FSSRE: Fusing Semantic Feature and Syntactic Dependencies Feature for Threat Intelligence Relation Extraction

Xuren Wang, Mengbo Xiong*
Information Engineering College
Capital Normal University
Beijing, China
{wangxuren, 2181002015}@cnu.edu.cn

Famei He*
Liabrary
Beijing Instiuite of Technology
Beijing, China
hefm@bit.edu.cn

Peian Yang*, Binghua Song, Jun Jiang, Zhengwei Jiang
Institute of Information Engineering
Chinese Academy of Sciences
School of Cyber Security
University of Chinese Academy of Sciences
Beijing, China
{yangpeian, songbinghua, jiangjun, jiangzhengwei}@iie.ac.cn

Zihan Xiong
Information Engineering College
Capital Normal University
Beijing, China
2181002064@cnu.edu.cn

***Abstract*—Threat intelligence relation extraction plays an important role in threat intelligence text analysis and processing. To extract the relation between two threat entities in a sentence, we develop a novel framework called FSSRE which fuses sematic feature and syntactic dependencies feature for threat intelligence relation extraction. We utilize graph convolutional networks (GCN) to extract syntactic dependencies features, and utilize Sentence-BERT to extract contextual semantic features. To keep vital information with irrelevant content removed to the most extent, we further apply a novel pruning strategy, SDP-VP, to the input trees. With retaining the shortest path and nodes that are *K* hops away from nodes on the shortest path, we give the edge connected to the verb nodes a weight of *w* times. We create an advanced persistent threat (APT) intelligence entities and intra-sentence relations dataset, APTER-SENT, for that there is no public dataset can be used for relation extraction research in the threat intelligence field. Experimental results on APTER-SENT demonstrate improved performance over competitive baselines. At the same time, we also conducted experiments on the SemEval-2010 dataset. The results of the experiment indicate that our method is still effective on this dataset.**

Keywords—relation extraction, threat intelligence, APTER-SENT, GCN, Sentence-BERT, SDP-VP

I. INTRODUCTION

With the rapid development of network and information technology, new types of threats and attacks represented by Advanced Persistent Threat (APT) are showing a continuous and expanding development trend. APT attacks mainly use special Trojan to target computers to steal confidential information, commercial information of important enterprises, and destroy network infrastructure. Threat intelligence is a kind of evidence-based knowledge, including context, mechanism, labeling, meaning, and recommendations that can be implemented. Threat intelligence reports usually describe a malicious organization using some malicious software to launch an attack. That is, the report contains information such as attacker, target, purpose, and approach. The report also contains file HASH (e.g. 356A192B7913B04C54574D18C28D46E6 395428 AB), encryption algorithm (e.g. AES128-ECB), counter measure (keep CMS plugins up-to-date), etc. Security companies release massive amounts of threat intelligence every day. Most of threat intelligence is presented in text, which cannot visually show the connections between the attack events. It is not conducive to the rapid perception of abnormalities by security operators.

In order to help researchers quickly understand the connection between new threat events and previous threat events, it is very important to design algorithms that can extract the relationships between threat intelligence entities from a large number of documents. And the task of extracting threat intelligence relation is one of the key tasks in constructing a knowledge graph of threat intelligence. Although the effect of relation extraction in the general field is good, there are still some problems in relation extraction in the field of threat intelligence: (1) There is no public dataset that can be used for relation extraction research in the threat intelligence field; (2) The longer sentence length of threat intelligence text makes it difficult to fully and efficiently extract sentence features and not all tokens in the sentence are related to the relation between the entity pair of target. (As shown in TABLE I, the sentence length of threat intelligence text is longer than that of general domain text); (3) Because the threat intelligence text contains file HASH, encryption algorithm, counter measure and other professional domain information, the Out-Of-Vocabulary (OOV) problem will inevitably occur in the word embedding process. However, the existing relation extraction model cannot solve the above problems.

