electronics

MDPI

Article

An APT Event Extraction Method Based on BERT-BiGRU-CRF for APT Attack Detection

Ga Xiang *, Chen Shi and Yangsen Zhang

School of Information Management, Beijing Information Science and Technology University, Beijing 100192, China
* Correspondence: xiangga@bistu.edu.cn

Abstract: Advanced Persistent Threat (APT) seriously threatens a nation's cyberspace security. Current defense technologies are typically unable to detect it effectively since APT attack is complex and the signatures for detection are not clear. To enhance the understanding of APT attacks, in this paper, a novel approach for extracting APT attack events from web texts is proposed. First, the APT event types and event schema are defined. Secondly, an APT attack event extraction dataset in Chinese is constructed. Finally, an APT attack event extraction model based on the BERT-BiGRU-CRF architecture is proposed. Comparative experiments are conducted with ERNIE, BERT, and BERT-BiGRU-CRF models, and the results show that the APT attack event extraction model based on BERT-BiGRU-CRF achieves the highest F1 value, indicating the best extraction performance. Currently, there is seldom APT event extraction research, the work in this paper contributes a new method to Cyber Threat Intelligence (CTI) analysis. By considering the multi-stages, complexity of APT attacks, and the data source from huge credible web texts, the APT event extraction method enhances the understanding of APT attacks and is helpful to improve APT attack detection capabilities.

Keywords: network security; event extraction; deep learning; APT event; BERT-BiGRU-CRF

Citation: Xiang, G.; Shi, C.; Zhang, Y. An APT Event Extraction Method Based on BERT-BiGRU-CRF for APT Attack Detection. *Electronics* **2023**, *12*, 3349. https://doi.org/10.3390/electronics12153349

Academic Editor: Myung-Sup Kim

Received: 30 June 2023
Revised: 30 July 2023
Accepted: 2 August 2023
Published: 4 August 2023

1. Introduction

APT refers to the sustained and effective attack activities of an organization against specific objects. It is defined as attackers with complex technologies to create opportunities with rich resources to achieve their own purposes [1]. APT aims to attack infrastructure and steal sensitive intelligence and has a strong national strategic intention so that the network security threat has evolved from a random attack to a purposeful, organized, and premeditated group attack [2]. APT seriously threatens the nation's cyberspace security. In recent years, organized APT attacks continue to occur at a high rate [3]. APT attacks are rampant and frequent, so it is urgent to carry out more research to improve detection and defense technology.

Current APT defense solutions include detection based on APT attack life cycle, big data analysis, and dynamic behavior analysis [4–12]. Unfortunately, current defense technologies are unable to detect an APT attack accurately in time, since APT attacks are highly targeted, have strong concealment, and have a long duration. More work is needed to understand APT attack features for effective detection. Currently, the APT sample data are not sufficient and the features for detection are not clear.

Except for traditional attack detection methods, there is a new direction as CTI has appeared. Threat intelligence information is analyzed and shared to improve detection accuracy, shorten response time, and reduce defense costs. CTI research includes the following: (1) CTI sharing; there are some works on CTI sharing [13–15] whereby researchers have studied the CTI sharing framework and format. (2) CTI analysis; to analyze CTI automatically from huge sources, Information Extraction (IE) attracts researchers' interest naturally.

Knowledge Graphs (KG) and Indicators of Compromise (IOC) are extracted from unstructured CTI texts [16,17]. It should be noted that most CTI research is in English. More CTI research in Chinese is needed.

Regarding CTI analysis research, currently, there is seldom specific CTI research for APT attacks. CTI analysis for APT attacks will bring benefit to understanding APT attack features. This is definitely helpful for APT detection. From the previous investigation, it was observed that many reports and articles on APT-related vulnerabilities, security reports, event analysis, corresponding organizations, and attack alarms are published from authoritative network security technology centers, major manufacturers, research institutions, honker organizations, forums, etc. They are good data sources for APT CTI analysis. At the same time, sometimes organizations are alarmed that they will launch an APT to a specific field or affiliation at a specific time, even with some details described. In addition, the same APT attack sometimes can be launched at different times in different fields. Such important information is worth analyzing carefully to strengthen the APT detection ability. To collect big data and accelerate data analysis, it is imperative to study automatic information exaction methods.

This paper explores a new APT event extraction method based on deep learning with orienting APT Web texts in Chinese. We address the following objectives:

(1) An APT event schema is proposed based on analyzing APT attack stages. Event schemas are different in different fields. For APT events, it needs to define a proper schema to extract effective information.
(2) An APT event dataset in Chinese is constructed to train models. There is no APT event dataset although there are many event datasets. It is necessary to construct a corresponding dataset to train extraction models.
(3) An APT event extraction method based on the BERT-BiGRU-CRF model is proposed. This offers numerous advantages, which are helpful for solving the issues of insufficient attack sample data and low detection accuracy.

This research provides a novel CTI analysis method to extract APT events from credible web texts. The current CTI analysis is mainly about KG construction and IOC extraction. There is little CTI analysis of APT event extraction. Event extraction is proper to extract APT attack features, since event types are proper to express different APT attack stages, and rich event arguments are applicable to extract APT attack signatures. At the same time, this paper studies the APT event extraction from Chinese web texts. Most of the existing CIT analysis is in English. In the Chinese language, there is no blank space between words in a sentence. It needs to first cut words. The accuracy of cut words impacts downstream extraction tasks. In addition, Chinese word semantics are richer, and sentence structures are more complex than English ones.

The remainder of this paper is organized as follows: Section 2 describes related works. Section 3 details our proposal for APT event extraction. Section 4 reports the results of our experiments. Finally, the conclusion and discussion are presented in Section 5.

2. Related Works

2.1. APT Attack Detection Method

From the perspective of the APT attack detection method, the traditional solutions mainly focus on three aspects: (1) Detection based on the APT attack lifecycle. Yang [4] proposed a classification frame of APT attack behavior based on phased characteristics to fully understand APT attack behavior. In article [5], it is proposed that a classification and evaluation method of APT attack behavior is based on stage characteristics. (2) Detection based on big data analysis. Fu [6] analyzed four APT attack detection technologies based on big data analysis. Chen [7] analyzed large data processing technologies to solve the real-time restoration and analysis of high-performance network traffic. Wang [8] analyzed data on user access control, data isolation, data integrity, privacy protection, security audit, advanced persistent attack prevention, etc. (3) Detection based on dynamic behavior analysis. Sun [9] applied the method that runs virus samples in the sandbox or virtual machine

to analyze the dynamic behavior of the APT virus. Sun [10] proposed a new APT detection model by combining MapReduce and the support vector machine (SVM) algorithm to reduce calculation costs. Eslam [11] studied the dynamic Windows malicious code detection method based on context understanding analysis of API calls. Hamid [12] proposed a method of deep learning for static and dynamic malware detection. Zhang [18] proposed a mathematical backdoor model to summarize all kinds of backdoor attacks.

