing additional features and knowledge distillation to fit different types of data, Table 5 shows the fine-grained performance in detail.

Table 5. Fine-grained performance comparison.

Model	SSAN [15]			GAIN [31]			ATLOP [8]		
	P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)
Target	62.74	60.29	61.49	59.52	56.99	58.23	61.02	59.55	60.28
Perform	48.19	53.09	50.52	42.48	59.13	49.44	46.53	41.88	44.08
Rel-to	58.77	40.56	48.00	42.92	37.08	39.79	48.25	24.38	32.39
Use	58.30	60.00	59.14	52.87	76.43	62.50	44.52	51.07	47.57
Occur	20.56	16.36	18.22	26.61	30.91	28.60	42.88	26.36	32.65
Influence	30.36	20.69	24.61	51.17	25.88	34.37	46.87	15.00	22.73
HasVul	72.16	59.22	65.05	50.23	23.53	32.05	74.11	18.82	30.02
Mitigate	30.78	53.97	39.20	40.70	50.48	45.07	43.50	12.38	19.27
Located	45.57	33.65	38.71	47.68	24.76	32.59	50.94	27.14	35.41
Exploit	46.03	67.92	54.87	50.50	20.00	28.65	24.76	12.50	16.61
Reported	34.02	27.50	30.41	16.12	25.00	19.60	43.43	22.50	29.64
Patched	44.71	48.67	46.61	41.33	36.00	38.48	68.45	52.00	59.10
Defend	35.05	9.79	15.31	21.33	7.50	11.10	90.00	5.62	10.58

Model	FEDRE			FEDRE-KD		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
Target	69.72	62.81	66.08	82.86	63.97	72.20
Perform	64.47	62.82	63.63	71.88	73.40	72.63
Rel-to	89.86	68.13	77.50	92.79	73.57	82.07
Use	55.32	92.86	69.33	58.54	85.71	69.57
Occur	50.00	50.00	50.00	78.57	50.00	61.11
Influence	61.54	23.53	34.04	71.43	14.71	24.40
HasVul	92.31	80.00	85.72	88.46	76.67	82.14
Mitigate	84.62	52.38	64.71	66.67	38.10	48.49
Located	60.00	14.29	23.08	62.50	23.81	34.48
Exploit	61.26	40.25	48.58	62.50	45.45	52.63
Reported	54.12	47.67	50.69	75.00	60.00	66.67
Patched	68.91	60.71	64.55	85.71	85.71	85.71
Defend	58.99	14.14	22.81	57.13	16.55	25.67

Combined with the distribution of relations in Table 2, it can be intuitively found that introducing additional features significantly improved the classification ability of most types, such as ”Target”, “Perform”, and “Use”. Meanwhile, it is obvious that introducing knowledge distillation brought further promotion, with the maximum improvement of 21.16.

4.3.4. Choice of Sampling Technique

To alleviate the imbalance of the dataset, oversampling and undersampling were introduced. The results in Table 6 prove that the oversampling algorithm could significantly improve the performance. However, the undersampling algorithm suffered from the risk of unreasonably removing instances of loss of important information.

Table 6. Performance comparison of different sample techniques.

Sample Technique	P (%)	R (%)	F1 (%)
None	58.46	48.67	53.12
Under Sampling	59.06	46.43	51.99
Under Sampling + Over Sampling	**76.44**	50.70	60.96
Over Sampling	72.37	**59.32**	**65.20**

4.4. Threat Intelligence Knowledge Graph Construction

We inputted threat intelligence documents with annotated entities into the trained FERED-KD model. The model predicted relations selected from predefined relation sets for

all the entity pairs. Then we inserted the entity-relation set into the knowledge graph using the neo4j-admin command. The results are shown in Figure 5.

