Hierarchical Chinese Legal event extraction via Pedal Attention Mechanism

Shirong Shen[1] and **Guilin Qi**[1*] and **Zhen Li**[1] and **Sheng Bi**[1] and **Lusheng Wang**[2]
[1]School of Computer Science and Engineering, Southeast University, China
[2]School of Law, Southeast University, China
{ssr, gqi, jeffery.lee.0628, bisheng}@seu.edu.cn,
lusheng.wang@gmail.com

Abstract

Event extraction plays an important role in legal applications, including case push and auxiliary judgment. However, traditional event structure cannot express the connections between arguments, which are extremely important in legal events. Therefore, this paper defines a dynamic event structure for Chinese legal events. To distinguish between similar events, we design hierarchical event features for event detection. Moreover, to address the problem of long-distance semantic dependence and anaphora resolution in argument classification, we propose a novel pedal attention mechanism to extract the semantic relation between two words through their dependent adjacent words. We label a Chinese legal event dataset and evaluate our model on it. Experimental results demonstrate that our model can surpass other state-of-the-art models.

1 Introduction

The number of available Chinese legal documents has increased dramatically in recent years. Event extraction (EE) of legal documents plays an important role in various legal applications, including case push and auxiliary judgment (Ashley, 2017). For example, a crime-related event contains the defendant's crime facts and crime details, which are key elements to the court's decision.

Traditional event extraction follows the event structure defined by ACE (Automatic Context Extraction) [1] and is divided into two subtasks: (1) event detection, extracting event trigger words in text and predicting event types; (2) event argument extraction, extracting arguments related to events and predicting the roles of the arguments. With the development of natural language processing technology, there are many excellent event extraction techniques in the open field (Xiang and Wang, 2019). There are few event extraction methods for the legal documents. Lagos et al. (2010) use a rule-based method to extract event in legal documents. Li et al. (2019a) apply the method of the neural network to the extraction of legal events. However, there are two major issues in legal event extraction that require more effort.

On the one hand, the traditional event structure and event definition cannot represent legal events well. For example, in Figure 1 traditional event structure cannot express some connections between arguments. If AGE is used as an argument of event, it will cause ambiguity. Besides, according to the traditional event definition method, **died** event will be separated from **stabbed** event so that the causal relationship between the victim's death and the defendant's behavior cannot be reflected. To solve this problem, we proposed a dynamic hierarchical event structure to represent legal events according to legal requirements. We first define the hierarchical event types to reflect the inclusion of different legal events. Then we carefully design different argument types to include the key elements of legal events. Moreover, we propose a dynamic event structure, which stipulates that an event's argument can have child-arguments related to the event. As shown in Figure 1, our event structure can contain all key elements in one event without ambiguity. We extend the event argument extraction task in predicting event-argument roles and

* Corresponding author.

[1]Ace (automatic content extraction) english annotation guidelines for events. https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf.

Proceedings of the 28th International Conference on Computational Linguistics, pages 100–113
Barcelona, Spain (Online), December 8-13, 2020

argument-argument roles at the same time. Furthermore, in order to distinguish between similar events, we add hierarchical event features in event detection.

On the other hand, there are long-distance semantic dependence and anaphora resolution problems in the legal sentence. In Figure 1, the first Pei is far from **died**, and in dependency syntax tree, **died** is related to the second Pei. Traditional methods like GNN(Sha et al., 2018) cannot capture the association between the two words because the two words are far away in the dependency syntax tree. So, we propose a novel pedal attention mechanism to solve this problem. We analyze the attention weights between a word and the dependency adjacent words of another word to determine the semantic relation between the two words. And the dependency adjacent words are called pedal.

Specifically, this paper proposes a pedal attention based joint hierarchical event extraction model for the legal event. In the training and prediction phase, we design a joint inference model to constrain the overall event. We label a Chinese legal event dataset and evaluate our model on it. Experiments show that our method outperforms previous state-of-the-art approaches on Chinese legal event extraction. Moreover, the dynamic event structure we defined for event extraction has important practical significance.