2.2. CTI Analysis

In addition to the above methods, in recent years CTI research appears which provides a new direction for carrying out the cyber-attack defense. CTI research mainly includes CTI sharing and analysis. CTI sharing studies the sharing format, standard, and framework [13–15,19,20]. As for CTI analysis, there are many types of research based on IE from Nature Language Processing (NLP). IE and textual data mining of open-source intelligence on the Web have become increasingly important topics in cyber security [16]. Liao [21] proposed iACE, a new solution for fully automated IOC extraction to obtain IP, MD5, and other IOC-like strings in the articles. Husari [22] developed automated and context-aware analytics of CTI to accurately learn attack patterns from commonly available CTI sources. Zhu [23] designed a network security knowledge ontology to construct KG from CTI sources. While inconsistencies exist in the constructed KG, Jo [16] studied semantic inconsistencies in finding methods. There is research on IOC extraction, malware KG construction, inconsistency checks, etc. While there is no specific APT-related CTI information extraction.

2.3. Event Extraction

For IE, it includes entity, entity relations, event, and event relation extractions. Event and relation extraction methods include the following: (1) Pattern matching, such as [24–27]. (2) Pattern matching and machine learning combination, such as [28–31]. (3) Deep learning, such as [32–36]. In [37], a tree-based neural network model is proposed to learn syntactic features automatically. The bidirectional recurrent neural networks described in [38] with a joint framework show good extraction performance. At the same time, event extraction data source and application fields are extended. Ritter presented a novel approach for discovering important event categories and classifying extracted events based on latent variable models from Twitter [39]. Lu studied event extraction in question-and-answer tasks and proposed a question–generation model to generate questions [40]. There are some IE works to unify the extraction model. The various IE predictions are unified into a linearized hierarchical expression under a GLM model [41]. There are many event extraction works but few for APT event extraction. Since APT is complex with multiple stages, it is meaningful to apply event extraction technology to describe the stages and features of APT attacks accurately.

To train extraction models, event extraction corpora and datasets are needed. There are many event corpora [42–45] but no APT-related event dataset.

In conclusion, it is interesting to extract APT events from CTI web texts based on deep learning technology. Considering APT defense's existing issues, namely, weak penetration protection, low detection accuracy, difficulty in obtaining evidence of attack range, and unknown new attack response, it is worth studying to extract information from unstructured APT-related texts, which can help understand APT attacks more completely. In this paper, based on current CTI analysis and event extraction technology, an APT event schema is proposed, an APT event dataset is constructed, and an APT event extraction method is proposed based on BERT-BiGRU-CRF.

3. Materials and Methods

The overview of the APT event extraction method is shown in Figure 1.

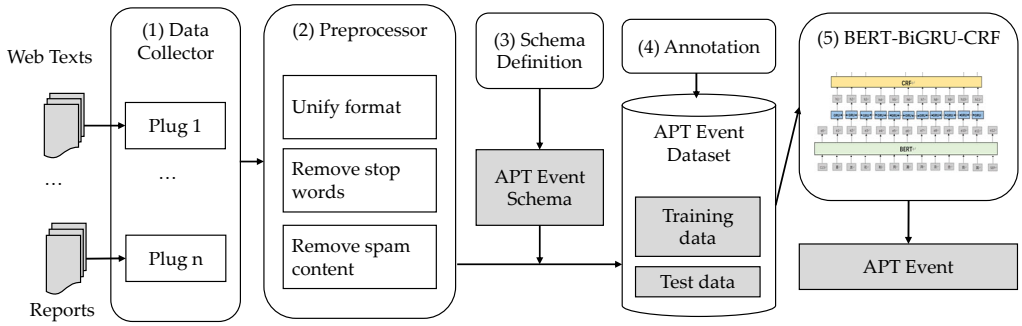

Figure 1. Overview of the APT event extraction.

It consists of a data collector, preprocessor, schema definition, annotation, and extraction model. (1) The data collector collects data from different sources, including web texts and reports. Different plugs are designed to handle multiple data sources. (2) Data are preprocessed to unify format and remove stop words, etc. (3) APT event types and schema are defined. (4) According to the schema, the APT event dataset is constructed for further model training. (5) The BERT-BiGRU-CRF model is trained to extract APT events.

3.1. Data Source and Preprocess

At first, corresponding data are collected and preprocessed. The key point is to find credible data sources for APT texts. As shown in Table 1, they are some credible websites that can provide potential APT information sources.

Table 1. APT information sources.

Types of Web Sites	Detail Information
Authoritative network security technology center	https://www.cert.org.cn/ * https://www.cnvd.org.cn/ * https://cve.mitre.org/ https://nvd.nist.gov/ https://www.cvedetails.com/
Major manufacturers	https://www.oracle.com/security-alerts/ https://msrc.microsoft.com/update-guide/ *
Research institutions	https://www.kaspersky.com.cn/ * https://www.nsfocus.com.cn/ * https://www.qianxin.com/ *
Forum	honker or hacker organizations and forums
APT dataset	https://github.com/cyber-research/APTMalware

Note: The links with * mean that the web texts are in Chinese.

The APT data collector is designed to acquire APT-related vulnerabilities, security reports, event analysis, corresponding organizations, alarms, etc. A distributed webpage crawler system based on the Scrapy framework is designed and implemented. The following rules are considered: (1) A distributed structure is used to improve crawling speed by easily adding hosts. (2) Modularization architecture is used for improved scalability. When adding a new target website, it can focus on creating a specific code for the website's crawling, parsing, and loading rules, while no big change is required for the common module. (3) It shall be easy to deploy. (4) There is real-time monitoring. (5) It has high performance.

Data preprocessing is necessary. It includes the following steps: (1) Remove the html label such as <a, <font, etc. Such labels are filtered and real content is obtained. (2) Download a file if a downloadable file is found. (3) Cut words and remove stop words. A word dictionary and stop word table are built for APT texts. The APT dictionary includes huge

network security words of APT attacks, new APT technologies, related affiliations, addresses, and period reference pronouns. At the same time, a customized stop word table is created for APT attacks. These are intended to improve word cutting accuracy. (4) Remove the spam content, such as advertisements.