TABLE I. SENTENCE LENGTH STATISTICS FOR OUR DATASET APTER-SENT (THREAT INTELLIGENCE TEXT) AND THE SEMEVAL-2010 DATASET

dataset	Average Sentence Length(Token)	Average Sentence Length(Char)
SemEval-2010	19	85
APTER-SENT	29	184

DOI reference number: 10.18293/SEKE2021-088

We regard the threat intelligence relation extraction task as a multi-classification task and construct a new model to extract relation. This model fuses semantic feature and syntactic dependencies feature for threat intelligence relation extraction. Graph convolutional networks (GCN) is used to capture the syntactic dependencies of the text. We observe that the longer sentence length of threat intelligence text makes it difficult to extract sentence features sufficiently and effectively, and not all tokens in the sentence are related to the relation between the entity pairs of target. In order to solve this problem, we need a pruning method to incorporate relevant information with maximally removing irrelevant content. We also found that the verbs in the text often play a decisive role in the classification of the relation between entities, so we propose a new pruning method of verb-based shortest dependent path pruning (SDP-VP). SDP-VP keeps the shortest dependent path and its K-hop nodes, and gives the edge connected to the verb nodes a weight of w times to form a new dependency graph, which help extract relation between entities more efficiently. But at the same time, there are new problems. If we only use structural information to extract the relations between entities in the longer text, we will lose semantic information, but semantic information is also very important for the task of relation extraction. Therefore, we use Sentence-BERT [1] to capture contextual semantic information, and at the same time alleviate the Out-Of-Vocabulary (OOV) problem caused by special text like HASH value, encryption algorithm and counter measure in the field of threat intelligence. Finally, semantic feature and syntactic dependencies feature are fused to classify the relation.

II. Related Work

Existing general domain relation extraction methods can be divided into two categories, sequence-based and dependency-based.

Sequence-based methods mainly rely on word sequences. This method uses the entire text as input, and uses convolutional neural networks (CNN), long and short-term memory networks (LSTM), etc. to capture the con-textual information of the text, and then extract the relation. Zhang et al. [2] first combines the LSTM sequence model with an entity position attention mechanism that is more suitable for relation extraction. Verga et al. [3] used Transformer to encode the context, and each encoded word generated two position-specific representations through the head and tail multi-layer perceptron (MLP) to achieve relation extraction. The above methods all promote the progress of relation extraction, but the above models only uses the contextual semantic information in the text and cannot capture the dependency information of the sentence.

The dependency-based model incorporates the dependency tree into the model. For example, Song et al. [4] proposed a graph-state LSTM model, which uses parallel state to model each word, and enriches state values through messaging to achieve entity extraction. Zhang et al. [5] proposed an extended graph convolutional network. In order to hold relevant information and remove irrelevant content to the maximum, the model further applies a new pruning strategy to the input tree. The pruning method is to retain the K-hops nodes that are the shortest dependent path between two entities. This pruning strategy can be visually called "hard pruning". Its advantage is to reduce the nodes with less information, reducing the size of the dependent graph, and improve the efficiency of model processing. Guo et al. [6] proposed an Attention Guided Graph Convolutional Networks (AGGCNs) model that directly takes completely dependent trees as input. However, a "soft pruning" method is proposed, which automatically learns how to selectively focus on the relevant substructures that are useful for the relation extraction task. The advantage of this method is that it does not sub-tract any node on the dependency graph, but assign different weight values to the edges on the graph, so that the model extracts text representations that are more beneficial to the relation extraction task. Can [7] proposed a novel model, which is based on the basic information in the SDP enhanced with information selected by several attention mechanisms with kernel filters, namely RbSP (Richer-but-Smarter SDP). But the extracted context information and the extracted dependency structure information are both extracted based on the input pruned dependency graph, and the original context information of the text is not obtained. In essence, Can's work is a dependency-based method. Context semantic information is very important for relation extraction, which is ignored in the above methods. Moreover, the above papers all propose relation extraction models for general domain datasets, which are directly used to extract the relationship between entities in threat intelligence domain. For example, threat intelligence sentences are usually longer than general domain sentences, and it is not easy to extract sentence features for relation extraction. It is necessary to propose a suitable pruning method to solve this problem, and pruning method can also improve the efficiency of model processing.

Threat intelligence entity relation extraction is relations' extraction for specific fields. [8] proposed a system for creating semantic triples on cyber security text, using deep learning methods to extract possible relation. Du et al. [9] propose a knowledge graph for People-Readable Threat Intelligence recommendation (PRTIRG) and incorporates knowledge graph representation into PRTI recommender system for click-through prediction. Wang et al. [10] propose a distant supervision relationship extraction method RL-ET-PCNN-ATT based on the PCNN-ATT model. Verbs are very important for judging the relationship between threat intelligence entities, which is not discussed in the above methods.