The rest of the paper is organized as follows. Section 2 introduces the related work of event extraction in the legal domain. Section 3 introduces the dynamic hierarchical event structure for the legal event. Section 4 details our event extraction model and the joint inference model. In Section 5, we describe our experimental setup and discuss the results of dynamic event extraction. We conclude in Section 6 with some ideas for future works.

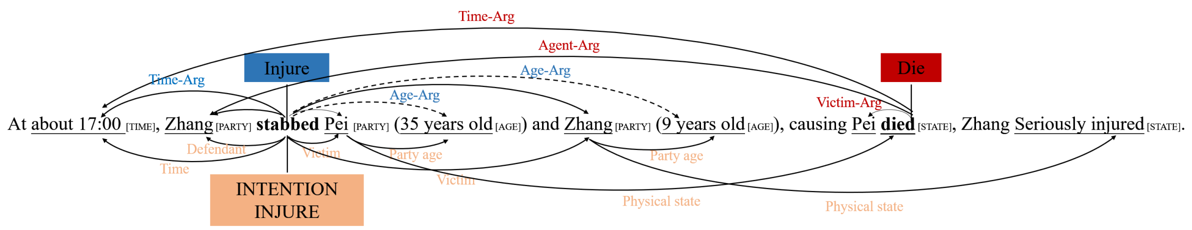


Figure 1: An example of a sentence in the legal domain, which contains legal events.. The upper side shows the traditional event structure. The dotted line indicates the ambiguity of the argument role. The lower side shows the dynamic event structure in this paper, and the subscript represents the type of argument.

2 Related Work

2.1 Event Definition in Legal Domain

Event definition refers to the definition of event types and argument roles relevant to each type of event. The definition of the event type determines what is an event. The definition of the arguments role determines what information the event contains. In ACE event extraction program, legal event type is defined in JUSTICE, with subtypes *(arrest-jail, release-parole, trial-hearing, charge-indict, sue, convict, sentence,fine, execute, extradite, acquit, pardon, appeal)*. Taking *arrest-jail* as example, its related arguments include *Person-Arg, Agent-Arg, Crime-Arg, Time-Arg, Place-Arg*. This method of definition takes the occurrence of events or changes in state as the basis for event types definition, which follows the definition of general events. Based on the ACE event definition method, some works redefine the event types in legal domain according to legal requirements. Maxwell et al. (2009) further analyze the relevant behavior and state in the legal domain and define new legal events. Lagos et al. (2010) define legal events as temporally bounded objects, which highlight the importance of event timing in the legal domain. In addition to defining new event types of the legal domain, some works adjust the arguments in legal events. Ingolfo et al. (2012) abstract the core concepts related to the judgment into the argument roles. Bertoldi et al. (2014) claim that the argument should be the information that experts pay attention to in the legal texts. These methods try to adapt the event definition to different legal requirements. However, due to the complexity of legal requirements, there is no unified method of legal event definition.