The acquisition of APT attack texts mainly adopts crawler technology, using the requests library in Python to request page data from web pages. For example, for the Qi'anxin Threat Intelligence Center, the BeautifulSoup library is used to complete the parsing of page data. After downloading, each article is named with the title of each event, and the content includes the title, time, and description of the APT attack. Afterward, further processing is carried out on the crawled information, such as removing duplicate content. For some websites, an anti-crawler mechanism exists, so a manual copy is used.

3.2. APT Attack Stages and Event Schema

The APT attack has a complex long duration and is hard to detect. There are some existing works that studied the APT stages. In [46], it described an APT attack tree model, including reconnaissance, establishing a foothold, lateral movement, exfiltration or impediment, and post-exfiltration or post-impediment stages. In [47–49], the APT lifecycle is divided into three stages: attack prelude, intrusion implementation, and subsequent attacks.

According to the APT lifecycle, when defining the event schema for APT, it is necessary to consider APT stages. We define the APT event categories, which match the stages. The lifecycle of an APT attack includes three stages, as shown in Figure 2. In different stages, it has different key features or signatures, which can be defined as APT event arguments.

Figure 2. The lifecycle of APT attacks.

In our research, to simplify the problem, we focus on the stages of preparation before attack and implementation. By analyzing the stages, we define the schema: (1) Define two event categories: preparation and implementation. (2) In each category, according to typical APT attack types, we define nine APT event types. (3) For each APT event type, we define the corresponding arguments. The schema of the APT event is defined below in Table 2.

Table 2. APT categories, event types, and arguments.

NO.	Event Category	Event Type	Argument Role1	Argument Role2	Argument Role3	Argument Role4	Argument Role5
1	Preparation	Spear phishing attack	Fake file	True file	Attacker	Target	Attack tactics
2		Water hole attack	Fake file	True file	Attack weapon		
3		Scan	Target				
4		Steal information	Attacker	Target	Stolen target	Attack weapon	
5	Implementation	Trojan	Attacker	Target	Attack weapon	Attack tactics	
6		Worm	Attacker	Target	Attack weapon		
7		Back door	Attacker	Target	Attack weapon		
8		Virus	Attacker	Target	Attack weapon	Attack tactics	
9		Vulnerability exploitation	Attacker	Target	Attack weapon	Attack tactics	

3.3. APT Dataset Construction

At present, there are many event datasets, but unfortunately, there is no existing event dataset for APT events. To train the model, it needs to construct an APT event dataset. Referring to the annotation method of DuEE1.0 (Chinese event extraction dataset) from Baidu released in 2020, we annotated APT event samples of an APT dataset. This annotation method is beneficial to define different event types and the flexibility of the corresponding arguments.

An example template of annotation for a single APT attack event is shown in Figure 3. The annotated events are saved in the data exchange format JSON, which is not only convenient for conversion but also easy to read.

```
{
  "text": " ",
  "event_list": [{
         "event_type":" ",
         "trigger":" ",
         "trigger_start_index": ,
         "arguments":[{
                    "argument_start_index": ,
                    "role":" ",
                    "argument":" ",
                    "alias":[]
          },{
                    "argument_start_index": ,
                    "role":" ",
                    "argument":" ",
                    "alias":[]
       }],
       "class":" "
  },{
         "event_type":" ",
         "trigger":" ",
         "trigger_start_index":,
         "arguments":[ ],
         "class":" "
        }]
 }
```

Figure 3. Example template of one APT attack event annotation.

The annotated events are saved in a JSON tree structure. There are many indentations, line breaks, and spaces that take up a lot of space. To save space, we save each annotated event as a single line. Therefore, when annotating events and writing them into a JSON file, the JSON is compressed by setting the attributes (setting the dump() function's indent = 4, separators = (',', ':')). Each line of the generated JSON file is the extraction result of one event, and the new line is another event. As in Figure 4.

```
{"text": "近日，红雨滴团队研究人员对国外厂商披露为海莲花的样本进行了深入
{"text": "近日，红雨滴团队研究人员在日常威胁狩猎中再次捕获到一例针对Linu
{"text": "MuddyWater组织的攻击通常始于向组织发送有针对性的电子邮件，然
{"text": "近日，奇安信威胁情报中心红雨滴在日常的威胁狩猎捕获一起 Donot
{"text": "双尾蝎组织具有Windows和Android双平台攻击武器，且仅Windows平
{"text": "近日，奇安信威胁情报中心在日常样本分析研判中捕获到多个以印度国
{"text": "近日，奇安信红雨滴团队在日常样本狩猎过程中捕获到一批使用了与海
{"text": "近日，奇安信威胁情报中心红雨滴团队在日常的威胁狩猎中捕获了该组
{"text": "奇安信威胁情报中心在日常威胁发现过程中发现一个专门针对贸易行业
{"text": "近日，奇安信威胁情报中心红雨滴团队在日常的威胁狩猎中捕获了该组
{"text": "近日，奇安信威胁情报中心红雨滴在日常的威胁狩猎捕获一起Donot A
{"text": "2021年11月11日，奇安信威胁情报中心红雨滴团队披露了SideCopy组
```

Figure 4. APT attack event dataset.

Finally, the APT event dataset is constructed, resulting in a total of 130 event information types. Although the size is not big, it covers the main APT attack stages, attack types, etc. It is divided into training sets, validation sets, and testing sets in a ratio of 8:1:1.

3.4. APT Attack Event Extraction Based on BERT-BiGRU-CRF

The event extraction tasks for an APT attack include the identification of event types according to the defined APT event type and arguments schema, and the extraction of all related arguments. After our investigation and large experiments, the BERT-BiGRU-CRF model is constructed to extract APT events and shows good performance. The overview of the APT event extraction model based on BERT-BiGRU-CRF is shown in Figure 5 as follows:

(1) BERT layer. At first, the BERT model is applied to pre-train word vectors. The BERT encoding layer is located at the bottom of the model. In the encoding layer, tokens are

segmented from the input of APT texts, and the segmented tokens are transformed into corresponding word vectors by extracting the semantic feature.

(2) BiGRU layer. Secondly, it connects with BiGRU to carry out the APT trigger word and event argument extraction. The pre-trained word vector is fed into the BiGRU layer, which will continue to extract its features and obtain the emission matrix of its sequence. The final output is the predicted label (APT-related trigger word or arguments defined in the schema) corresponding to each word.

(3) CRF layer. The obtained result is then constrained by the CRF layer and its transfer matrix is obtained. Ultimately, the optimal label sequence is output.