III. Dataset

In order to solve the problem that there is no public dataset that can be used for relation extraction research in the threat intelligence field, we constructed threat intelligence entities and intra-sentence relations dataset, APTER-SENT, through manual annotation. APTER-SENT includes annotations for cybersecurity-related entities and relations. The APT reports we marked come from web. They have been issued by major security vendors (such as Kaspersky, FireEye, etc.) since 2008. We have marked 12,906 sentences in reports published between 2015 and 2020. After preprocessing, 23225 triples are obtained. To ensure the quality of the dataset, a three-person cross-check is used in manual annotation. We did not use automatic and semi-automatic annotation solutions, although they are better at facing texts in general domains, they will generate a lot of noise data when facing texts in the threat intelligence domain. According to STIX2.0, "National Standard of the People's

Republic of China-Information Security Technology - Cybersecurity Threat Information Format Specification" and domain experts' knowledge, we predefined 36 entity categories. And we predefined 8 relation types. TABLE II shows 36 predefined entity types and their corresponding examples and quantities. And we predefined 8 relation types as shown in TABLE III. The example is only one case in the instance relationship but not all, such as "located" which can also indicate that the file is located in the operating system such as "The "d3d9.dll" file is malicious and is loaded into memory of Windows".

TABLE II. ENTITY TYPE AND EXAMPLE TABLE

Entity type	Example	Counts
threatActor_name	APT10	1447
threatActor_aliases	MenuPass	62
program_language	JavaScript	81
security_team	AntiyCERT	349
vulnerability_cve	CVE-2014-0160	43
vul_aliases	Heartbleed	38
encryption_algo	AES128-ECB	157
sample_name	control.exe	711
sample_function	Delete files	513
government	FBI	103
target_crowd	military	181
attack_activity	Watering hole	270
counter_measure	Keep CMS plugins up-to-date	118
sub_activity	encrypts this data	771
OS_name	Windows	149
email_evil	acc.signnin.send@gmail.com	31
malware	Reaver	1439
string	VIEWS0018x	342
domain_evil	www.tashdqdxp.com	331
domain	https://www.proofpoint.com	18
attack_goal	steal e-mail and contacts	230
location	Western Europe	543
industry	maritime industries	111
company	Microsoft	170
function	postDown()	201
protocol	HTTP	170
person	Tom Smith	445
IP	192.168.1.206	6
IP_evil	98.126.156.210	71
md5	292843976600e8ad2130224d70356bfc	167
sha1	356A192B7913B04C54574D18C28D46E6 395428AB	16
sha2	12dedcdda853da9846014186e6b4a5d6a82b a0cf61d7fa4cbe444a010f682b5d	421
url	https://support.google.com/mail/answer/703 6019	15
url_evil	http://download.data-server.cloudns.club/GAZA2017.mdb	261
time	November 2017	417
tool	Nmap	2475

TABLE III. RELATION TYPE AND EXAMPLE TABLE, THE RED ENTITY REPRESENTS THE HEAD ENTITY, AND THE BLUE ENTITY REPRESENTS THE TAIL ENTITY.

Relation type	Example	Counts
attack	The Gaza cybergang's attacks have never slowed down and its typical targets include government entities/embassies , oil and gas.	652
located	The energy sector in Europe and North America is being targeted by a new wave of cyber attacks.	555
part_of	Costin Raiu , director of Global Research and Analysis Team at Kaspersky Lab , was the first to find a code connection between APT17 and the backdoor in the infected.	2290
occur_time	WhiteBear focused on various embassies and diplomatic entities around the world in early 2016.	501
use	Gaza cybergang started using the CVE 2017-0199 vulnerability which enables direct code execution.	1638
launch	Proofpoint detected and blocked spearphishing emails from Leviathan targeting a US shipbuilding company	657
goal	Gaza cybergang started using the CVE 2017-0199 vulnerability which enables direct code execution.	1360
find	CVE-2017-8759 is the second zero-day vulnerability used to distribute FINSPY uncovered by FireEye in 2017.	459

APTER-SENT and SemEval-2010 dataset are both focused on annotating intra-sentence relations. The difference is that APTER-SENT is a threat intelligence field dataset and SemEval-2010 is a general field dataset. The former contains at least one relation in each sentence, while the latter contains only one relation ((1.8 vs. 1.0 relations per sentence). TABLE IV gives a comparison of statistics among the two datasets. SemEval-2010 dataset only marked entity position but did not define the entity type. For the convenience of comparison, we counted the number of entities, the entities counted contain the entity types in Table II. The statistics of the number of relations in the two datasets includes the "Other" type.