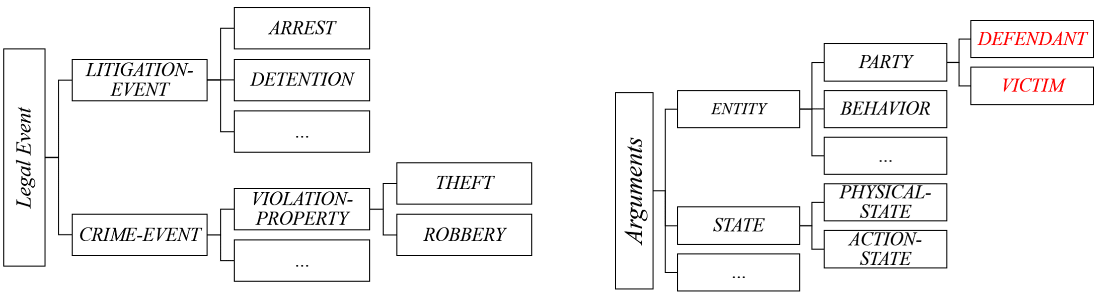
2.2 Event Extraction

Event extraction (EE) is the task of extracting structured event expressions from text. EE can be divided into two subtasks: event detection (ED) and event argument extraction (EAE). ED detects event triggers from text and classify the type they are. EAE aims to identify event arguments in text and classify the roles they play in an event. There are two main approaches to EE: (i) the pipelined approach that first performs trigger detection and then identifies arguments base on the results of trigger detection. Yubo et al. (2015) construct event extraction models through dynamic multi-pooling convolutional neural networks, this work is an influential neural network EE model. Sha et al. (2016) use the dependency between argument roles to construct regularization constraints to improve the accuracy of EAE. The latest pipeline event extraction approach is to treat event extraction as a sequence labeling task and uses the most advanced pre-trained language model to encode text (Yang et al., 2019; Wadden et al., 2019; Tian et al., 2019). (ii) the joint approach that treats event extraction as a structure extraction task, and predicts triggers and corresponding arguments at the same time. In the model proposed by Nguyen et al. (2016), a memory matrix is defined to store the trigger words and arguments that appear before to help the last arguments and trigger words extraction. Some joint approaches use dependency syntactic features to strengthen the semantic association between arguments and triggers (Sha et al., 2018; Liu et al., 2018; Li et al., 2019b). These methods all train the model by maximizing the joint probability of triggers and arguments. Li et al. (2019c) formulate manual rules for the extraction process and convert parameter optimization into an integer linear programming problem to improve the accuracy of extraction. Yang et al. (2016) convert the event into an event tree, and defines the EE task as the problem of extracting the optimal tree.

3 Dynamic Hierarchical Event Structure for Legal Event

In this section, we introduce the definition of our legal dynamic hierarchical event (DHE). We first define the legal event as follows:

- **Legal event**. *a legal event is a specific occurrence related to the judicial process involving key elements.*



(a) Hierarchical Event Types.

(b) Hierarchical Argument Types. The red font indicates the role of the argument.

Figure 2: A Part of Hierarchical Event Types and Hierarchical Argument Types.

Then we define the hierarchical event to manage all legal events in an organized manner. The structure of the hierarchical event is shown in Figure 2 (a).

- **Hierarchical event**. *In legal domain, an* ***event type*** T_1 *is a collection of events which are a specific occurrence with specific key elements. If another event type* T_2 *is a proper subset of* T_1, T_2 *is called a* ***sub-type*** *of* T_1. *All event types are organized in an orderly manner according to the sub-type relation which contitute* ***hierarchical event***.

As shown in Figure 2 (a), *CRIME-EVENT* is the collection of events that describe the defendant's specific criminal behavior and implementation details. *VIOLATION-PROPERTY* is a sub-type of *CRIME-*

EVENT, which represents the events that are violations of the country's property or citizens. The details of the hierarchical event are shown in Appendix A.1.

Arguments are the elements in the legal event (e.g., parties, locations). We expand the hierarchy argument structure (Wang et al., 2019) to cover all elements in the legal event.

- **Hierarchical argument**. *In legal domain, an **argument type** T_1 is a collection of legal elements with common properties. If another argument type T_2 is a proper subset of T_1, T_2 is called a **sub-type** of T_1. All argument types are organized in an orderly manner according to the sub-type relation which conduct **hierarchical argument**.*

For example, in Figure 2 (b)*PARTY* is a sub-type of *ENTITY*. In addition, in order to include all legal key elements in the hierarchical argument, we define two special argument types.

- ***STATE***. *STATE* arguments are the objective state of an event or other argument. Its sub-types contain *PHYSICAL-STATE* and *ACTION-STATE*.
- ***BEHAVIOR***. *BEHAVIOR* is a sub-type of *ENTITY*, *BEHAVIOR* arguments are the behaviors of independent significance in legal events.