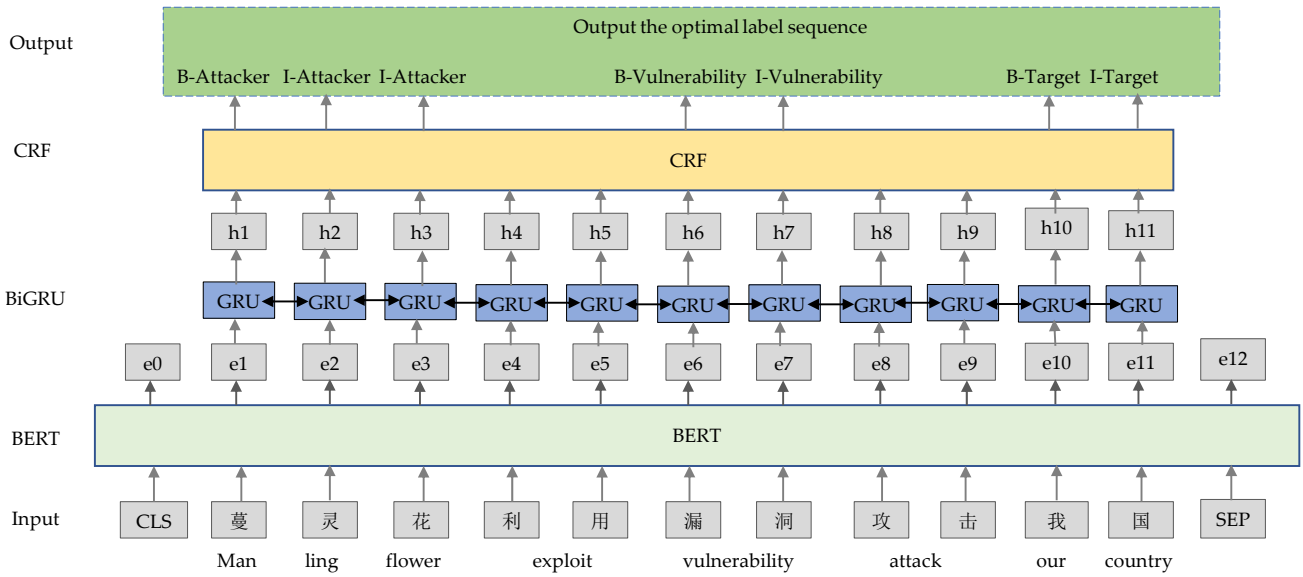

Figure 5. Overview of the APT event extraction model based on BERT-BiGRU-CRF.

3.4.1. BERT Pre-Training Layer

To improve extraction performance, BERT is applied as a pre-training model.

As shown in Figure 6, using the text "蔓灵花利用漏洞 (Manling flower exploits vulnerability)" as an example, the input text is first cut into single Chinese words, and the CLS mark is added at the beginning of the sentence, and the SEP mark is added at the end. Then, through multi-layer transformers, the vectors of word, clause, and position are obtained, and they are integrated together. Finally, it serves as the input vector for the BiGRU layer.

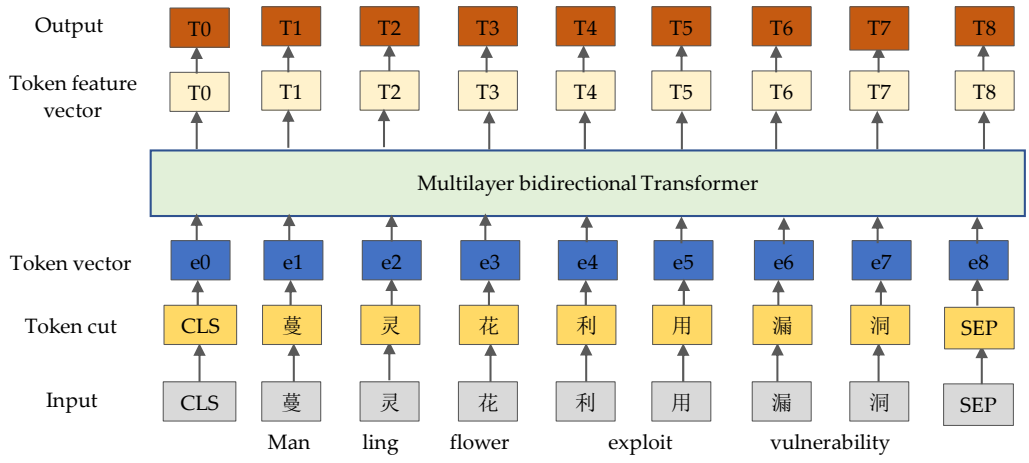

Figure 6. BERT layer structure.

3.4.2. BiGRU Layer

The input of the BiGRU layer is a word vector pre-trained by the BERT layer, and the output is the score of the predicted label corresponding to each word (as shown in Figure 7).

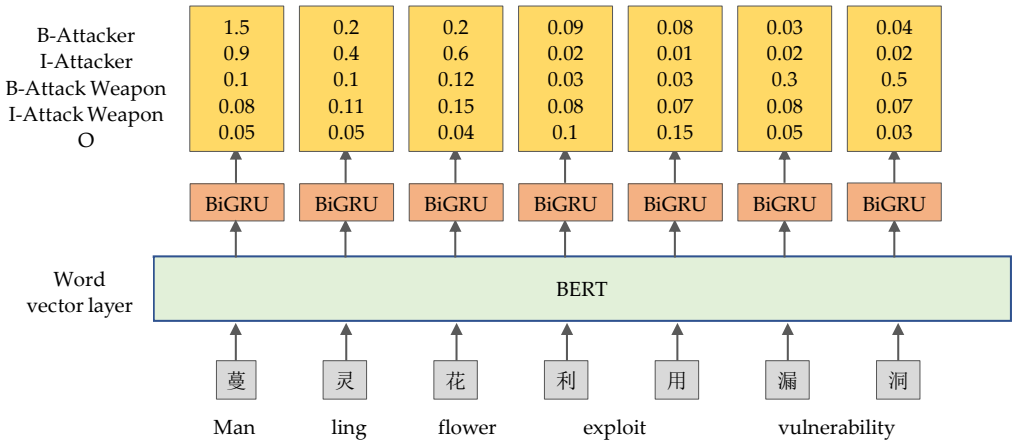

Figure 7. Output of BiGRU layer-1.

The output of the BiGRU layer is also known as the emission matrix. It consists of emission scores. Each score represents the value of each label corresponding to the character. Using the word "蔓 (Man)" as an example, the outputs through the BiGRU level are 1.5 (B-Attacker), 0.9 (I-Attacker), 0.1 (B-Attack Weapon), 0.08 (I-Attack Weapon), and 0.05 (O). These numbers are the scores given to the word "蔓" based on each label. That is, for the word "蔓", its score of the label "B-Attacker" is 1.5 which is the highest one, and the score of the label "I-attacker" is 0.9, and so on. The higher the score, the greater the likelihood of representing this category. The character "蔓" has the highest score in the "B-Attacker" category, so the word "蔓" is temporarily labeled as "B-Attacker". The matrix that combines the emission scores of each word together is called the emission matrix, which will also serve as the input to the CRF layer.

