electronics

MDPI

Article

Effective Event Extraction Method via Enhanced Graph Convolutional Network Indication with Hierarchical Argument Selection Strategy

Zheng Liu [1,2], Yimeng Li [2,3,*], Yu Zhang [2,3], Yu Weng [2,3], Kunyu Yang [2,3] and Chaomurilige [2,3,*]

[1] School of Chinese Ethnic Minority Languages and Literatures, Minzu University of China, Beijing 100081, China; 21400174@muc.edu.cn

[2] Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance, Ministry of Education, Minzu University of China, Beijing 100081, China

[3] School of Information Engineering, Minzu University of China, Beijing 100081, China

[*] Correspondence: 21302023@muc.edu.cn (Y.L.); chaomurilige@muc.edu.cn (C.)

Abstract: As one of foundation technologies for massive data processing for AI, event mining is attracting more and more attention, mainly including event detection (event trigger identification and event classification) and argument extraction. At present, EE-GCN is one of the most effective methods for event detection. However, since EE-GCN only focuses on event detection, complete event multi-tuple extraction needs to be improved. Inspired by the EE-GCN event detection method, this paper proposes an effective event extraction method via graph convolutional network indication with a hierarchical argument selection strategy. The method mainly includes the following steps. (1) Based on the ACE2005 argument extraction template, a new argument extraction template is established for the Baidu event extraction dataset. (2) The trigger events and event classification detected by EE-GCN are used as indicators to determine the argument extraction template, and the alternative arguments are extracted via named entity recognition based on the determined template. (3) Making full use of the side information of EE-GCN graph to solve the local and global correlation degree, and based on the local and global correlation degrees, the final argument multi-tuple is determined. (4) Finally, several experiments are conducted on the Baidu event extraction dataset to compare the proposed method with other methods. The experimental results show that the proposed method has improved the accuracy and completeness of the event extraction compared to other existing methods.

Keywords: data processing technology; event extraction; graph convolutional neural network; trigger words

check for updates

Citation: Liu, Z.; Li, Y.; Zhang, Y.; Weng, Y.; Yang, K.; Chaomurilige. Effective Event Extraction Method via Enhanced Graph Convolutional Network Indication with Hierarchical Argument Selection Strategy. *Electronics* **2023**, *12*, 2981. https://doi.org/10.3390/electronics12132981

Academic Editors: Maciej Ławryńczuk, Qingshan Jiang, John (Junhu) Wang and Min Yang

Received: 29 May 2023
Revised: 24 June 2023
Accepted: 29 June 2023
Published: 6 July 2023

1. Introduction

Due to the rapid growth of information on the Internet, event extraction has become a crucial task in data processing. This task involves automatically extracting structured information from massive and diverse unstructured texts. It is the foundation for mining the value of big data by using AI technology, such as automatic summarization, intelligent writing, machine translation, etc. Furthermore, it has been successfully applied in many important fields such as network public opinion monitoring, emergency alarms, intelligence collection, business and economy, automatic response, human–computer interaction, and biomedicine [1–8].

Generally, the study of event extraction can be divided into two sub-tasks. The first sub-task is event detection (event trigger identification and event classification). The second sub-task is to extract the elements involved in the event, such as person, time, place and so on, which are called arguments. In recent years, due to the great success of deep learning (especially deep convolutional neural networks and graph convolutional neural networks) and its powerful representation and learning ability in many fields [9,10], most emerging research focuses on a deep learning-based event extraction approach. According to the order of completion of the two sub-tasks, event extraction methods based on deep learning can be divided into pipeline methods and joint learning methods. Theoretically, the joint learning method is superior to the pipeline method because it considers the interaction between trigger words and arguments. However, trigger recognition and argument recognition are coupled and restricted to each other, so the learning algorithm and training mode are complex, which makes it difficult to implement. Because of this, people tend to use the pipeline method to study event extraction. The pipeline method focuses on trigger identification and takes the syntactic dependency between arguments as a supplement. Event detection is performed first, and then argument extraction is performed. Based on this idea, Cui et al. [11] proposed a new event detection model, Edge-Enhanced Graph Convolution Networks for event detection with Syntactic Relation (EE-GCN). The adjacency matrix describing the presence or presence of syntactic dependencies is transformed into a tensor describing the presence or presence of syntactic dependencies and the category of syntactic dependencies. The recognition of event trigger words and event categories is realized by using a graph convolutional neural network and softmax, which is verified on the ACE2005 dataset. Experiments show that this method is one of the most effective methods for event detection.

However, EE-GCN only completes the first sub-task of event extraction, namely event detection. For complete event extraction, it is necessary to further extract the event arguments. Inspired by this idea, on the basis of the existing research on EE-GCN, this paper proposes an event tuple extraction method. This method extends beyond existing research on EE-GCN by proposing a novel approach that effectively extracts complete event multi-tuples using a combination of graph convolutional network indication and hierarchical argument selection strategy. The hierarchical argument selection strategy can effectively improve the accuracy and completeness of multi-tuple extraction. The key contributions of this paper are as follows:

(1) We establish a more complete argument extraction template for the Baidu event extraction dataset based on the ACE2005.

(2) We propose an effective event extraction method using trigger events and event classifications detected by EE-GCN as indicators. Moreover, we employ a hierarchical argument selection strategy, which helps in determining the final argument multi-tuple more accurately.

(3) A series of experiments are conducted on the Baidu event extraction dataset to compare the proposed method with other methods, demonstrating that it improves both accuracy and completeness of event extraction compared to existing methods.

2. Related Works

From the development process of event extraction, the methods can be divided into three categories, which are pattern matching-based methods, machine learning-based methods and deep learning methods.

At present, machine learning methods and deep learning methods have gradually become the mainstream technology of event extraction. Compared to the pattern matching method used by early researchers such as Grishman et al. [12], it is easier to implement, more scalable, and portable to various fields. Following the rise of GCN (Kipf and Welling [13]), Orr et al. [14] proposed a directed acyclic graph GRU model that introduces syntactic structure into the sequence structure. Nguyen and Grishman [15] et al. proposed to transform the syntactic dependency tree into a graph, and used GCN to carry out event detection through information propagation on the graph. They used the graph to represent the syntactic dependency relationship of fields in a sentence, which captured deeper semantic features.

Event extraction methods based on deep learning can be divided into pipeline methods and joint learning methods. Ahn [16] clearly divides event extraction into four steps:

trigger detection, argument detection, event alignment and event relationship detection, and adopts the pipeline extraction idea to extract the four tasks relatively independently. With the development of information technology, researchers can enhance the effect of event extraction by combining additional information. Ji and Grishman [17] collect event extraction results from a set of related documents and use rule-based methods to infer and constrain, thus improving the performance of event extraction. Differing from the above cross-document research perspective, Liao and Grishman [18] focused on cross-event information within the same document, and believed that the previous research may be limited to the extraction of a single event in the document, ignoring the logical and temporal relationship between all different events in the same document. Liu et al. [19] designed a model based on Probabilistic Soft Logic (PSL), proposed three types of local information and two kinds of global information that play a positive role in event extraction, and combined local information and global information for event extraction. This makes the research on event extraction based on deep learning even further advanced. Due to the phased training, the pipeline method will face the problem of error propagation. A misclassified trigger will affect the results of the argument extraction, and this negative influence is irreversible. Therefore, another group of researchers prefer joint learning models that can slow down the impact of error propagation. Nguyen et al. [20] applied a recurrent neural network (RNN) to the study of event extraction and introduced the global text features proposed by Li et al. [21] to achieve the joint extraction of trigger words and arguments by combining global text features and deep learning. Liu [22] designed a joint learning event extraction framework to extract events through graph convolutional networks. Although the joint learning method has certain advantages in theory, its model design is complex, and more parameters and multi-thread output need to be considered, which causes some difficulties in practical operation.