Defining the *BEHAVIOR* argument can prevent fragmentation of events and make the existing event more coherent. For example, *escape* is a specific *BEHAVIOR*, it is an important component of the defendant's criminal facts, and has a significant impact on the verdict. But *escape*is meaningless as an independent event. If such behaviors are defined as events separately, the event types will be redundant, and the relationship between these pieces of information and existing events (e.g., CRIME-EVENT) will be missing. Considering that these behaviors are often expressed in short terms and are independent of other information in legal documents, we abstract it as *BEHAVIOR* argument. The details of the hierarchical argument are shown in Appendix A.2.

Next, we should define an event structure to represent the relation between events and arguments. The traditional approach is to assign a role to each argument related to the event. However, there are close links between many arguments in legal events. As shown in Figure 1, there is a one-to-one correspondence between ages and parties. However, traditional event structure has no way to reflect this correspondence. In the case of Figure 1, traditional event structure leads to ambiguity. In fact, in linguistics, a dynamic event structure theory (Pustejovsky, 2013) is proposed for this situation. In order to accurately retain the key elements in the legal event, we design a dynamic event structure to represent legal events.

- **Dynamic event structure**. *A is an argument, B is an event (or argument), if A is an element of B, A is called a **child-argument** of B, B is called the **father node** of A, the relation between A and B called A's **role** in B. We define the **dynamic event structure** as a structure that consists of an event and several arguments, where the event has at least one child-argument, and each argument is the child-argument of the event or an argument.*

We define the dynamic event structure for event types without sub-type. As Wang et al. (2019) did, we define possible roles for the lowest level argument. For example, in Figure 1, the *PARTY* arguments have child-argument *AGE* and *STATE* with role *PARTY-AGE* and *PHYSICAL-STATE*. This method changes the fixed event structure into dynamics and improves the expressive power of the event structure. The details of the dynamic event structure can be found in Appendix A.3.

4 Pedal Attention based Joint Hierarchical Event Extraction Model

In this section, we introduce a novel extraction model for dynamic event structure named **P**edal **A**ttention based **J**oint **H**ierarchical **E**vent **E**xtraction (**PAJHEE**). Traditional event structure is a special case of dynamic event structure, so PAJHEE also suitable for the extraction of traditional event structure. The overall architecture of PAJHEE is shown in Figure 3.

For a given sentence $S = x_1, x_2, \ldots, x_n$ with lenght n, where x_i is the i-th token, PAJHEE extract the dynamic event in S. Our model consists of the following modules: (1) candidate argument extractor that extracts candidate argument mentions $A = a_1, a_2, \ldots, a_k$ and the correspond argument type $A^t = a_1^t, a_2^t, \ldots, a_k^t$ from S, (2) hierarchical event feature construction module that constructs the hierarchical event type features of S, (3) pedal attention mechanism that extracts the semantic relationship between

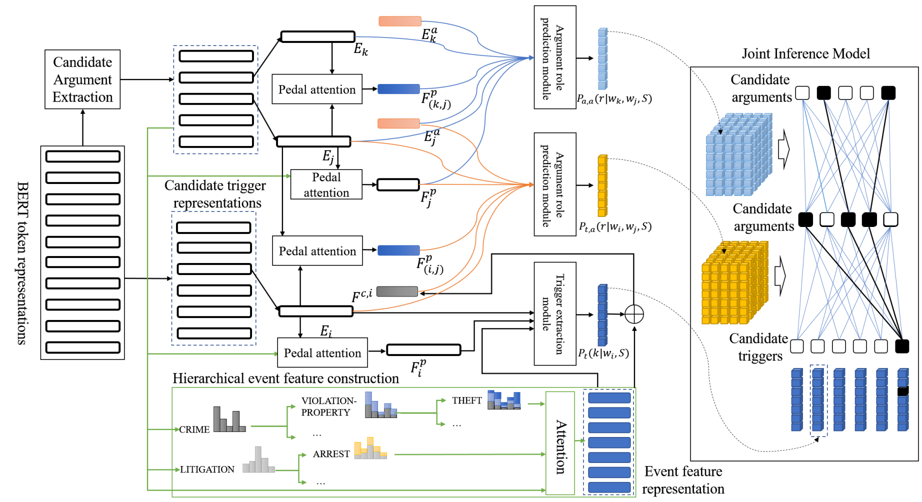


Figure 3: Overall architecture of dynamic hierarchical event extraction.

the arguments and the triggers, (4) trigger extraction module that predicts the labels of all candidate triggers [2], (5) argument role prediction module that predicts the *event-argument* roles and the *argument-argument* roles in dynamic event structure, (6) joint inference model that generates the joint probability for training and extraction.