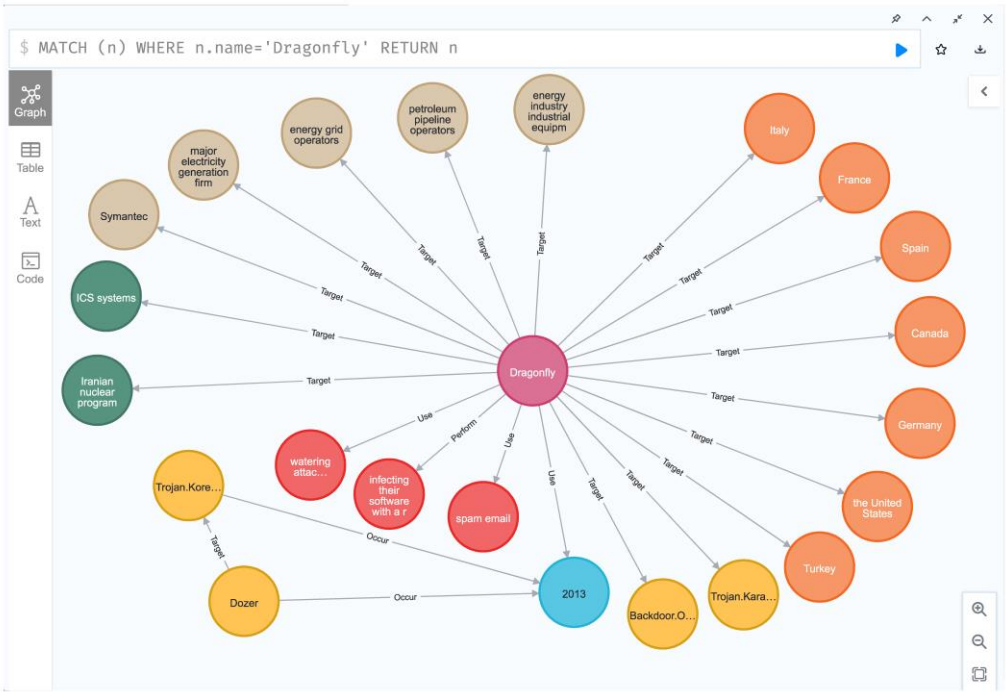

Figure 5. Part of threat intelligence knowledge graph. Documents were fed into the trained FEDRE-KD model to obtain structured data and construct KG. This figure displays an instance of KG center on Dragonfly.

5. Conclusions

In this paper, we propose a novel document-level relation extraction model introducing additional features and knowledge distillation. Experimental results show that integrating features can enhance the fitting ability of most types. Meanwhile, the teacher–student model can further improve the performance. Additionally, we constructed a threat intelligence knowledge graph displaying internal association between vital elements in documents. In summary, the proposed model FEDRE-KD can provide significant support to transform network defense from passive to active. It can be utilized in tracing an attacker and making auxiliary decisions. In future work, we plan to extend our dataset to avoid imbalance. In addition, we will consider solving the overlap entity problem. Finally, we will focus on knowledge reasoning to obtain new information.