Considering the limitations of existing methods in effectively addressing tuple extraction challenges, this study draws inspiration from the EE-GCN approach, and proposes a novel method that seeks to enhance both the accuracy and completeness of extraction. This is achieved by utilizing an improved EE-GCN model in conjunction with a hierarchical argument selection strategy, thereby offering a more robust solution for tuple extraction tasks.

3. Methods

Figure 1 shows the proposed event multi-tuple extraction model based on graph convolution indication and event correlation degree. First, a new argument role table based on ACE2005 was constructed for the dataset extracted from Baidu events. Second, the event trigger words and event categories obtained using EE-GCN were used as argument extraction indicators. Third, based on the new argument role table and indicator, named entity recognition was used to obtain all the alternative arguments of the event. Fourth, based on the side information of EE-GCN graph, the local correlation degree between the trigger word and all the alternative arguments is calculated, and the alternative arguments are divided into confirmed argument 1 and a to-be-confirmed argument based on the local correlation degree. Fifth, the global correlation degree between confirmed argument 1 and the non-argument words in the sentence was calculated, and the to-be-confirmed arguments were further divided into confirmed argument 2 and other arguments according to the global correlation degree. Last, confirmed argument 1 and confirmed argument 2 are merged into the final event argument, and then the final event argument, event classification, and event trigger words are combined to form an event tuple.

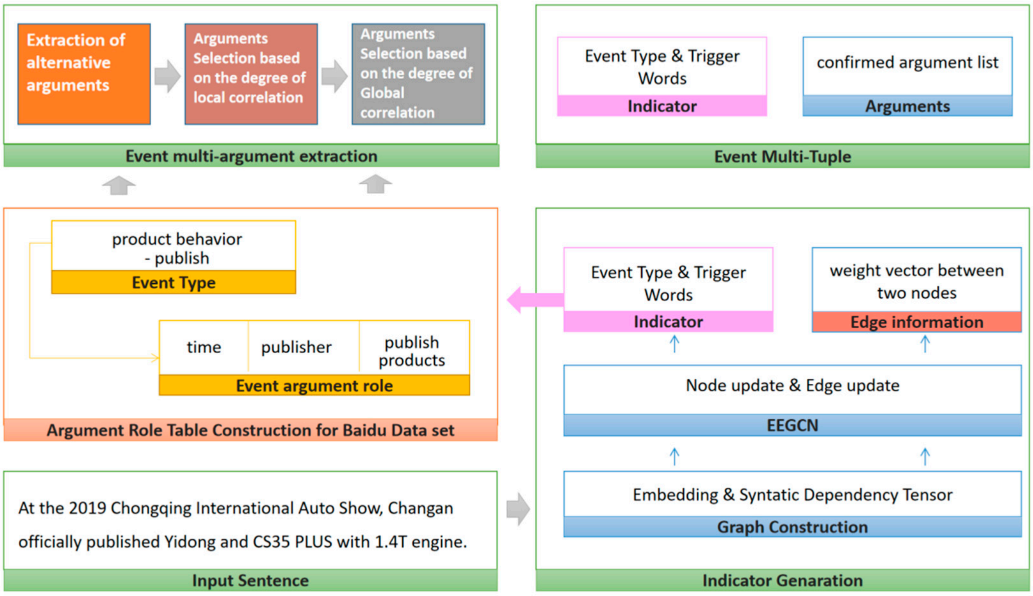
Extraction of alternative arguments
Arguments Selection based on the degree of local correlation
Arguments Selection based on the degree of Global correlation
Event multi-argument extraction
Event Type & Trigger Words
Indicator
confirmed argument list
Arguments
Event Multi-Tuple
product behavior - publish
Event Type
time
publisher
publish products
Event argument role
Argument Role Table Construction for Baidu Data set
Event Type & Trigger Words
Indicator
weight vector between two nodes
Edge information
Node update & Edge update
EEGCN
Embedding & Syntatic Dependency Tensor
Graph Construction
At the 2019 Chongqing International Auto Show, Changan officially published Yidong and CS35 PLUS with 1.4T engine.
Input Sentence
Indicator Genaration

Figure 1. The overall architecture.

3.1. Argument Role Table Construction for Baidu Dataset

The Baidu event extraction dataset [19] defines 65 event types and provides a table of event types and corresponding argument roles, but does not give the entity types corresponding to argument roles. The ACE 2005 Role Table [22] is a table used in the ACE2005 task, which is a language-based information extraction task that aims to identify named entities, relations, and events from natural language texts. The table describes the roles involved in events, including the names, meanings, and occurrences of the roles in the corpus for different event types. Referring to the correspondence between argument roles and entity types in ACE 2005, we construct the correspondence between argument roles and entity types as follows. (1) For argument roles in the Baidu event extraction dataset that are similar to the argument roles in ACE 2005, the definitions in ACE2005 are directly adopted. For example, “award winners” and “person to thank” in the Baidu event extraction data-set are equivalent to “PERSON” in ACE 2005. (2) If there is no similar argument role, its corresponding entity type, such as “translation:sale price” and other argument roles in the Baidu event extraction dataset should be defined, and “MONEY” defined as its corresponding named entity type.

Using the 65 event types defined by the Baidu event extraction dataset, and using the above ideas, the corresponding relationship between argument roles and entity types is established based on 121 event argument roles, and a new argument role table is constructed. The correspondence between partial argument roles and entity types is shown in Table 1.

Table 1. Role of event argument in Baidu dataset.

Event Type	Event Argument Role [Entity Type]
finance/trading—listing	location [GPE LOC FAC] listed enterprises [ORG] amount of financing [MONEY]
product behavior—publish	time [DATE TIME] publish products [PRODUCT WORK_OF_ART] publisher [PERSON ORG GPE NORP]
product behavior—release	time [DATE TIME] release party [ORG] release film and television [WORK_OF_AR]

Table 1. *Cont.*

Event Type	Event Argument Role [Entity Type]
competitive behavior—winning	time [DATE TIME] winner [PERSON ORG] championship event [EVENT]
life—marriage	time [DATE TIME] married parties [PERSON]
...	...

According to Table 1, the set of event types can be defined as $ET = [et_1, et_2, \ldots, et_{65}]$, where the set of argument roles of event type et_i (i = 1, 2, ... , 65) is $argument_i = \left[\mathrm{Ar}_i^1, \mathrm{Ar}_i^2, \ldots, \mathrm{Ar}_i^{ai}\right]$, and ai is the number of argument roles of event et_i. Then the current event MI_{i} can be represented as $MI_i = [0, 0, \ldots, et_{\mathrm{i}}, \ldots, 0]$ and the type number of the current event MI_i is:

$$TN_i = Num(MI_i) \tag{1}$$

where Num represents the position sequence number of "et_i" in MI_i.