4.1 Feature Representation

In this paper, we transform discrete features into continuous vectors as inputs of our model. All features are transformed into the following vectors:

- The word representation of candidate triggers and arguments: We use the output of BERT(Devlin et al., 2018) as the word representation. If a word contains more than one tokens, we aggregate the vectors of tokens in this word by average pooling. We use E_i to represent the word representation of the word w_i.
- The argument type embedding: We encode argument type T_i^a as a real-value vector by looking up the randomly initialized position embedding table. Then, we use E_i^a to represent the embedding of T_i^a.
- The event type embedding: Similarly to the argument type embedding, we use real-valued vector E_i^e to represent the event type embedding of event type T_i^e.
- The dependency syntax edge embedding: we transform an edge type T_i^d in dependency parse to a real-valued vector E_i^d by looking up a trainable embedding table.

4.2 Candidate Argument Extraction

Since there are no candidate arguments given in advance, we first extract the candidate arguments in S. Argument mentions are annotated in the BIO annotation schema. We extract candidate arguments and predict their types from the sentence through a BERT-based sequence annotation model. Candidate argument extraction is more sensitive to local semantics and is independent of event structure. To improve the recall of candidate argument extraction, we enhance the data through sentence splitting and reorganization. After this process, we get candidate argument mention $A = a_1, a_2, \ldots, a_K$ and the argument type $A^t = a_1^t, a_2^t, \ldots, a_K^t$.

4.3 Hierarchical Event Feature Construction

The superordinate event type feature is important in trigger classification. We use a hierarchical attention mechanism to construct features of each event type. For a event type T_i^e and the candidate argument $A = a_1, a_2, \ldots, a_K$, we use scaled dot-product attention (Vaswani et al., 2017) to generate the T_i^e's

[2]We enumerate every noun, verb, and adjective in the sentence as candidate triggers

attention to each candidate argument.

$$\hat{W}_i^e = Attention(E_i^e, [E_{a_1}, \dots, E_{a_K}]) \tag{1}$$

where E_{a_j} is the word representation of a_j, $\hat{W}_i^e \in \mathbf{R}^k$ is the attention weight to each argument. We make each T_i^e inherit the attention weight of its superordinate type (if the superordinate type exists). Let the superordinate type of T_i^e is T_i^s and its attention weight to arguments is W_i^s. We recursively construct the attention weight of each type in the following way,

$$W_i^e = \begin{cases} \hat{W}_i^e & T_i^e \text{ has no superordinate type,} \\ (\hat{W}_i^e + W_i^s)/2 & T_i^s \text{ is the superordinate type of } T_i^e. \end{cases} \tag{2}$$

Then we use following method construct the feature of T_i^e,

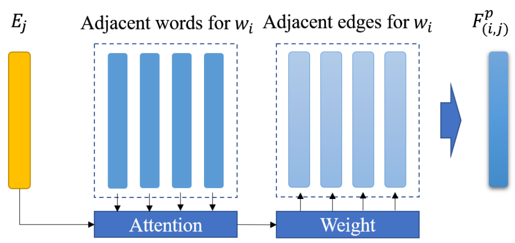
$$F_i^e = \sum_{j=1}^{K} (W_{i,j}^e (M_e E_{a_j} + b_e)) \tag{3}$$

where $W_{i,j}^e$ means the j-th element of W_i^e, M_e and b_e is the parameter of linear transformation.