TABLE IV. DATASET STATISTICS FOR OUR DATASET APTER-SENT AND THE SEMEVAL-2010 DATASET

Statistics	APTER-SENT	SemEval-2010
Entities	12873	21434
Relations	23225	10717
Sentences	12906	10717
Relations/Sentence	1.8	1.0

IV. MODELS

The overall architecture of the proposed FSSRE is illustrated in Fig. 1. It is mainly composed of four parts: (1) Word Processor which extracts word-level semantic features; (2) Regional Dependency Feature Extractor that learn syntactic dependencies information; (3) Semantic Feature Extractor that learn contextual Semantic information; (4) Relation Classifier that classifies relation between entity pairs into predefined categories.

A. Word Processor

In order to fully capture the threat intelligence text features, we use four features to form the final word embedding:

a. We use the pre-trained 300-dimensional GloVe vectors.
b. We use the BERT [11] model to embed words in the text to alleviate the OOV problem.
c. Part-of-speech (POS) being very important for the relation extraction task.
d. Entity tags are beneficial for relation extraction between entities.

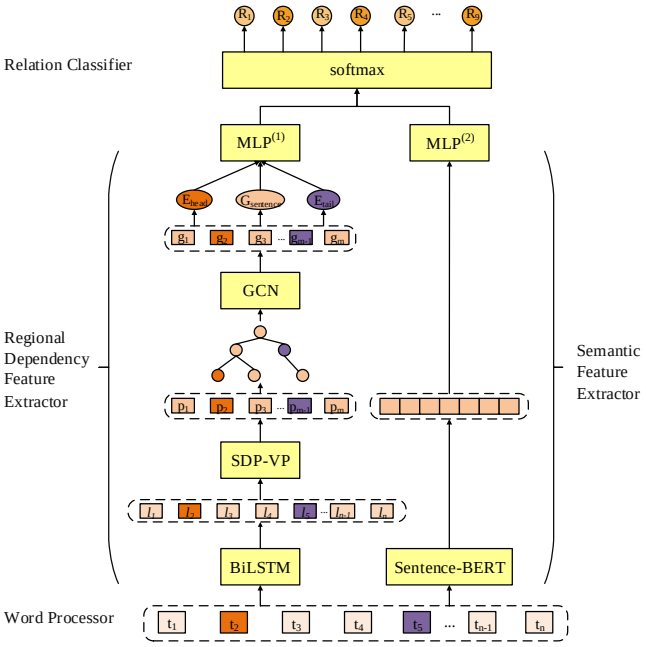

Fig. 1. Overview of FSSRE

As shown in Fig. 1, the input is a sequence of sentences $S = \{t_1, t_2, \cdots, t_n\}$, where t_n represents the n-th token in the sentence. Our final input representation for token t_n is:

$$t_n = G_n + B_n + P_n + E_n \tag{1}$$

Where G_n is the GloVe token embedding vector, B_n is the BERT embedding vector, P_n is the POS embedding and E_n is the NER embedding. Some modules in Word Processor can be replaced with a wide range of different neural network designs. For example, FastText [12] can be used to replace GloVe. It is very flexible and can be adjusted according to needs.

B. Regional Dependency Feature Extractor

a) SDP-VP: To capture the regional dependency feature of threat intelligence text, we use graph convolutional networks (GCN) to process the dependency graph generated by the text. We have noticed that tokens of the verb part are crucial in determining the relation between entities, which plays a decisive role in judging the relation between entities. We propose a new pruning method SDP-VP. SDP-VP keeps the shortest dependent path and the nodes that are K hops away from the node above it, and gives the edge of the verb part-of-speech node a weight of w times to form a new dependency graph. As shown in the Fig. 2, assuming that nodes N_1, N_5 and N_7 are verb nodes, the red path represents the shortest dependent path between entity E_1 and entity E_2. After SDP-VP pruning, in addition to the nodes on the shortest path will be retained, nodes that are $K(K = 1)$ hops away from the shortest path will also be retained. And the weight of the edge with the verb part of speech node is $w(w = 2)$ times the weight of other edges. Finally, the pruning result as shown in the Fig. 2 is obtained.