3.2. Event Multi-Argument Extraction

3.2.1. Generation of Argument Indicators Based on EE-GCN

Suppose the input is a sentence sequence $S = [w_1, w_2, \ldots, w_M]$, where M represents the number of words in the sentence, $w_i(i \in (1, 2, \ldots, M))$ is the word vector $w_i \in R^{d_w}$ of the ith word in the sentence, and d_w is the dimension of the word vector. The recognition is performed in the input sentence using STANZA [23] to obtain the named entity sequence $\widehat{N} = [N_1, N_2, \ldots, N_M]$, where $N_i(i \in (1, 2, \ldots, M))$ represents the named entity type for each word. Make the embedding of each element in sequence $\widehat{N}$, and transform it into sequence $N = [n_1, n_2, \ldots, n_M]$, where $n_i \in R^{d_n}$ is the embedding corresponding to N_i, and d_n is the dimension of the embedding of a named entity type.

Concatenate w_i with n_i to form a vector embedding $x_i = [w_i; n_i] \in R^{d_w + d_n}$ for each word. Finally, the sequence of embedding vectors of the input sentence is $X = [x_1, x_2, \ldots, x_M]$. By inputting X into BiLSTM, the hidden state sequence $H^0 = \left[h_1^0, h_2^0, \ldots, h_M^0\right]$ of $X = [x_1, x_2, \ldots, x_M]$ is obtained, where $H^0 \in R^{M \times d}$, d are the dimensions of $h_i^0(i \in (1, 2, \ldots, M))$ and $\mathrm{h}_1^0, h_2^0, \ldots, h_M^0$ corresponds to $x_1, x_2, \ldots, x_M$ in the original sentence sequence one-to-one. H^0 is taken as the initial node of the EE-GCN model. Bidirectional Long Short-Term Memory (BiLSTM) is a type of artificial neural network incorporating information from both past and future time steps in a sequence. The main advantage of BiLSTM is its ability to capture context information from both forward and backward directions, resulting in a more accurate prediction. This makes it particularly useful for tasks that involve analyzing sequences of data. BiLSTM models have proven successful in tasks, including genomic sequence analysis, life prediction, and video analysis [24,25].

Then, we use STANZA [23] for a syntactic dependency analysis to obtain syntactic dependency relations between words, and to produce dependency embedding. The initial adjacency matrix $E^0 = \left\{E^0_{:,:,:}\right\} \in R^{M \times M \times K}$ and K are the dimensions of the dependency embedding. Consider E^0 the initial edge of the EE-GCN model.

The update iteration formula for the nodes and edges of EE-GCN, composed of initial nodes and edges, is shown in Formulas (2)–(4) [11].

$$H^l = relu\left(Average\_pooling\left(H_1^l, H_2^l, \ldots, H_K^l\right)\right) \tag{2}$$

$$H_1^l = E_{:,:,k}^{l-1} H^{l-1} W_H \tag{3}$$

$$E^l_{i,j,:} = W_E\left[E^{l-1}_{i,j,:} \bigoplus h^l_i \bigoplus h^l_j\right], i,j \in [1,M] \tag{4}$$

where $E^{l-1}_{:,:,k}$ is the kth channel component of E^{l-1} and $E^l_{i,j,:} \in R^K$ represents the (i,j)th element of E^{l-1}. $W_E \in R^{(2\times d+K)\times K}$ and $W_H \in R^{d\times d}, d = d_w + d_n$ are two learnable weight matrices. h^l_i and h^l_j are the representations of the i and j nodes of the $l-th$ layer, respectively.

At the last layer, each node h^l_i is fed into a fully connected network (FCN_i), followed by a Softmax ($softmax_i(.)$) used to calculate the probabilities of h^l_i for all event types, expressed as Equation (5) [14].

$$p\left(et_j \middle| h^l_i\right) = softmax_j\left(w_j h^l_i + b_j\right), i,j \in [1,M] \tag{5}$$

where w_j maps the node representation h^l_i to the feature score for each event type, and b_j is a bias term.

So, the type of event and position of *trigger word* are calculated with Equation (6), and it is taken as the indicator of event detection, expressed with I.

$$I = Arc \underset{i\in[1,M],j\in[1,q]}{max} p\left(et_j \middle| h^l_i\right) \tag{6}$$

Figure 2 illustrates the process of generating argument indicators using EE-GCN.

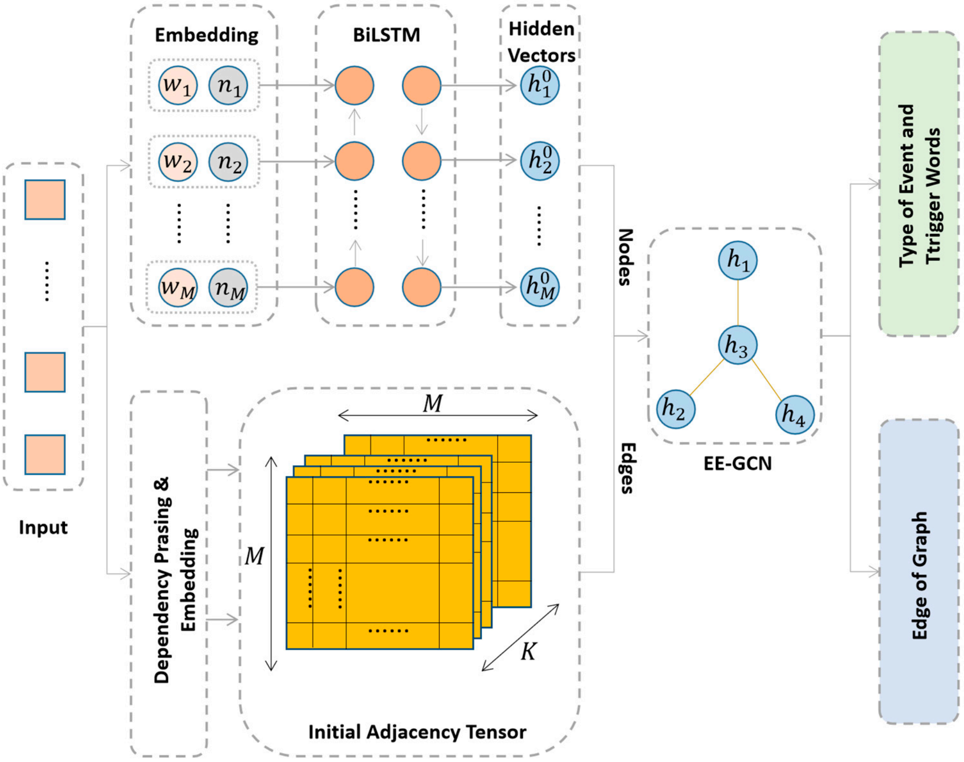

Figure 2. The generation of argument indicators based on EE-GCN.