4.4 Pedal Attention Mechanism

In a long sentence, two reasons make it difficult to capture the semantic relationship between two words. First, the distance between the two words is too far. Second, the two words are related by pronouns. We propose a novel pedal attention mechanism to capture the semantic relationship between two words. The structure of the pedal attention mechanism is shown in the figure on the right.

Structure of pedal attention mechanism



For two words w_i and w_j, let $N^i = N_1^i, N_2^i, \dots, N_l^i$ be the set of words adjacent to w_i in dependency parse tree. $D^i = d_1^i, d_2^i, \dots, d_l^i$ is the set of edge between w_i and N^i. We treat N^i as a pedal to construct the semantic relationship between w_i and w_j. A multi-head attention (Vaswani et al., 2017) is used to generate the semantic relation feature between w_j and N^i.

$$F_{(i,j)}^p = Multi\_head(E_j, [E_{N_1^i}, \dots, E_{N_l^i}], [E_{d_1^i}^d, \dots, E_{d_l^i}^d]) \tag{4}$$

where E_j is the word representation of w_j, $E_{N_k^i}$ is the word representation of k-th node in N^i, $E_{d_k^i}^d$ is the embedding of the dependency edge between w_i and N_k^i.

4.5 Trigger Extraction Module

Trigger extraction module trains a multi-class classifier to predict the label of each candidate trigger. We add *other* in event type represent the word is not an event trigger. In the trigger extraction module for current word w_i, we compute a feature representation for each event type using the concatenation of the following vectors:

- E_i: the word representation of w_i.
- F_k^e: the k-th event type feature of current sentence.
- F_i^p: the semantic relation between w_i and all candidate arguments.

where F_i^p is calculate by max pooling along the relation feature between w_i and candidate arguments.

$$F_i^p = max\_pooling([F_{(i,j)}^p]_{j \in [1,K]}) \tag{5}$$

where $F_{(i,j)}^p$ is the relation feature between w_i and a_j which is calculate by pedal attention mechanism. Then $[E_i, F_k^e, F_i^p]$ is fed into a feed-forward neural network NN^k with a softmax layer in the end to compute the probability $P(k|w_i, S)$.

$$P_t(k|w_i, S) = NN_k^t([E_i, F_k^e, F_i^p]) \tag{6}$$

where k is a event type with no sub-type.

4.6 Argument Role Prediction Module

In dynamic event structure, argument roles contain *event-argument* roles and the *argument-argument* roles. We use two models with the same structure to predict these two argument roles. Take *event-*

argument role prediction as an example. For a given trigger w_i and candidate argument w_j, we generate the feature representation for role prediction by concatenating the following vectors:

- E_i and E_j: the word representation of current trigger and argument.
- $F^p_{(i,j)}$: the semantic relation between w_i and w_j calculated by pedal attention mechanism.
- F^p_j: $F^p_j = max\_pooling([F^p_{(j,m)}]_{m\in[1,K]\bigwedge m\neq j})$ is the semantic relation between w_j and other arguments. Where $F^p_{(j,m)}$ is the semantic relation representation between w_j and a_m calculated by pedal attention mechanism.
- $F^{c,i}$: the event type feature of current trigger. $F^{c,i} = \sum_k (P(k|w_i, S)E^e_k)$, where k is the event type without sub-type.
- E^a_j: the argument type representation of w_j.

The we use a feed-forward neural network $NN^{t,a}$ with a softmax layer in the end to predict the probability $P_{t,a}(r|w_i, w_j, S)$ over the possible argument roles:

$$P_{t,a}(r|w_i, w_j, S) = NN^{t,a}([E_i, E_j, F^p_{(i,j)}, F^p_j, F^{c,i}, E^a_j]) \tag{7}$$

For two arguments w_i and w_j, we simple replace event type feature $F^{c,i}$ with E^a_i, and use a new neural network $NN^{a,a}$ to predict the probability $P_{a,a}(r|w_i, w_j, S)$:

$$P_{a,a}(r|w_i, w_j, S) = NN^{a,a}([E_i, E_j, F^p_{(i,j)}, F^p_j, E^a_i, E^a_j]) \tag{8}$$

4.7 Joint Inference Model

We tailor a joint inference model to add the global constraint of dynamic event structure. The structure of joint inference is shown in Figure 3.