This method holds the advantages of "hard pruning" and "soft pruning", which can not only reduce the size of the dependent graph, improve the efficiency of model processing, but also give higher weight to the edges of the verb part of speech, making the model more concerned information that is more favorable to relation extraction. Following Zhang et al [5], we further merge the input sentence sequence S through the bidirectional long short-term memory network (BiLSTM) to obtain the word embedding with context information to obtain a new sentence sequence $S_L = \{l_1, l_2, \cdots, l_n\}$. This BiLSTM contextualization layer is trained jointly with the rest of the network.

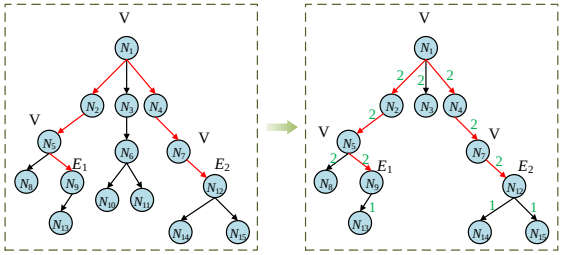

Fig. 2. The Processing of SDP-VP ($K = 1, w = 2$)

$$\vec{h}_n = LSTM_{fw}(t_n, \vec{h}_{n-1}) \tag{2}$$

$$\overleftarrow{h}_n = LSTM_{bw}(t_n, \overleftarrow{h}_{n-1}) \tag{3}$$

$$l_n = [\vec{h}_n; \overleftarrow{h}_n] \tag{4}$$

For each word t_n of the sentence, the hidden layer state $\vec{h}_n$ is obtained through the forward LSTM, and the hidden state $\overleftarrow{h}_n$ is obtained through the backward LSTM, and the two are connected to obtain the hidden layer states l_n. l_n is the word embedding to indicate the state of the hidden layer obtained after t_n passing BiLSTM.

The sentence sequence S_L is pruned by SDP-VP to obtain a new sequence $P = \{p_1, p_2, \cdots, p_m\}$, p_m represents the m-th token in the sequence after pruning.

b) Extract Dependency Graph Regional Dependency Feature: In order to explicitly use structural information to further improve the model, we propose to use Stanford dependency parser creating a dependency tree for the input sentence. We use the dependency tree as the input sentence's adjacency matrix and use GCN extracting regional dependency features. Graph convolutional network (GCN) was proposed by Kipf and Welling [13] in 2017. A GCN layer retrieves new node features by considering neighboring nodes' features with the following equation:

$$h_v^{l+1} = ReLU\left(\sum_{V \in N(v)} (w^l h_v^l + b^l)\right) \quad (5)$$

Where v is the target node and $N(v)$ represents the neighborhood of v, including v itself; h_v^l denotes the hidden feature of node v at layer l; W and b are learnable weights, mapping the feature of a node onto adjacent nodes in the graph.

After applying an L-layer GCN over word vectors, we obtain hidden representations of each token $S_G = \{g_1, g_2, \cdots, g_m\}$. To make use of these word representations for relation extraction, following Zhang et al [5], we first obtain a sentence representation as follows:

$$G_{sentence} = f(S_G) \quad (6)$$

Where f is a max pooling function that maps from n output vectors to the sentence vector, and $G_{sentence}$ is the sentence representation through the GCN network. Similarly. We can obtain the entity representations. For the head entity and the tail entity, their representation E_{head}、E_{tail} can be computed as:

$$E_x = f(g_x) \quad (7)$$

Where x is "head" or "tail", g_x represents the head entity representation (tail entity representation) after passing through the GCN.

We obtain the regional dependency feature representation by concatenating the sentence and the entity representations, and feeding them through a Multi-layer Perceptron (MLP):

$$h_{dependency} = MLP([G_{sentence}; E_{head}; E_{tail}]) \quad (8)$$

C. Semantic Features Extractor

In order to capture the Semantic Features of threat intelligence text, we use Sentence-BERT [1] to process the input threat intelligence text to obtain the semantic information of the text. At the same time, Sentence-BERT is used to alleviating one of the key obstacles, Out-Of-Vocabulary (OOV) problem, to processing threat intelligence texts. Sentence-BERT (SBERT), a modification of the pre-trained BERT network that use siamese and triplet network structures to deriving semantically meaningful sentence embeddings that can be compared using cosine-similarity. Sentence-BERT can be flexibly replaced with other language model such as RoBERTa [14].

$$\mathrm{B}_{sentence} = \mathrm{SBERT}(S) \quad (9)$$

We obtain the semantic features representation by feeding $\mathrm{B}_{sentence}$ through a Multi-layer Perceptron(MLP):

$$h_{\mathrm{semantic}} = MLP(\mathrm{B}_{sentence}) \quad (10)$$

D. Relation Classifier

After obtaining dependent feature representation and semantic feature representation, we obtain the syntactic-semantic representation used for classification by concatenating them. This syntactic-semantic representation is then fed into a linear layer followed by a softmax operation to obtain a probability distribution over relations.