3.2.2. Extraction of Alternative Arguments

According to the event trigger word and event type determined by Formula (6), suppose et_i, combined with the event indicator determined by Formula (1), the corresponding argument extraction template is found based on the event argument role table of the new Baidu dataset, and the argument role list $argument_i = \left[\mathrm{Ar}_i^1, \mathrm{Ar}_i^2, \ldots, \mathrm{Ar}_i^{al_i}\right]$ corresponding to the event type is obtained. The set of arguments is $Ar_i^j = \left\{RO^1_{i,j}, RO^2_{i,j}, \ldots, RO^{ar_j}_{i,j}\right\}$,

$j \in [1, al_i]$, where $RO_{i,j}^{t} = \left\{role_{i,j}^{1}, role_{i,j}^{2}, \ldots, role_{i,j}^{r_t}\right\}, t \in \left[1, ar_j\right], role_{i,j}^{p}, p \in [1, r_t]$ denotes the pth argument in the jth argument set of the ith type event, and r_t is the number of roles. Then, each word in the sentence is BIO-tagged via named entity recognition. Finally, the candidate arguments are selected according to the template determined and the entity type of the word.

After determining the argument roles and arguments using named entity recognition [26], the arguments indicated in the template are extracted from the sentence, and they are regarded as alternative arguments. The formula is described as follows:

$$Eventmulti - tuple = \{et_i, trigger\ word, RO_i\} \tag{7}$$

where *trigger word* is the event trigger word, et_i is the event type corresponding to the event trigger word, and $RO_{i,j}^{t}$ represents the argument list of the event.

Obviously, if there is only one argument for each argument role, then it can be directly included in the final argument list. If there is no argument corresponding to an argument role, then the argument corresponding to that role is empty in the final argument list. If there are multiple arguments corresponding to the same argument role in the sentence, these multiple arguments need to be further filtered. Algorithm 1 shows the extraction method of alternative arguments.

Algorithm 1 ERAR based on NER

Input: Event type E, Event Argument Dataset S, the initial empty argument dictionary D, Input text T.
Output: the final argument dictionary D.

```
roles ← $S$.get($E$)
ner_results ← NER($T$)
for $e$ ← roles do
    role_name ← $e$ [0]
    role_entity_type ← $e$ [1 ... len(e)]
    temp_words ← []
    for $v$ ← ner_results.values do
        if $v$ in role_entity_type then
            add the key corresponding to the value to the temp_words
        end if
    end for
    $D$.put(role_name,temp_words)
end for
return $D$
```

3.3. Hierarchical Argument Selection Strategy

3.3.1. The Selection of Arguments Based on the Degree of Local Correlation

Based on the output of the EE-GCN model, not only can the event trigger words be determined, but the arguments can also be filtered by analyzing the weight of the graph network edges. The edge tensor information in the output of the graph convolutional layer of the EE-GCN model is denoted as $E_{out} \in R^{M \times M \times K}$, where M represents the number of nodes in the graph and K is the dimension of the weight vector between two nodes. The weight of the edge between the ith node and the jth node in the graph is denoted as the dimension vector $E_{out}(i, j) = \left[e_{i,j}^{1}, e_{i,j}^{2}, \ldots, e_{i,j}^{K}\right]$. In the convolution process, nodes with a closely related degree will interact more and produce more information. In other words, information about the degree of association between nodes is contained in the edge tensor message $E_{out} \in R^{M \times M \times K}$. Therefore, in this section, the association relevance of edges between two nodes is first defined to represent the closeness between nodes, as shown in Formula (8):

$$relevance(i,j) = \sum\nolimits_{q=1}^{K} \left(e_{i.j}^{q}\right)^{2}, i,j \in [1,M] \tag{8}$$

If the degree of association between the alternative argument and the event trigger is closer, the probability that the alternative argument will be identified as the final argument is higher. Therefore, this paper uses the local relevance degree between alternative arguments and event trigger words to filter alternative arguments in the first round.

Denote the trigger word as $Trigger_i$ in the sentence sequence S, the arguments in the t-th alternative argument set $RO_{i,j}^{t}$ of the jth argument role Ar_i^j, and the edges corresponding to $role_{i,j}^{p}, p \in [1, r_t]$ as $E_{out}(i,p) = \left[e_{i,j}^{p,1}, e_{i,j}^{p,2}, \ldots, e_{i,j}^{p,K}\right]$. Define the mean correlation degree of the edges connected to the event trigger word and the alternative arguments as the local correlation degree threshold $threshold_{local}$. Based on Formula (8), the local correlation degree threshold $threshold_{local}$ can be calculated from Formulas (9) and (10):

$$relevance(i,p) = \sum\nolimits_{q=1}^{K} \left(e_{i.j}^{p,q}\right)^{2}, i,j \in [1,M] \tag{9}$$

$$threshold_{local} = \frac{1}{r_t} \sum\nolimits_{p=1}^{r_t} relevance(i,p) \tag{10}$$

Use Formula (9) to calculate the correlation degree $relevance(i,p)$ between each argument $role_{i,j}^{p}, p \in [1, r_t]$ and the trigger word in $RO_{i,j}^{t}$. If $relevance(i,p) \geq threshold_{local}$, $role_{i,j}^{p}, p \in [1, r_t]$ is regarded as the confirmed argument role and stored in the confirmed argument set $\left(RO_{i,j}^{t}\right)_1$, that is, $\left(RO_{i,j}^{t}\right)_1 = \left\{role_{i,j}^{1}, role_{i,j}^{2}, \ldots, role_{i,j}^{r_{t1}}\right\}$ and r_{t1} are the number of confirmed arguments. Otherwise, $role_{i,j}^{p}, p \in [1, r_t]$ is regarded as an unconfirmed argument role stored in the unconfirmed argument set $\left(\widehat{RO_{i,j}^{t}}\right)_1$, that is, $\left(\widehat{RO_{i,j}^{t}}\right)_1 = \left\{\widehat{role}_{i,j}^{1}, \widehat{role}_{i,j}^{2}, \ldots, \widehat{role}_{i,j}^{\widehat{r_{t1}}}\right\}$, $\widehat{r_{t1}}$ is the number of unconfirmed arguments.

3.3.2. Argument Selection Based on the Global Correlation Degree

For $\left(\widehat{RO_{i,j}^{t}}\right)_1 = \left\{\widehat{role}_{i,j}^{1}, \widehat{role}_{i,j}^{2}, \ldots, \widehat{role}_{i,j}^{\widehat{r_{t1}}}\right\}$, if the closeness between the unconfirmed argument role in $\left(\widehat{RO_{i,j}^{t}}\right)_1$ and the confirmed argument role in $\left(RO_{i,j}^{t}\right)_1 = \left\{role_{i,j}^{1}, role_{i,j}^{2}, \ldots, role_{i,j}^{r_{t1}}\right\}$ is higher, the probability that the unconfirmed argument will be determined as the final argument is higher. Remember that all the words excluding the alternative argument set in the sentence sequence are $Se = S/\left(RO_{i,j}^{t}\right)_1 = \left[\widehat{\mathrm{we}}_{i,j}^{t,1}, \widehat{\mathrm{we}}_{i,j}^{t,2}, \ldots, \widehat{\mathrm{we}}_{i,j}^{t,(M-r_{t1})}\right]$ (including the unknown argument role), and the dimension vector of the edge weight between any word $\widehat{we}_{i,j}^{t,u}$ in Se and any argument $role_{i,j}^{v}$ in $\left(RO_{i,j}^{t}\right)_1 = \left\{role_{i,j}^{1}, role_{i,j}^{2}, \ldots, role_{i,j}^{r_{t1}}\right\}$ is $E_{out}\left(\widehat{we}_{i,j}^{t,u}, role_{i,j}^{v}\right) = \left[\left(e_{i,j}^{\mathrm{u,v}}\right)_1, \left(e_{i,j}^{\mathrm{u,v}}\right)_2, \ldots, \left(e_{i,j}^{\mathrm{u,v}}\right)_K\right]$. Then, according to Formula (8), the correlation degree between the two can be calculated as follows:

$$relevance(u,v) = \sum\nolimits_{q=1}^{K} \left(\left(e_{i,j}^{\mathrm{u,v}}\right)_q\right)^{2}, u \in [1, (M - r_{t1})], v \in [1, r_{t1}] \tag{11}$$

define the global correlation threshold as $threshold_{global}$:

$$threshold_{global} = \frac{1}{r_{t1}(M - r_{t1})} \sum\nolimits_{v=1}^{r_{t1}} \sum\nolimits_{u=1}^{M-r_{t1}} relevance(u,v) \tag{12}$$

using Formula (11), calculate the degree of correlation degree $relevance(u,v)$ between each argument $\widehat{role}_{i,j}^{u}$ in $\widehat{\left(RO_{i,j}^{t}\right)}_{1} = \left\{\widehat{role}_{i,j}^{1}, \widehat{role}_{i,j}^{2}, \ldots, \widehat{role}_{i,j}^{\widehat{r}_{t1}}\right\}$ and any argument $role_{i,j}^{v}$ in $\left(RO_{i,j}^{t}\right)_{1} = \left\{role_{i,j}^{1}, role_{i,j}^{2}, \ldots, role_{i,j}^{r_{t1}}\right\}$. When $relevance(u,v) \geq threshold_{global}$, $\widehat{role}_{i,j}^{u}$, $u \in [1, \widehat{r}_{t_1}]$ is regarded as a confirmed argument role and stored into the confirmed argument set $\left(RO_{i,j}^{t}\right)_{2}$, that is, $\left(RO_{i,j}^{t}\right)_{2} = \left\{role_{i,j}^{1}, role_{i,j}^{2}, \ldots, role_{i,j}^{r_{t2}}\right\}$ and r_{t2} are the number of confirmed arguments. Otherwise, it can be discarded.

3.3.3. Event Multi-Tuple Generation

The confirmation argument set $\left(RO_{i,j}^{t}\right)_{1}$ is merged with the confirmation argument set $\left(RO_{i,j}^{t}\right)_{2}$ to generate the jth argument role Ar_j final argument list.

$$\left(RO_{i,j}^{t}\right)_{F} = \left(RO_{i,j}^{t}\right)_{1} \cup \left(RO_{i,j}^{t}\right)_{2} \tag{13}$$

Finally, based on Formulas (5)–(7) and (13), the event type, event trigger word, and the final argument list are merged to generate a multi-tuple of events.

$$\text{Event multi} - \text{tuple} = \left\{\text{et}_i, triggerword, \left(RO_{i,j}\right)_F\right\} \tag{14}$$

To better illustrate the above process, we provide a brief description of the argument selection procedure using an example of multi-tuple extraction from a single sentence, as shown in Figure 3.

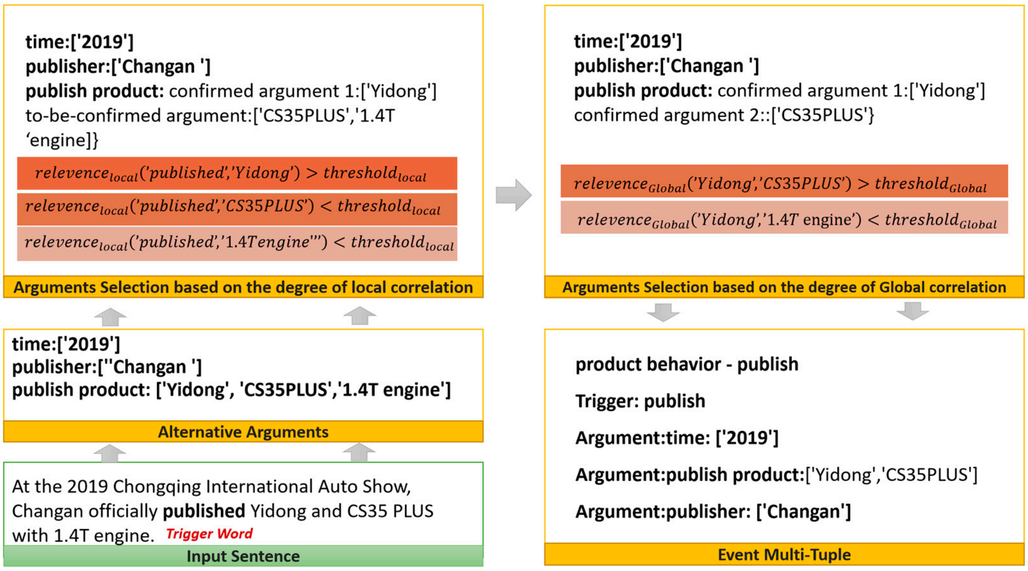
time:['2019']
publisher:['Changan ']
publish product: confirmed argument 1:['Yidong']
to-be-confirmed argument:['CS35PLUS','1.4T
'engine]}
$relevence_{local}$('published','Yidong') > $threshold_{local}$
$relevence_{local}$('published','CS35PLUS') < $threshold_{local}$
$relevence_{local}$('published','1.4Tengine'") < $threshold_{local}$
Arguments Selection based on the degree of local correlation
time:['2019']
publisher:['Changan ']
publish product: confirmed argument 1:['Yidong']
confirmed argument 2::['CS35PLUS'}
$relevence_{Global}$('Yidong','CS35PLUS') > $threshold_{Global}$
$relevence_{Global}$('Yidong','1.4T engine') < $threshold_{Global}$
Arguments Selection based on the degree of Global correlation
time:['2019']
publisher:["Changan ']
publish product: ['Yidong', 'CS35PLUS','1.4T engine']
Alternative Arguments
At the 2019 Chongqing International Auto Show,
Changan officially published Yidong and CS35 PLUS
with 1.4T engine.
Trigger Word
Input Sentence
product behavior - publish
Trigger: publish
Argument:time: ['2019']
Argument:publish product:['Yidong','CS35PLUS']
Argument:publisher: ['Changan']
Event Multi-Tuple

Figure 3. The extraction of arguments.

4. Experiment

4.1. Experimental Environment

Several experiments were conducted to compare the effectiveness of the algorithm proposed in this paper with existing methods. The experiments mainly consist of three parts, as follows. (1) The sentence-level event multi-tuple extraction is implemented using our method. (2) The HIT language technology platform LTP is used to extract the sentence-level event multi-tuple. (3) The Baidu ERNIE model is used to achieve sentence-level event tuple extraction. The experimental results of the proposed method are compared with the results of LTP and Baidu to illustrate the effectiveness of the proposed method.

These experiments are composed of a hardware and software environment. The hardware mainly includes 128 GB memory, an Intel(R) Xeon(R) E5-2620 v4 CPU and a NVIDIA GeForce GTX 1080 Ti GPU. The software framework employed for the experiments includes Ubuntu 9.4.0 as the operating system and python3.7 as the programming language. The Baidu event extraction dataset [27] was utilized as the primary dataset for the experiments.