3.4.3. CRF Layer

Even without the CRF layer, we can still train an event extraction model based on BERT-BiGRU, because the BiGRU model provides scores for each label corresponding to each word. We can choose the label with the highest score (marked in red) as the prediction result. For example, if the character "灵 (Ling)" has the highest score of "I-attacker" (0.4), then we can choose "I-attacker" as the prediction result. However, the actual situation may result in the following predicted results (as shown in Figure 8).

The CRF layer can add some constraints to ensure the effectiveness of the final prediction result. The constraints can be automatically learned by the CRF layer during data training. Possible constraints include the following:

(1) The beginning of the sentence should be "B-" or "O", not "I-"; as shown in Figure 8, the sentence cannot start with "I-Attack Weapon".
(2) B-lablel1 I-label2 I-label3… "In this case, categories 1, 2, and 3 should be the same entity category." For example, "B-attacker I-attacker" is correct, while "B-attacker I-attack weapon" is incorrect.
(3) "O I-Attack Weapon" is incorrect, the beginning of the named entity should be "B-" instead of "I-".

With the above useful constraints, erroneous prediction sequences will be greatly reduced. The CRF layer mainly utilizes a transition matrix to ensure these constraints. The transition score is the score transferred from one label to another label, as shown in Table 3.

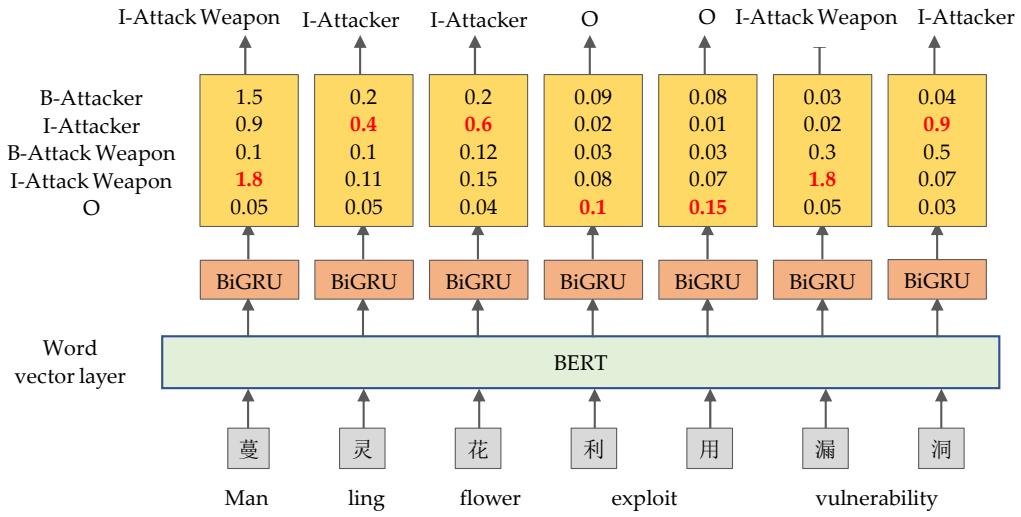

Figure 8. Output of BiGRU Layer-2.

Table 3. Transition matrix.

Transition Matrix	0	B-Attacker	I-Attacker	B-Attack Weapon	I-Attack Weapon
0	0.8	0.07	0	0.12	0
B-Attacker	0	0	1	0	0
I-Attacker	0.18	0	0.85	0	0
B-Attack Weapon	0	0	0	0	1
I-Attack Weapon	1	0	0	0	0

Using the third row and third column in Table 3 as an example, 0.85 represents the score for transitioning from the label "I-Attacker" to the label "I-Attacker".

Finally, combining the emission matrix obtained from the BiGRU layer and the transition matrix obtained from the CRF layer, we can calculate the tag path with the highest score, as shown in Figure 9.

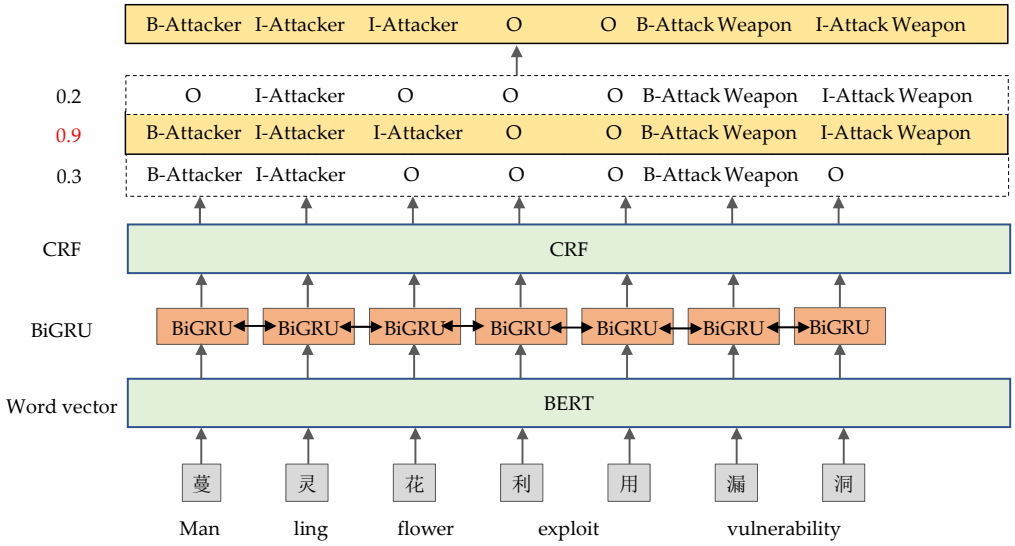

Figure 9. The final output of the BERT-BiGRU-CRF model.

The tag path of "B-Attacker I-Attacker I-Attacker O O B-AttackWeapon I-Attack Weapon" has a score of 0.9 (marked in red), which is the highest score. Therefore, this path is the final output.

4. Experimental Results

4.1. Model Construction and Training

When implementing the models of the APT event extraction described in Section 3, the deep learning framework of Baidu PaddlePaddle is applied. It integrates the functions of model training, inference framework, and basic model library. BERT-BiGRU-CRF models are constructed based on the PaddlePaddle. The specific training parameters are shown in Table 4.

Table 4. Specific training parameters.