V. Experiments

In this section, we present the experimental results of the proposed FSSRE. We first describe the experimental setup, the baselines we compare with, and experimental results. Finally, we also do experiments on the SemEval-2010 Task 8 dataset.

A. Experimental Setup

The BERT pre-training model we use is 768-dimensional BERT-Base-Cased. We concatenate together the last 4 hidden layers and sum them to get the final BERT word vector. We use the pre-trained 300-dimensional GloVe vectors to initialize word embeddings, and use embedding size of 30 for all other embeddings (i.e., POS, NER). We use the dependency parse trees, POS and NER sequences in our dataset (APTER-SENT). NER sequences are generated by manual annotation. POS sequences and the dependency parse trees were generated with Stanford Core NLP. We use 2 GCN layers and employ the ReLU function for all nonlinearities in the GCN layers and the standard max pooling operations in all pooling layers. In GCN model, we use the hidden layer of 200 as the output feed-forward layer. We set LSTM hidden size to 200. The APTER-SENT dataset contains many triples with the relation "Other", resulting in data imbalance. In order to mitigate the impact of data imbalance on the experimental results, we use the Undersampling method dividing the positive instances (the relation is pre-relation) and negative instances (the triple with the relation "Other") of the data. The ratio is adjusted to 1:1. We use cross-entropy as the loss function in Eq. (11). In the experiment, we tried the following four optimization functions: sgd, adagrad, adam, and adamax, which were verified by experiments: the adagrad optimization function was selected in the final model to achieve the best results.

$$f_{loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \log \hat{y}_{ij} \quad (11)$$

Where N represents the number of samples in a batch, M represents the number of relationship categories, $\hat{y}_{ij}$ represents the predicted value, y_{ij} represents the true value.

B. *Results on APTER-SENT dataset*

We compared FSSRE with two baselines:

- C-GCN [5]: A new pruning method keeps the nodes which are K hops away from the shortest path on the basis of the shortest dependent path. This model use LSTM+GCN network to extract text features for relation extraction.
- AGGCN [6]: This paper proposes a soft pruning method to automatically learn and selectively focus on related substructures for relation extraction.

As shown in Table V, for model C-GCN, We experimented with $K \in \{-1,0,1,2,\cdots,10\}$ on our dataset (APTER-SENT), where when the number of pruning $K = 2$, the F1 value is the largest. For the model AGGCN, We experimented on our dataset (APTER-SENT). FSSRE surpasses the AGGCN model by 1.585% (F1). Our model is an improvement based on the model C-GCN and surpasses the C-GCN model by 2.274%. For a fair comparison with the compared model, we removed the BERT word embedding from our model, namely FSSRE (-BERT), and the experimental results are still better than the compared model, e.g., surpassing the AGGCN model by 1.139% (F1). Although our model (FSSRE) precision is 0.25% lower than AGGCN, but recall increased by 3.525% and F1 increased by 1.585%.

TABLE V. RESULTS ON THE APTER-SENT DATASET

Model	Precision (%)	Recall (%)	F1 (%)
C-GCN (K=2) (Zhang et al., 2018)[5]	76.062	81.860	78.855
AGGCN (Guo et al., 2019)[6]	**79.363**	79.726	79.544
FSSRE(-BERT)	79.170	82.255	80.683
FSSRE (ours)	79.113	**83.251**	**81.129**

We examine the contributions of three components: BERT, SDP-VP and Sentence-BERT. We removed Sentence-BERT, SDP-VP and BERT from FSSRE model successively and cumulatively for ablation experiment. Table VI shows the results. We can observe that adding either SDP-VP or Sentence-BERT improves the performance of the model. This suggests that both layers can assist FSSRE to learn better information aggregations, where the SDP-VP seems to be playing a more significant role. And we find that: (1) The Sentence-BERT contribute F1 with 0.233%. (2) The SDP-VP layers improve F1 with 0.723%. (3) The BERT layers improve F1 with 1.318%.