4.2. Extracting the Event Multi-Tuple

4.2.1. Extraction of Alternative Arguments

Four sentences (id 1–4, as shown in Table 2) were randomly selected from the Baidu event extraction dataset as the input sentence text. First, the trigger words and event types were generated using Formulas (1)–(6), and the argument was determined using the trigger words and event types. Then, based on the argument template, Formula (7) is used to determine the alternative arguments, and the output results are shown in Tables 2–5, where the second column is the content of the input sentence, the third column is the generated indicator, and the fourth column is the alternative argument.

Table 2. Extraction of alternative arguments (1).

Sentence ID	**1**
Sentence	Tsingtao Brewery shares (00168) employee representative supervisor Xing Jun resigned.
Indicator	trigger: resign Event Type: organizational relationship—resignation
Alternative arguments	time: [] resigned person: [‘Xing Jun’] original organization: [‘Tsingtao Brewery Shares’]

Table 3. Extraction of alternative arguments (2).

Sentence ID	**2**
Sentence	A man named Ma has been suspended and is being investigated by the county’s discipline inspection commission, according to a person in charge of the county management over-speed station on June 4.
Indicator	trigger: suspension Event Type: organizational relations—suspension
Alternative arguments	time: [‘June 4’] organization: [‘the county management over-speed station’, ‘county’s discipline inspection commission’] suspended person: [‘Ma’]

Table 4. Extraction of alternative arguments (3).

Sentence ID	**3**
Sentence	At the 2019 Chongqing International Auto Show, Changan officially published Yidong and CS35 PLUS with a 1.4 T engine.
Indicator	trigger: publish Event Type : product behavior—publish
Alternative arguments	time: [‘2019’] publish products: [‘Yidong’, ‘CS35PLUS’, ‘1.4 T engine’] publisher: [‘Chang’ an’]

Table 5. Extraction of alternative arguments (4).

Sentence ID	4
Sentence	On July 18, the reporter learned from the provincial flood control and disaster reduction work dispatch meeting that since July 16, Dazhou City has experienced heavy rainfall. Affected by rainfall, Dazhou City's main rivers have suffered varying degrees of flooding.
Indicator	trigger: flood Event Type: disaster/accident—flood
Alternative arguments	time: ['16 July', '18 July'] location: ['Dazhou City']

Tables 2–5 shows that for sentences 1 and 2, each role of the argument corresponds to only one argument, and the event tuple can be obtained by matching by the type of the named entity. For sentences 2–4, in the event tuple extracted after matching by the type of named entity, there is a situation in which one argument role corresponds to more than one argument. For example, the argument role "organization" in sentence 2 corresponds to two arguments: "the county management over-speed station" and "county's discipline inspection commission", one of which needs to be further selected.

4.2.2. Selecting of Alternative Arguments

For the problem that one argument role corresponds to more than one argument, the selection method proposed in this paper in Section 3.3 is used. For a specific alternative argument, (a) Formula (10) is used to calculate the local correlation degree of the edges connecting the event trigger word and the alternative argument; then, Formula (9) is used to calculate the correlation degree between each alternative argument and the trigger word, and the two are compared. According to the comparison results, the alternative arguments are classified into confirmed argument 1 and the to-be-confirmed argument. (b) Formula (12) is used to calculate the global correlation degree between the words in the text, except the alternative argument and the confirmation argument 1; secondly, Formula (11) is used to calculate the correlation degree between the argument to be confirmed and the confirmed argument 1. The two arguments were compared, and arguments to be confirmed were classified as confirmed argument 2, or deleted according to the comparison results. (c) The final argument is the union of confirmation argument 1 and confirmation argument 2. The results are shown in Table 6, where the second column is the alternative arguments, the third column is the result based on the selection, and the fourth column is the result of the second round of screening.

Table 6 shows that by calculating the correlation degree of edges between each alternative argument and the trigger word and comparing it with the local correlation degree threshold, the alternative arguments are divided into confirmed argument 1 and to-be-confirmed arguments, and confirmed argument 1 can be directly included in the final event multi-tuple. Then, by calculating the degree of correlation of edges between the to-be-confirmed argument and the confirmed argument 1 and comparing it with the global correlation threshold, the to-be-confirmed argument is divided into confirmed argument 2 or deleted. In sentence 2, the to-be-confirmed argument "county's discipline inspection commission" is deleted in the second round of selecting, which determines the argument composition in the final event multi-tuple. However, in sentence 3, the to-be-confirmed argument "1.4-T engine" is deleted, while "CS35PLUS" is confirmed. Therefore, in the multi-tuple event in sentence 5, there are two arguments, "Yidong" and "CS35PLUS", in the argument role of "publish products", and a similar phenomenon also appears in sentence 4.

Table 6. Results of the selection of alternative arguments.

Sentence ID	Alternative Arguments	Results of the Selection Based on Local Association	Results of the Selection Based on Global Association
2	time: ['June 4'] organization: ['the county management over-speed station', 'county's discipline inspection commission'] suspended person: ['Ma']	organization:{confirmed argument 1: ['the county management over-speed station'] to-be-confirmed argument: ['county's discipline inspection commission']})	organization: {confirmed argument 2: []}
3	time: ['2019'] publish products: ['Yidong', 'CS35PLUS', '1.4 T engine'] publisher: ['Chang' an']	publish products: {confirmed argument 1: ['Yidong'] to-be-confirmed arguments: ['CS35PLUS','1.4 T engine']}	publish product: {confirmed argument 2: ['CS35PLUS']}
4	time: ['16 July', '18 July'] location: ['Dazhou City']	time:{confirmed argument 1: ['16 July'] to-be-confirmed arguments: ['18 July']}	time: {confirmed argument 2: []}

Finally, the resulting final arguments are combined with event triggers and event types to form the final event multi-tuple, as shown in Table 7.

Table 7. The final result of the event multi-tuple.

	Sentence	Event Multi-Tuple
2	A man named Ma, has been suspended and is being investigated by the county's discipline inspection commission, according to a person in charge of the county management over-speed station on June 4.	organizational relations-suspension Trigger: suspension time: ['June 4'] Argument: organization: ['the county management over-speed station'] Argument: suspended person: ['Ma ']
3	At the 2019 Chongqing International Auto Show, Changan officially published Yidong and CS35 PLUS with a 1.4 T engine.	Event Type: product behavior—publish Trigger: publish Argument: time: ['2019'] Argument: publish product: ['Yidong', 'CS35PLUS'] Argument: publisher: ['Chang' an ']
4	On July 18, the reporter learned from the provincial flood control and disaster reduction work dispatch meeting that since July 16, Dazhou City has experienced heavy rainfall. Affected by rainfall, Dazhou City's main rivers have suffered varying degrees of flooding.	Event Type: disaster/accident—flood Trigger: flood Argument: time: ['July 16th'] Argument: location: ['Dazhou City']

Obviously, the event tuples extracted using the proposed method are the same as the event tuples refined after human reading.

4.3. Case Study of the Comparison Experiment

To prove the effectiveness of the method proposed in this paper, the HIT language technology platform (LTP, a Chinese natural language tool processing package) and Baidu ERNIE model were selected as baselines and compared with our model for the same input sentences; the results are shown in Tables 8 and 9.