We abstract the dynamic event structure as a tree, with the trigger word as the root node and the arguments as the nodes. The edge between the nodes represents the argument role of the child node. We define the association probability matrix of $M^{t,a}$ and $M^{a,a}$. $M^{t,a}_{i,j}$ represents the probability that i-th type event contains a j-th type argument. $M^{a,a}_{i,j}$ represents the probability that i-th type argument contains a j-th type child-argument. $M^{t,a}_{i,j}$ and $M^{a,a}_{i,j}$ are trainable parameters. We define the probability of each edge as the association probability times the role prediction probability. Then we calculate the joint probability of a dynamic event as follows,

$$\begin{aligned} P(event|S) = P_t(k|w_i, S) &\prod_{w_j\in\mathbf{A}(w_i)} (M^{t,a}_{i,j} P_{t,a}(r_j|w_i, w_j, S) \\ &\prod_{w_m\in\mathbf{A}(w_j)} (M^{a,a}_{j,m} P_{a,a}(r_m|w_j, w_m, S)\ldots)) \end{aligned} \tag{9}$$

where $\mathbf{A}(w)$ represent the child-argument set of w. We maximize the log-likelihood $log(P(event|S))$ to compute the model parameters during the training phase. During prediction phase, we construct the event tree with the largest weight through a greedy algorithm to extract the entire event. The weight of each edge is the log-likelihood of its probability.

5 Experiments

5.1 Experimental Setup

Dataset. We manually labeled a Chinese legal event extraction dataset from a Chinese legal documents corpus [3] by an open-source annotation tool [4]. The dataset contains 2380 instances with 11 pre-defined event types, 26 pre-defined event-argument roles, and 17 pre-defined argument-argument roles. 7 Masters of Laws participated in the labeling process and took one month to complete.

Contrasted models. We select the following state-of-the-art methods for comparison: (1) DMCNN (Yubo et al., 2015) extracts sentence-level features by dynamic multi-pooling CNN; (2) DBRNN (Sha et al., 2018) extracts event triggers and arguments by dependency-bridge RNN; (3) PLMEE (Yang et al., 2019) explores pre-trained language model for event extraction. In order to verify the effectiveness of

[3] http://wenshu.court.gov.cn

[4] http://brat.nlplab.org/about.html

hierarchical event feature and pedal attention mechanism, we set up the following models for comparison: (1) BERT-base only uses the word representation output by BERT for trigger extraction and role prediction; (2) JHEE joins hierarchical event feature based on BERT-base; (3) PAJEE joins the pedal attention mechanism on the basis of BERT-base; (4) PAJHEE is the model with both hierarchical event feature and pedal attention mechanism. For a fair comparison, all candidate arguments are generated by our candidate argument extraction module, and only the final result is evaluated.

Training setup and metric. We randomly select 30% instances from the labeled dataset as blind test data and train all models on the left date. All state-of-the-art models follow their optimal parameter settings. We use the open-source dependency syntax analysis tool on Language Technology Platform[5] (LTP) to build the dependency syntax trees of instances. We set the embedding dimension of the argument category and dependency syntax to 100, The hidden layer size of the multi-head attention mechanism is 256, and update parameter through gradient descent over shuffled mini-batches with the Adadelta (Zeiler, 2012) update rule with 0.00001 as the learning rate. we use the following criteria to judge the correctness of each predicted event: (1) A trigger is correct if its event type and offsets match those of a reference trigger. (2) An argument is correctly identified if its father node and offsets match those of any reference argument mentions. We divide the child-arguments of the event and the child-arguments of the argument into two sets for evaluation. (3) An argument is correctly classified if its father node, offsets, and argument role match any of the reference argument mentions. Finally, we use Precision (P), Recall (R), and F measure (F_1) as the evaluation metrics.