TABLE VI. AN ABLATION STUDY FOR FSSRE MODEL ON THE APTER-SENT DATASET

Model	Precision (%)	Recall (%)	F1 (%)
FSSRE(ours)	79.113	83.251	81.129
- Sentence-BERT	78.880	83.018	80.896
- SDP-VP	77.379	83.177	80.173
- BERT	76.062	81.860	78.855

In order to show the effect of pruning number K and weight w on the experimental results in our pruning method, we experimented with $K \in \{-1,0,1,2,\cdots,10\}$ on our dataset (APTER-SENT), where $K = -1$ means input full tree. The experimental results are shown in Fig. 3. And we experimented with $w \in \{1.0,1.5,2.0,2.5,\cdots,5.5,6.0\}$ on APTER-SENT, the experimental results are shown in Fig. 4. It can be seen from Fig. 3 that when the pruning number $K = 1$, the F1 value is the largest. It can be concluded from Fig. 4 that the F1 value is the largest when the weight $w = 2.0$.

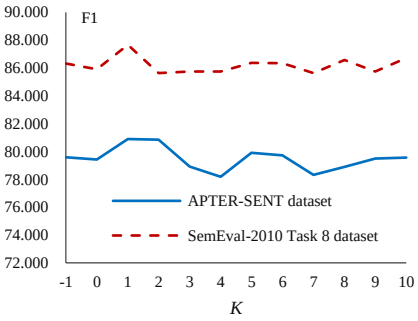

Fig. 3. Influence of pruning number K on experimental results ($w = 2$)

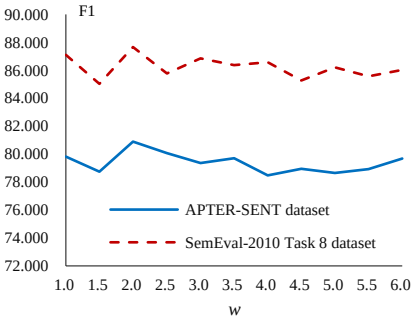

Fig. 4. Influence of weight w on experiment results ($K = 1$)

C. *Results on the SemEval-2010 Task 8 dataset*

We also conducted experiments on the general domain public dataset SemEval-2010 Task 8. We use Stanford CoreNLP to preprocess the SemEval-2010 Task 8 dataset to generate dependency parse trees, POS and NER annotations. The other experimental settings are consistent with APTER-SENT. The experimental results are shown in Table VII. For a fair comparison with the baseline model, we removed the BERT word embedding from our model, namely FSSRE (-BERT) which still surpasses the AGGCN model by 1.5% (F1). Compared with AGGCN, our model (FSSRE) improves by 2.2% (F1). Experimental results prove that our method is still effective on this dataset.

TABLE VII. RESULTS ON THE SEMEVAL-2010 TASK 8 DATASET

Model	F1 (%)
PA-LSTM (Zhang et al., 2017)[2]	82.7
C-GCN (Zhang et al., 2018)[5]	84.8
AGGCN (Guo et al., 2019)[6]	85.7
MVC(Veyseh et al, 2020)[15]	86.1
RbSP (Can et al., 2019)[7]	86.7
FSSRE(-BERT)	87.2
FSSRE (ours)	**87.9**

For the SemEval-2010 Task 8 dataset, we also examine the contributions of three components: BERT, SDP-VP and Sentence-BERT. Similarly, we removed Sentence-BERT, SDP-VP and BERT from FSSRE model successively and cumulatively for ablation experiment. Table VIII shows the results: (1) The Sentence-BERT contribute F1 with 0.221%. (2) The SDP-VP layers contribute F1 with 0.554%. (3) The BERT layers contribute F1 with 1.990%.

TABLE VIII. AN ABLATION STUDY FOR FSSRE MODEL ON THE SEMEVAL-2010 TASK 8 DATASET

Model	Precision (%)	Recall (%)	F1 (%)
FSSRE(ours)	87.957	87.811	87.884
- Sentence-BERT	86.393	88.972	87.663
- SDP-VP	85.475	88.806	87.109
- BERT	83.952	86.318	85.119

In order to show the influence of the pruning number K and weight w on the experimental results in our pruning method. The experimental results are shown in Fig. 3. From the Fig. 3, it can be concluded that when the pruning number K=1, the F1 value is the largest. And we experimented with $w \in \{1.0,1.5,2.0,2.5,\cdots,5.5,6.0\}$ on the SemEval-2010 Task 8 dataset, the experimental results are shown in Fig. 4, from which we can draw a conclusion: when the weight $w = 2.0$, the F1 value is the largest.