Table 8. Experimental results compared with LTP.

Sentence ID	Sentence	Our	LTP
1	Tsingtao Brewery shares (00168) employee representative supervisor Xing Jun resigned.	Event Type: organizational relationship-resignation Trigger: resignation Argument: resigned person: Xing Jun Argument: original organization: Tsingtao Brewery shares	[]
2	A man named Ma has been suspended and is being investigated by the county's discipline inspection commission, according to a person in charge of the county management over-speed station on June 4.	Event Type: organizational relations-suspension Trigger: suspension Argument: time: ['June 4th'] Argument: organization: ['the county management over-speed station'] Argument: suspended person: ['Ma']	Trigger: intervene Argument: entity: county's discipline inspection commission Argument: time:for now, June 4th
3	At the 2019 Chongqing International Auto Show, Changan officially published Yidong and CS35 PLUS with a 1.4 T engine.	Event Type: product behavior-publish Trigger: publish Argument: time: ['2019'] Argument: publish product: ['Yidong', 'CS35PLUS'] Argument: publisher: ['Chang' an ']	Trigger: publish Argument: publisher: Chang' an Argument: publish product: equipped with a 1.4 T engine) Argument: time: 2019

Table 9. Experimental results compared with ERINE.

Sentence ID	Sentence	Our	ERINE
1	Tsingtao Brewery shares (00168) employee representative supervisor Xing Jun resigned.	Event Type: organizational relationship-resignation Trigger: resignation Argument: resigned person: Xing Jun Argument: original organization: Tsingtao Brewery shares	Event Type: organizational relationship-resignation Trigger: resignation Argument: resigned person: employee representative supervisor Xing Jun Argument: original organization: Tsingtao Brewery shares
2	A man named Ma has been suspended and is being investigated by the county's discipline inspection commission, according to a person in charge of the county management over-speed station on June 4.	Event Type: organizational relations-suspension Trigger: suspension Argument: time: ['June 4th'] Argument: organization: ['the county management over-speed station'] Argument: suspended person: ['Ma']	Event Type: organizational relations-suspension Trigger: suspension Argument: suspension person:negative) Argument: suspended person: ['The involved man, Ma']
4	On July 18, the reporter learned from the provincial flood control and disaster reduction work dispatch meeting that since July 16, Dazhou City has experienced heavy rainfall. Affected by rainfall, Dazhou City's main rivers have suffered varying degrees of flooding.	Event Type: disaster/accident—flood Trigger: flood Argument: time: ['July 16th'] Argument: location: ['Dazhou City']	Event Type: disaster/accident—flood Trigger: flood Argument: time: ['July 16th'] Argument: location: Dazhou City Argument: location: the main rivers of Dazhou city

In order to prove the effectiveness of the proposed method, we used LTP, the Baidu model, and the proposed method to extract arguments for the same set of sentences, and compared the arguments extracted by the three methods.

As can be seen in Table 8, the arguments extracted by the proposed method have better completeness and a higher accuracy compared with LTP. For example, in sentence 3, the arguments extracted by this paper's method are "publish products" are "Yidong" and "CS35PLUS", while the arguments extracted by LTP are "equipped with a 1.4 T engine". According to the meaning of the sentence, the arguments extracted by this paper's method are correct. In the sixth sentence, the argument roles extracted by LTP are only "product" and "time", while the argument roles extracted by the method in this paper are "publish products", "publish time", and "publisher".

As can be seen in Table 9, for the Baidu ERINE model, the proposed method outperforms it in terms of completeness, and has higher precision. For example, in sentence 2, ERINE incorrectly extracted the argument "suspension person: negative" whose argument role is "suspended person: ['Ma']". In sentence 4, the proposed method extracted the argument "Dazhou City" with the argument role as "location", while ERINE erroneously extracted "the main rivers of Dazhou city".

4.4. Statistical Analysis of the Comparison Experiment

Traditional overlapping matching evaluation methods frequently produce sub-optimal outcomes due to their reliance on scoring metrics that depend on the ratio between the length of extracted arguments and reference values. As a result, shorter yet more precise arguments often receive lower scores, compromising the overall efficacy of these techniques. For instance, the extracted result "employee representative supervisor Xing Jun" obtains a higher matching score than "Xing Jun" in sentence 1. Upon examining the Baidu dataset and the extracted results, numerous instances resembling Sentence 1 were identified. Employing traditional overlapping matching for evaluation purposes could lead to significant discrepancies, as demonstrated by the aforementioned case study.

To ensure the reliability of evaluation outcomes, this study adopted a combination of sampling and manual assessment methodologies. Five native language evaluators were assigned to randomly select and review 100 data samples from the experimental results. These evaluators examined each sample using three key metrics (coverage (C), precision (P), and recall (R)), thereby establishing a rigorous and systematic evaluation process. Besides, the harmonic mean of C, P, and R—namely, the F1-value—was calculated to provide a comprehensive measure of performance. This approach allowed for a more logical and academically sound analysis of the dataset under investigation.

$\mathrm{N}^{\mathrm{ex}}_{\mathrm{total}}$ represents the total number of predicted arguments. $\mathrm{N}^{\mathrm{ex}}_{\mathrm{right}}$ denotes the number of correctly predicted arguments. N^{re} signifies the total number of annotated arguments. These key metrics can be calculated as follows:

$$\text{Coverage} = \frac{\mathrm{N}^{\mathrm{ex}}_{\mathrm{total}}}{\mathrm{N}^{\mathrm{re}}} \tag{15}$$

$$\text{Precision} = \frac{\mathrm{N}^{\mathrm{ex}}_{\mathrm{right}}}{\mathrm{N}^{\mathrm{ex}}_{\mathrm{total}}} \tag{16}$$

$$\text{Recall} = \frac{\mathrm{N}^{\mathrm{ex}}_{\mathrm{right}}}{\mathrm{N}^{\mathrm{re}}} \tag{17}$$

$$\mathrm{F1} = \frac{\mathrm{C} * \mathrm{P} * \mathrm{R} * 3}{\mathrm{C} * \mathrm{P} + \mathrm{P} * \mathrm{R} + \mathrm{C} * \mathrm{R}} \tag{18}$$

The final results are shown as Table 10 and ranked based on the F1 score, which provides a comprehensive measure of performance by considering all key metrics. This approach allows for a more logical and academically sound analysis of the results. Upon examining data samples from various systems in comparison with our method, it was observed that our method effectively extracts not only the correct arguments, but also those beyond the reference set, with high accuracy rates. Additionally, our approach demonstrates greater granularity and precision compared to other systems. Although the recall (R) value is marginally lower than that of ERINE, our method's overall performance surpasses that of both ERINE and LTP. With an F1 value 1.6 points higher than ERINE and significantly stronger than LTP, it is evident from Table 10 that our method possesses superior extraction capabilities and can extract more accurate arguments.

Table 10. Statistical analysis results of the comparison experiment.