Parameter Name	Values
num_epoch(training rounds)	60
learnin_rate(learning_rate)	5×10^{-5}
weight_decay(weight decay)	0.01
warmup_proportion(warmup proportion)	0.1
gru_hidden_size(gru hidden size)	300

4.2. Experimental Results

4.2.1. Comparison with Other Models

The evaluation indicators will use the following three indicators as references:

1. Precision = number of correct predictions with "Positive"/number of predictions with "Positive", mainly focusing on the accuracy of the results predicted by the model. The formula is as shown below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

For TP, FP, etc., the meanings are as shown in Table 5.

2. Recall = number of correctly predicted items with "Positive"/number of manually annotated items with "Positive", mainly focusing on what the model missed. The formula is as shown below:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

3. F1 = 2 × Precision × Recall/(Precision + Recall), the formula is calculated as follows:

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Table 5. Confusion matrix.

True/False Examples	Prediction	
	Positive	Negative
True	TP	FN
False	FP	TN

We compared several models with BERT-BiGRU-CRF to extract APT events. The results are as shown in Table 6.

From Table 6, for APT trigger word detection, it can be seen that the F1 values are 1.00 for ERNIE and BERT, and 0.9951 for BiGRU-CRF. The F1 values for trigger word extraction are all very high. This is because the trigger words of APT attacks are not huge, or relatively concentrated, so models show high precision and recall performance to identify APT trigger words.

Table 6. Comparison of experimental results.

Model	Trigger Word Detection			APT Event Argument Recognition		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
ERNIE	**1.00**	**1.00**	**1.00**	0.5859	0.8189	0.6831
BERT	**1.00**	**1.00**	**1.00**	0.5812	0.8813	0.7004
BiGRU-CRF	0.9903	1.00	0.9951	0.5211	**0.8462**	0.6451
BERT-BiGRU-CRF	**1.00**	**1.00**	**1.00**	**0.7013**	0.8011	**0.7479**

The APT event argument recognition is harder than the APT trigger word detection. It can be seen that the F1 value of BERT-BiGRU-CRF is 0.7479, which is better than the F1 values of BERT (0.7004) and BiGRU-CRF (0.6451). From Table 6, without BERT as a pre-training model, the final extraction F1 value can be seen as lower by about 10% than our proposed model.

From our experiment, it is found that pre-training improves the final extraction performance for APT events. ERNIE and BERT can both be used as the pre-training model. For our APT event extraction, BERT pre-training shows better performance. Therefore, we ultimately used the BERT model for the pre-training of word vectors, and then connected it to the BiGRU-CRF model.

Firstly, we applied BERT as a pre-training model. The BERT model can effectively learn the underlying information of the sequence, and if the data volume is small, it is also recommended to pre-train word vectors.

Secondly, it connected with the BiGRU model, which learned the sequence information well, resulting in better learning of the APT semantics. The BiGRU model can effectively solve the problem of long sentences, enabling better learning of deep semantics, and ultimately using the CRF model for constraints.

At last, we applied CRF to carry out constraints to improve accuracy.

In summary, for APT attack events, according to the experimental results, the BERT-BiGRU-CRF model has the best extraction effect and the highest event extraction efficiency.

4.2.2. Performance Analysis of BERT-BiGRU-CRF Model for APT Attack Event Extraction

For the BERT-BiGRU-CRF model, during training for APT attack event extraction, the corresponding F1 values are shown in Figure 10, showing the F1 change trend during the training process.

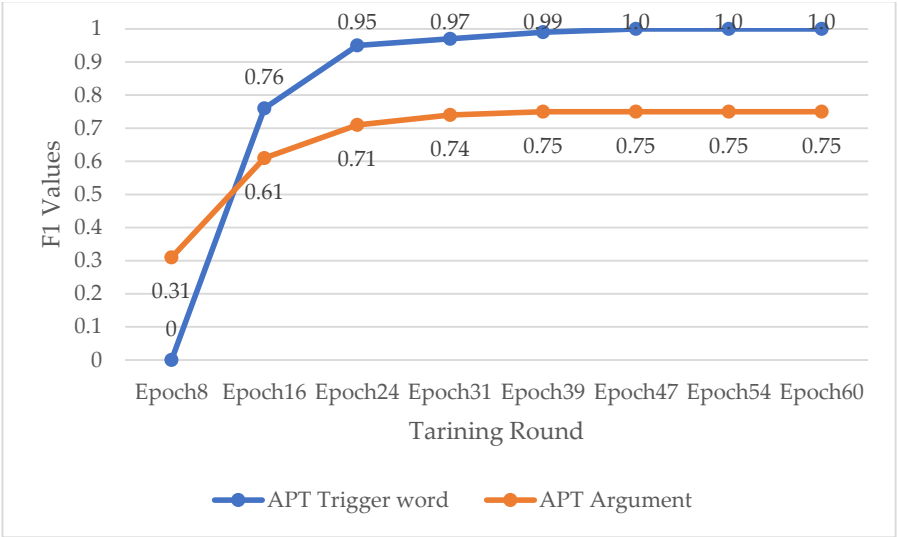

Figure 10. F1 values of BERT-BiGRU-CRF training.

As shown in Figure 10, the F1 value of the trigger words significantly increased at Epoch16, reaching a peak of 1.0 at Epoch16. The argument character also showed a significant increase in F1 value at Epoch16, ultimately reaching a peak of 0.75 at Epoch39.

4.2.3. Case Study

After the model training is finished, it can be used to carry out event extraction. The process of APT extraction is shown in Figure 11.

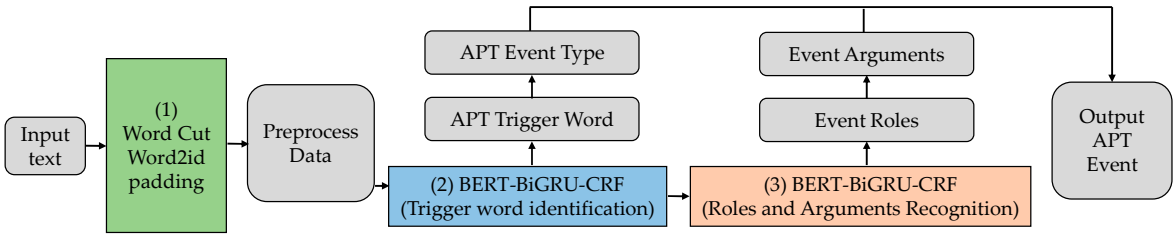

Figure 11. APT event extraction based on BERT-BiGRU-CRF model.

Using a case for example, the input text is as follows:

"南亚次大陆地区的响尾蛇组织被发现利用CVE-2019-2215漏洞针对安卓终端目标用户实施移动端的APT攻击。" (Translation: it is found that the Rattlesnake organization in the subcontinent of South Asia implemented mobile APT attacks against target users of Android terminal exploiting the CVE-2019 2215 vulnerability.)