VI. CONCLUSIONS

In this paper, we propose a new dataset APTER-SENT, which contains annotations for cyber security-related entities and intra-sentence relations. We have presented FSSRE, a novel model of relation extraction between two entities in a sentence that can simultaneously contain contextual semantic features and syntactically dependent features. We use GCN to extract syntactically dependent features and use Sentence-BERT to extract contextual semantic features, and then fuse the two to extract relation. In order to obtain the most useful information, we propose a new pruning method SDP-VP. We evaluated our model on APTER-SENT and SemEval-2010 task 8 dataset, then compared the results with very recent state-of-the-art models. The results demonstrated the advantage and robustness of our model. We aim to improve this model so that it can extract inter-sentence relations in future works.

ACKNOWLEDGMENT

We are grateful to the anonymous reviewers for their valuable comments and suggestions to significantly improve the quality of this paper. This work is supported by the National Key Research and Development Program of China (Grant No. 2018YFC0824801).

REFERENCES

[1] Nils Reimers, Iryna Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China, 2019.

[2] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, Christopher D. Manning, "Position-aware Attention and Supervised Data Improve Slot Filling," In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 35–45. Association for Computational Linguistics, Copenhagen, Denmark, 2017.

[3] Patrick Verga, Emma Strubell, Andrew McCallum, "Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction," In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 872–884. Association for Computational Linguistics, New Orleans, Louisiana, 2018.

[4] Linfeng Song, Yue Zhang, Zhiguo Wang, Daniel Gildea, "N-ary Relation Extraction using Graph-State LSTM," In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2226–2235. Association for Computational Linguistics, Brussels, Belgium, 2018.

[5] Yuhao Zhang, Peng Qi, Christopher D. Manning, "Graph Convolution over Pruned Dependency Trees Improves Relation Extraction," In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2205–2215. Association for Computational Linguistics, Brussels, Belgium, 2018.

[6] Zhijiang Guo, Yan Zhang, Wei Lu, "Attention Guided Graph Convolutional Networks for Relation Extraction," In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 241–251. Association for Computational Linguistics, Florence, Italy, 2019.

[7] Duy-Cat Can, Hoang-Quynh Le, Quang-Thuy Ha, Nigel Collier, "A Richer-but-Smarter Shortest Dependency Path with Attentive Augmentation for Relation Extraction," In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2902–2912. Association for Computational Linguistics, Minneapolis, Minnesota, 2019.

[8] Aditya Pingle, Aritran Piplai, Sudip Mittal , et al, "RelExt: relation extraction using deep learning approaches for cybersecurity knowledge graph improvement," In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 879–886. Vancouver, Canada , 2019.

[9] Du M., Jiang J., Jiang Z., Lu Z., Du X. PRTIRG: A Knowledge Graph for People-Readable Threat Intelligence Recommendation. In: Douligeris C., Karagiannis D., Apostolou D. (eds) Knowledge Science, Engineering and Management. KSEM 2019. Lecture Notes in Computer Science, vol 11775. Springer, Cham. 2019.

[10] Xuren Wang, Jie Yang, Qiuyun Wang, Changxin Su. Threat Intelligence Relationship Extraction Based on Distant Supervision and Reinforcement Learning. In: The 32nd International Conference on Software Engineering and Knowledge Engineering, SEKE 2020, KSIR pp. 572-576. Virtual Conference Center, USA, 2020.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, 2019.

[12] Bojanowski P , Grave E , Joulin A , et al. "Enriching Word Vectors with Subword Information," Transactions of the Association for Computational Linguistics, 5:135-146, 2017.

[13] Thomas N Kipf and Max Welling, "Semi-supervised Classification with Graph Convolutional Networks." In: International Conference on Learning Representations. Toulon, France, 2017.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[15] Amir Veyseh, Franck Dernoncourt, My Thai, Dejing Dou, Thien Nguyen. Multi-View Consistency for Relation Extraction via Mutual Information and Structure Prediction.In Proc of AAAI. pp.9106-9113 Online, 2020.