Author Contributions: Conceptualization, Y.L. (Yongfei Li); formal analysis, Y.L. (Yongfei Li); methodology, Y.L. (Yongfei Li); software, Y.L. (Yongfei Li); supervision, Y.G., C.F., Y.H., Y.L., (Yingze Liu) and Q.C.; writing—original draft, Y.L. (Yongfei Li); writing—review and editing, Y.G., C.F., Y.H., Y.L. (Yingze Liu), and Q.C.; conceptualization, Y.L. (Yongfei Li); formal analysis, Y.L. (Yongfei Li); Methodology, Y.L. (Yongfei Li); Software, Y.L. (Yongfei Li); Supervision, Y.G., C.F., Y.H., Y.L. (Yingze Liu), and Q.C.; writing—original draft, Y.L. (Yongfei Li); All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant numbers 61501515 and 61601515).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dohare, I.; Singh, K.; Ahmadian, A.; Mohan, S. Certificateless aggregated signcryption scheme for cloud-fog centric industry 4.0. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6349–6357. [CrossRef]
2. Thirumalai, C.; Mohan, S.; Srivastava, G. An efficient public key secure scheme for cloud and IoT security. *Comput. Commun.* **2020**, *150*, 634–643. [CrossRef]
3. Simonov, N.; Klenkina, O.; Shikhanova, E. Leading Issues in Cybercrime: A Comparison of Russia and Japan. In Proceedings of the 6th International Conference on Social, Economic, and Academic Leadership (ICSEAL-6-2019), Prague, Czech, 13–14 December 2019; pp. 504–510.
4. Maschmeyer, L.; Dunn Cavelty, M. Goodbye Cyberwar: Ukraine as Reality Check. *CSS Policy Perspect.* **2022**, *10*. [CrossRef]
5. McMillan, R. Definition: Threat Intelligence. *March*. 2013. Available online: https://www.gartner.com/en/documents/2487216 (accessed on 30 September 2022).
6. Liu, C.; Wang, J.; Chen, X. Threat intelligence ATT&CK extraction based on the attention transformer hierarchical recurrent neural network. *Appl. Soft Comput.* **2022**, *122*, 108826.
7. Nguyen, T.H.; Grishman, R. Relation Extraction: Perspective from Convolutional Neural Networks. In Proceedings of the Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, CO, USA, 5 June 2015; pp. 39–48.
8. Zhou, W.; Huang, K.; Ma, T.; Huang, J. Document-Level Relation Extraction with Adaptive Thresholding and Localized Context Pooling. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; pp. 14612–14620.
9. Peng, H.; Gao, T.; Han, X.; Lin, Y.; Li, P.; Liu, Z.; Sun, M.; Zhou, J. Learning from Context or Names? An Empirical Study on Neural Relation Extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual Event, 16–20 November 2020; pp. 3661–3672.
10. Soares, L.B.; Fitzgerald, N.; Ling, J.; Kwiatkowski, T. Matching the Blanks: Distributional Similarity for Relation Learning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 2895–2905.
11. Guo, Z.; Zhang, Y.; Lu, W. Attention Guided Graph Convolutional Networks for Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 241–251.
12. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1441–1451.
13. Wang, D.; Hu, W.; Cao, E.; Sun, W. Global-to-Local Neural Networks for Document-Level Relation Extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual Event, 16–20 November 2020; pp. 3711–3721.
14. Zhang, L.; Cheng, Y. A Densely Connected Criss-Cross Attention Network for Document-level Relation Extraction. *arXiv* **2022**, arXiv:2203.13953.
15. Xu, B.; Wang, Q.; Lyu, Y.; Zhu, Y.; Mao, Z. Entity Structure Within and Throughout: Modeling Mention Dependencies for Document-Level Relation Extraction. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; pp. 14149–14157.
16. Yuan, C.; Huang, H.; Feng, C.; Shi, G.; Wei, X. Document-level relation extraction with entity-selection attention. *Inf. Sci.* **2021**, *568*, 163–174. [CrossRef]
17. Xie, Y.; Shen, J.; Li, S.; Mao, Y.; Han, J. Eider: Evidence-enhanced Document-level Relation Extraction. *arXiv* **2021**, arXiv:2106.08657.
18. Long, Z.; Tan, L.; Zhou, S.; He, C.; Liu, X. Collecting indicators of compromise from unstructured text of cybersecurity articles using neural-based sequence labelling. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 4–19 July 2019; pp. 1–8.
19. Gasmi, H.; Laval, J.; Bouras, A. Information extraction of cybersecurity concepts: An lstm approach. *Appl. Sci.* **2019**, *9*, 3945. [CrossRef]
20. Wang, W.; Ning, K.; Song, H.; Lu, M.; Wang, J. An Indicator of Compromise Extraction Method Based on Deep Learning. *J. Comput.* **2021**, *44*, 15.
21. Satyapanich, T.; Ferraro, F.; Finin, T. Casie: Extracting cybersecurity event information from text. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 8749–8757.
22. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
23. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv* **2014**, arXiv:1412.6550.
24. Zhang, Z.; Shu, X.; Yu, B.; Liu, T.; Zhao, J.; Li, Q.; Guo, L. Distilling knowledge from well-informed soft labels for neural relation extraction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 9620–9627.
25. Liu, Q.; Li, Y.; Duan, H.; Liu, Y.; Qin, Z. Knowledge Graph Construction Techniques. *J. Comput. Res. Dev.* **2016**, *53*, 582–600. [CrossRef]

26. Lv, X.; Han, X.; Hou, L.; Li, J.; Liu, Z.; Zhang, W.; Zhang, Y.; Kong, H.; Wu, S. Dynamic Anticipation and Completion for Multi-Hop Reasoning over Sparse Knowledge Graph. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual Event, 16–20 November 2020; pp. 5694–5703.
27. Zhou, K.; Zhao, W.X.; Bian, S.; Zhou, Y.; Wen, J.-R.; Yu, J. Improving conversational recommender systems via knowledge graph based semantic fusion. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Diego, CA, USA, 6–10 July 2020; pp. 1006–1014.
28. Gao, P.; Liu, X.; Choi, E.; Soman, B.; Mishra, C.; Farris, K.; Song, D. A System for Automated Open-Source Threat Intelligence Gathering and Management. In Proceedings of the 2021 International Conference on Management of Data, Xi'an, China, 20–25 June 2021; pp. 2716–2720.
29. Piplai, A.; Mittal, S.; Abdelsalam, M.; Gupta, M.; Joshi, A.; Finin, T. Knowledge enrichment by fusing representations for malware threat intelligence and behavior. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–10 November 2020; pp. 1–6.
30. Mittal, S.; Joshi, A.; Finin, T. Cyber-all-intel: An ai for security related threat intelligence. *arXiv* **2019**, arXiv:1905.02895.
31. Zeng, S.; Xu, R.; Chang, B.; Li, L. Double Graph Based Reasoning for Document-level Relation Extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual Event, 16–20 November 2020; pp. 1630–1640.