(1) The input data were preprocessed including word cut, word2id, long text cut, and short text padding.
(2) The preprocessed data were input to the first BERT-BiGRU-CRF model to extract the trigger word. In this case, the trigger word is "漏洞利用" ("exploit vulnerability"), and the corresponding event type is "攻击实施-漏洞利用" ("Attack implementation-Vulnerability exploitation").
(3) According to the APT event type, the event roles are decided. Data are input to the second BERT-BiGRU-CRF model to extract the corresponding arguments.

All the outputs are merged to generate the final APT event for this text, as shown below:

event0–event_type: 攻击实施-漏洞利用 (attack implementation–vulnerability exploitation), trigger: 漏洞利用 (exploit vulnerability)

role_type: 攻击者 (Attacker), argument: 响尾蛇组织 (Rattlesnake organization)

role_type: 攻击武器 (Attack weapon), argument: CVE-2019 2215漏洞 (vulnerability)

role_type: 受害目标 (Target), argument: 安卓终端目标用户 (target users of Android terminal).

5. Conclusions and Discussion

This paper defines APT event types and templates, constructs an APT attack event extraction dataset, and builds an APT attack event extraction model based on BERT-BiGRU-CRF. Through comparative experiments with ERNIE, BERT, and BiGRU-CRF models, it was found that the APT attack event extraction model based on BERT-BiGRU-CRF had the highest F1 value, with an F1 value of 1.00 for trigger word extraction and 0.75 for argument role extraction, indicating the best extraction effect. This model first uses the BERT model to pre-train word vectors, then connects the BiGRU model for feature extraction, and then connects the CRF model for constraints, ultimately completing event extraction.

Considering there is little APT event extraction research, the work in this paper is a valuable contribution to CTI analysis and APT detection. It proposes a novel CTI analysis method by extracting APT events from Chinese web texts. During the APT event schema design, it considers the APT multi-stages and complexity, which is good for deeply understanding APT attacks. This is beneficial to improve APT attack detection ability.

There is a limitation to this research. For dataset construction, although there are some event datasets, unfortunately, there is no APT event dataset. During the research, we constructed the APT attack event dataset. The annotation cost is not low. Limited by the annotation resources, the annotation dataset is not big. This causes the following: (1) Event type numbers are not balanced. For example, much threat intelligence started with the spear-phishing attack as the starting point, the constructed dataset has a high proportion of "Attack preparation" and "Spear-phishing attack" events. (2) The trigger words related to APT are not completely included, indirectly resulting in some complex APT attack event information not being able to be extracted well.

Although the extraction method is effective, the potential weak point is that the model is a pipeline model, which separates trigger word extraction from argument role extraction. For APT attack events, there is some correlation between trigger words and argument roles, so using a joint extraction model may be worth studying further.

To remove the limitation and weak point, the next steps include the following: (1) Expand the data sources, obtain more unstructured information related to APT attack events on multiple websites, and make the constructed dataset contain complete event types and APT-related trigger words. (2) Consider applying the few-shot learning method to mitigate the data sparse issue. (3) Improve the use of a joint model for extraction.

Author Contributions: Conceptualization, G.X.; funding acquisition, G.X. and Y.Z.; methodology, G.X. and C.S.; software, C.S.; validation, G.X. and C.S.; writing—original draft, G.X. and C.S.; writing—review and editing, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the R&D Program of Beijing Municipal Education Commission, grant number KM202311232014, and the National Natural Science Foundation of China, grant number 62176023.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. National Institute of Standards and Technology. *SP800—53 Managing Information Security Risks*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2013.
2. Zhang, Y.; Pan, X.; Liu, Q.; Cao, J.; Luo, Z. APT attacks and defenses. *J. Tsinghua Univ. (Sci. Technol.)* **2017**, *57*, 1127–1133.
3. Chinese CNCERT. 2020 China Cybersecurity Analysis, [EB/OL]. (2021-5-26) [2021-6-4]. Available online: https://www.cert.org.cn/publish/main/upload/File/2020%20CNCERT%20Cybersecurity%20Analysis.pdf (accessed on 1 May 2023).
4. Yang, H. Research on APT Attack of Behavior Analyzing and Defense Decision. Master's Thesis, Information Engineering University, Zhengzhou, China, 2017.
5. Yang, H.; Wang, K. Phase-based classification and evaluation of APT attack behaviors. *Comput. Eng. Appl.* **2017**, *53*, 97–104, 234.
6. Fu, Y.; Li, H.; Wu, X.; Wang, J. Detecting APT attacks: A survey from the perspective of big data analysis. *J. Commun.* **2015**, *36*, 1–14.
7. Chen, X.; Zeng, X.; Wang, W.; Shao, G. Big Data Analytics for Network Security and Intelligence. *Adv. Eng. Sci.* **2017**, *39*, 112–129.
8. Wang, D.; Zhao, W.; Ding, Z. Review of Big Data Security Critical Technologies. *J. Beijing Univ. Technol.* **2017**, *43*, 335–349.
9. Sun, L. Research on Key Technology of APT Detection Based on Malicious Domain Name. Master's Thesis, Harbin Engineering University, Harbin, China, 2017.
10. Sun, J.; Wang, C. Research on APT attack detection based on behavior analysis. *Electron. Des. Eng.* **2019**, *27*, 142–146.
11. Eslam, A.; Ivan, Z. A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence. *Comput. Secur.* **2020**, *92*, 101760. [CrossRef]
12. Hamid, D.; Sajad, H.; Ali, D.; Sattar, H.; Hadis, K.; Reza, P.; Raymond, C. Detecting Cryptomining Malware: A Deep Learning Approach for Static and Dynamic Analysis. *J. Grid Comput.* **2020**, *18*, 293–303. [CrossRef]
13. Yang, P.; Wu, Y.; Su, L.; Liu, B. Overview of Threat Intelligence Sharing Technologies in Cyberspace. *Comput. Sci.* **2018**, *45*, 9–18, 26.
14. Ramsdale, A.; Shiaeles, S.; Kolokotronis, N. A Comparative Analysis of Cyber-Threat Intelligence Sources, Formats and Languages. *Electronics* **2020**, *9*, 824. [CrossRef]
15. Lin, Y.; Liu, P.; Wang, H.; Wang, W.; Zhang, Y. Overview of Threat Intelligence Sharing and Exchange in Cybersecurity. *J. Comput. Res. Dev.* **2020**, *57*, 2052–2065.