Systems	Coverage	Precision	Recall	F1
LTP	69.8	79.7	76.5	75.1
ERINE	94.8	83.9	**86.8**	88.3
OUR	**97.5**	**87.4**	85.6	**89.9**

5. Concluding Remarks

In this paper, we propose an effective event extraction method via graph convolutional network indication with a hierarchical argument selection strategy. The retrieval and classification of event trigger words based on EE-GCN is used as the indicator of an argument extraction template, and the argument extraction template is selected for the Baidu event extraction dataset. On the basis of this template, the candidate arguments are obtained using named entity recognition. Then, the local and global correlation degree based on EE-GCN solution is used to further filter the alternative arguments to improve the accuracy and completeness of the extracted arguments. The experimental results compared with the LTP and Baidu model verify the effectiveness of the proposed method. The results show that the proposed method is an effective event multi-tuple extraction method, which provides an effective support technology for text mining applications. At present, this method is limited to sentence-level event multi-tuple extraction. The extension of this method to document-level event multi-tuple extraction is an important objective of future research works.

Author Contributions: Z.L. proposed the initial ideas and the overarching research aims. In addition, he proofread and revised the manuscript. Y.L. and Y.Z. carried out the experiment. Y.L. and K.Y. wrote the main manuscript text. C. and Y.W. proposed the research. All authors have read and agreed to the published version of the manuscript.

Funding: We would like to thank Guosheng Yang for providing helpful constructive comments that improved the manuscript substantially, and for providing the financial support for the project leading to this publication. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1406702-3, and in part by the National Natural Science Foundation of China under Grant 62006257.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The dataset used for this work is the DuEE 1.0 dataset published by Li X [27]. It can be found at https://aistudio.baidu.com/aistudio/competition/detail/65/0/introduction (accessed on 28 June 2023).

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Zhan, L.; Jiang, X. Survey on event extraction technology in information extraction research area. In Proceedings of the IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 2121–2126.
2. Xiang, W.; Wang, B.; Bang, W. A survey of event extraction from text. *IEEE Access* **2019**, *7*, 173111–173137. [CrossRef]

3. Conlon, S.J.; Abrahams, A.S.; Simmons, L.L. Terrorism information extraction from online reports. *J. Comput. Inf. Syst.* **2015**, *55*, 20–28. [CrossRef]
4. Tanev, H.; Piskorski, J.; Atkinson, M. Real-time news event extraction for global crisis monitoring. In Proceedings of the 13th International Conference on Applications of Natural Language to Information Systems (NLDB 2008), London, UK, 24–27 June 2008; pp. 207–218.
5. Atkinson, M.; Du, M.; Piskorski, J.; Tanev, H.; Yangarber, R.; Zavarella, V. Techniques for multilingual security-related event extraction from online news. In *Computational Linguistics*; Springer: Berlin, Germany, 2013; pp. 163–186.
6. Nuij, W.; Hogenboom, F.; Frasincar, F.; Kaymak, U. An automated framework for incorporating news into stock trading strategies. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 823–835. [CrossRef]
7. Capet, P.; Delavallade, T.; Nakamura, T.; Sandor, A.; Tarsitano, C.; Voyatzi, S. A risk assessment system with automatic extraction of event types. In Proceedings of the Intelligent Information Processing IV: 5th IFIP International Conference on Intelligent Information Processing, Beijing, China, 19–22 October 2008; pp. 220–229.
8. Vanegas, J.A.; Matos, S.; González, F.; Oliveira, J.L. An overview of biomolecular event extraction from scientic documents. *Comput. Math. Methods Med.* **2015**, *2015*, 571381. [CrossRef] [PubMed]
9. Zhang, J.; Li, X.; Tian, J.; Jiang, Y.; Luo, H.; Yin, S. A variational local weighted deep sub-domain adaptation network for remaining useful life prediction facing cross-domain condition. *Reliab. Eng. Syst. Saf.* **2023**, *231*, 108986. [CrossRef]
10. Zhang, J.; Li, X.; Tian, J.; Luo, H.; Yin, S. An integrated multi-head dual sparse self-attention network for remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2023**, *233*, 109096. [CrossRef]
11. Cui, S.; Yu, B.; Liu, T.; Zhang, Z.; Wang, X.; Shi, J. Edge-enhanced graph convolution networks for event detection with syntactic relation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), Online, 16–20 November 2020; p. 10757.
12. Grishman, R. Information extraction: Techniques and challenges. In *Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 1997; pp. 10–27.
13. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
14. Orr, J.W.; Tadepalli, P.; Fern, X. Event detection with neural networks:A rigorous empirical evaluation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 999–1004.
15. Nguyen, T.H.; Grishman, R. Graph convolutional networks with argumant-aware pooling for event detection. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LO, USA, 2–7 February 2018; pp. 5900–5907.
16. Ahn, D. The stages of event extraction. In Proceedings of the Workshop on Annotating and Rcasoning about Time and Events, Sydney, Australia, 23 July 2006; pp. 1–8.
17. Ji, H.; Grishman, R. Refining event extraction through cross-document inference. In Proceedings of the Meeting of the Association for Computational Linguistics, Columbus, OH, USA, 15–20 June 2008; pp. 254–262.
18. Liao, S.; Grishman, R. Using document level cross-event inference to improve eventextraction. In Proceedings of the Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 789–797.
19. Liu, S.; Liu, K.; He, S.Z.; Zhao, J. A Probabilistic Soft Logic based approach to exploiting latent and global information in event classification. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2993–2999.
20. Nguyen, T.H.; Cho, K.; Grishman, R. Joint event extraction via recurent neural networks. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, San Diego, CA, USA, 12–17 June 2016; pp. 300–309.
21. Li, Q.; Wan, H.; Huang, L. Joint event extraction via structured prediction with global features. In Proceedings of the Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; pp. 73–82.
22. Walker, C.; Strassel, S.; Medero, J.; Maeda, K. *ACE 2005 Multilingual Training Corpus LDC2006T06*; Web Download; Linguistic Data Consortium: Philadelphia, PA, USA, 2006.
23. Qi, P.; Zhang, Y.; Zhang, Y.; Bolton, J.; Manning, C.D. Stanza: A Python natural language processing toolkit for many human languages. In Proceedings of the Association for Computational Linguistics (ACL) System Demonstrations, Online, 5–10 July 2020.
24. Zhang, J.; Tian, J.; Li, M.; Leon, J.I.; Franquelo, L.G.; Luo, H.; Yin, S. A parallel hybrid neural network with integration of spatial and temporal features for remaining useful life prediction in prognostics. *IEEE Trans. Instrum. Meas.* **2022**, *72*, 1–12. [CrossRef]
25. Zhang, J.; Huang, C.; Chow, M.Y.; Li, X.; Tian, J.; Luo, H.; Yin, S. A Data-model Interactive Remaining Useful Life Prediction Approach of Lithium-ion Batteries Based on PF-BiGRU-TSAM. *IEEE Trans. Ind. Inform.* **2023**. [CrossRef]
26. Akbik, A.; Blythe, D.; Vollgraf, R. Contextual String Embeddings for Sequence Labeling. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 1638–1649.
27. Li, X.; Li, F.; Pan, L.; Chen, Y.; Peng, W.; Wang, Q.; Lyu, Y.; Zhu, Y. DuEE: A Large-scale dataset for chinese event extraction in real-world scenarios. In Proceedings of the Natural Language Processing and Chinese Computing: 9th CCF International Conference (NLPCC 2020), Zhengzhou, China, 14–18 October 2020; pp. 534–545.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.