16. Jo, H.; Kim, J.; Porras, P.; Yegneswaran, V.; Shin, S. GapFinder: Finding Inconsistency of Security Information from Unstructured Text. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 86–99. [CrossRef]
17. Christian, R.; Dutta, S.; Park, Y.; Rastogi, N. An Ontology-driven Knowledge Graph for Android Malware. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 15–19 November 2021; pp. 2435–2437.
18. Zhang, Q.; Ma, W.; Wang, Y.; Zhang, Y.; Shi, Z.; Li, Y. Backdoor Attacks on Image Classification Models in Deep Neural Networks. *Chin. J. Electron.* **2022**, *31*, 199–212. [CrossRef]
19. Li, Y.; He, J.; Li, J.; Yu, Y.; Tan, F. US Cyber Threat Intelligence Sharing Technology Analysis of Framework and Standards. *Secrecy Sci. Technol.* **2016**, *6*, 16–21.
20. Wagner, T.; Mahbub, K.; Palomar, E.; Abdallah, A. Cyber threat intelligence sharing: Survey and research directions. *Comput. Secur.* **2019**, *87*, 10158. [CrossRef]
21. 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, Vienna, Austria, 24–28 October 2016; pp. 755–766.
22. 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, Orlando, FL, USA, 4–8 December 2017; pp. 103–115.
23. Shang, W.; Zhu, P.; Wang, B.; Cao, Z.; Zhang, M. Key Technologies for Building Knowledge Graphs for Threat Intelligence. *Autom. Panor.* **2023**, *40*, 15–19.
24. Khoo, C.S.; Kornfilt, J.; Oddy, R.N.; Myaeng, S.H. Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. *Lit. Linguist. Comput.* **1998**, *13*, 177–186. [CrossRef]
25. Khoo, C.S.; Chan, S.; Niu, Y. Extracting causal knowledge from a medical database using graphical patterns. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, 1–8 October 2000; pp. 336–343.
26. Hashimoto, C.; Torisawa, K.; De Saeger, S.; Oh, J.H. Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju, Korea, 12–14 July 2012; pp. 619–630.
27. Sadek, J.; Meziane, F. Extracting Arabic Causal Relations Using Linguistic Patterns. *ACM Trans. Asian Lang. Inf. Process.* **2016**, *15*, 14. [CrossRef]
28. Girju, R. Automatic detection of causal relations for question answering. In Proceedings of the ACL 2003 workshop on Multilingual Summarization and Question Answering, Sapporo, Japan, 11–12 July 2003; pp. 76–83.
29. Blanco, E.; Castell, N.; Moldovan, D. Causal relation extraction. In Proceedings of the Sixth International Conference on Language Resources and Evaluation, Marrakech, Morocco, 26 May–1 June 2008.
30. Wang, H.; Shi, Y.; Zhou, X.; Zhou, Q.; Shao, S.; Bouguettaya, A. Web service classification using support vector machine. In Proceedings of the 2010 22nd IEEE International Conference on Tools with Artificial Intelligence, Arras, France, 27–29 October 2010; Volume 1, pp. 3–6.
31. Zhao, S.; Liu, T.; Zhao, S.; Chen, Y.; Nie, J.Y. Event causality extraction based on connectives analysis. *Neurocomputing* **2016**, *173*, 1943–1950. [CrossRef]
32. De Silva, T.N.; Zhibo, X.; Rui, Z.; Kezhi, M. Causal relation identification using convolutional neural networks and knowledge based features. *Int. J. Comput. Syst. Eng.* **2017**, *11*, 696–701.
33. Jin, G.; Zhou, J.; Qu, W.; Long, Y.; Gu, Y. Exploiting Rich Event Representation to Improve Event Causality Recognition. *Intell. Autom. Soft Comput.* **2021**, *30*, 161–173. [CrossRef]
34. Gao, J.; Luo, X.; Wang, H. Chinese causal event extraction using causality-associated graph neural network. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6572. [CrossRef]
35. Xu, J.; Zuo, W.; Liang, S.; Wang, Y. Causal Relation Extraction Based on Graph Attention Networks. *J. Comput. Res. Dev.* **2020**, *57*, 16.
36. Tan, Y.; Peng, H.; Qin, J.; Xue, Y. Chinese causality analysis based on weight calculation. *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* **2022**, *50*, 112–117. [CrossRef]
37. Fei, H.; Ren, Y.; Ji, D. A tree-based neural network model for biomedical event trigger detection. *Inf. Sci.* **2020**, *512*, 175–185. [CrossRef]
38. Nguyen, T.; Cho, K.; Grishman, R. Joint event extraction via recurrent neural networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 300–309.
39. Ritter, A.; Mausam; Etzioni, O.W.; Clark, S. Open domain event extraction from twitter. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 1104–1112.
40. Lu, D.; Ran, S.; Tetreault, J.; Jaimes, A. Event Extraction as Question Generation and Answering. *arXiv* **2023**, arXiv:2307.05567.
41. Fei, H.; Wu, S.; Li, J.; Li, B.; Li, F.; Qin, L.; Zhang, M.; Zhang, M.; Chua, T. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 15460–15475.
42. Pustejovsky, J.; Hanks, P.; Sauri, R.; See, A.; Gaizauskas, R.; Setzer, A.; Radev, D.; Sundheim, B.; Day, D.; Ferro, L. The timebank corpus. *Corpus Linguist.* **2003**, *2003*, 40.

43. Doddington, G.R.; Mitchell, A.; Przybocki, M.A.; Ramshaw, L.A.; Strassel, S.; Weischedel, R.M. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. *LREC* **2004**, *2*, 837–840.
44. Wang, X. Event-Oriented Text Knowledge Discovery and Representation. Ph.D. Thesis, Shanghai University, Shanghai, China, 2017. Available online: https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CDFD0911&filename=2010252946.nh (accessed on 15 May 2023).
45. Mariko, D.; Akl, H.A.; Labidurie, E.; Mazancourt, H.; El-Haj, M. Financial document causality detection shared task. *arXiv* **2020**, arXiv:2012.02505.
46. Drury, B.; Gonçalo Oliveira, H.; De Andrade Lopes, A. A survey of the extraction and applications of causal relations. *Nat. Lang. Eng.* **2020**, *28*, 361–400. [CrossRef]
47. Alshamrani, A.; Myneni, S.; Chowdhary, A.; Huang, D. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1851–1877. [CrossRef]
48. Auty, M. Anatomy of an advanced persistent threat. *Netw. Secur.* **2015**, *2015*, 13–16. [CrossRef]
49. Chen, P.; Desmet, L.; Huygens, C. A study on advanced persistent threats. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8735, pp. 63–72. [CrossRef]