5.2 Overall results

	Trigger Identification (%)			Trigger Classification (%)			Argument (Event) Identification (%)			Argument (Event) Classification (%)			Argument (Argument) Identification(%)			Argument (Argument) Classification (%)		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
DMCNN	85.5	87.6	85.7	84.8	80.2	82.4	79.9	85.5	81.8	74.7	77.7	75.6		N/A			N/A	
DBRNN		N/A		89.2	82.7	83.93	84.3	87.6	84.0	78.1	84.7	79.2		N/A			N/A	
PLMEE	90.1	96.4	92.7	86.0	88.5	86.37	90.9	82.1	85.0	85.4	80.6	80.9		N/A			N/A	
BERT-base	95.4	95.3	94.9	89.7	90.0	89.8	94.2	87.6	88.9	74.8	85.9	79.9	77.5	91.2	81.4	64.6	90.3	75.5
JHEE	94.2	99.0	96.5	92.6	94.3	93.1	86.0	90.5	87.9	85.2	80.7	82.8	80.9	86.3	82.6	76.4	75.8	74.0
PAJEE	97.1	95.4	96.3	91.9	92.7	92.3	92.1	93.5	92.8	88.9	89.5	88.2	98.0	78.6	87.2	95.7	76.6	85.2
PAJHEE	97.6	97.3	**97.4**	93.4	95.1	**94.2**	92.7	94.4	**93.5**	88.7	90.6	**89.6**	98.6	78.4	**87.4**	95.7	77.2	**85.4**

Table 1: Overall performance on test data.

Table 1 shows the overall extraction results on our legal event extraction dataset. As is shown, in both the trigger extraction task and the argument extraction task, PAJHEE has achieved the best results among all the compared methods. The BERT-base model is better than all state-of-the-art models. It illustrates that our joint inference model can obtain better prediction results through global constraints. The performance of PLMEE is better than DMCNN and DBRNN, which shows the effect of the pre-trained language model on legal domain event extraction. The classification performance in models without hierarchical event features and pedal attention mechanism is significantly worse than identification. This shows that the traditional method can not get the semantic relationship in the legal text well. Then we will show the effect of the hierarchical event feature and pedal attention mechanism.

5.3 Effect of hierarchical event feature

After using the hierarchical event (HE) feature, trigger identification performance and classification performance have been improved. It proves the effectiveness of the HE feature of sentence-level event feature extraction. Compared with BERT-base, joining HE feature achieves a 1.6% F_1 increase on trigger identification and 3.3% F_1 increase on trigger classification. This shows that for the identified trigger words, the HE feature leads to correct classification. Moreover, the HE features it lead to a 2.9% F_1 increase on argument (Event) classification, which indicates that the HE feature can help correctly classify the role of event's argument.

[5]https://github.com/HIT-SCIR/ltp

5.4 Effect of pedal attention mechanism

As shown in Table 1, the model with pedal attention (PA) achieves F_1 improvements of 8.3% and 9.7% over Bert-base on argument (Event) classification and argument (Argument) classification. PA can also support trigger classification by constructing the semantic relationship between trigger words and candidate arguments. It proves that PA can better extract the semantic relation between two words. PA-JEE achieves F_1 improvements of 9.0% over DBRNN on argument (Event) classification. This occurs because DBRNN suffers from the problem that related words in legal sentences often have no direct dependency so that the dependency parse information in DBRNN has little effect. It illustrates that PA can overcome the problem and construct word semantic relations in complex contexts.

5.5 Case Study

To verify whether the HE feature and PA work as we designed, we conduct a case study. We visualize the attention score of HE feature construction and PA mechanism on a sentence randomly sampled from our dataset. The left side of Figure 4 shows the attention score of words at different levels of event types that can reflect their characteristics well. Fine-grained event types need more information than superordinate types. As shown in the right side of Figure 4, during PA, *Geng Li* is associated with **strangled** through *Li's neck*. And it can be inferred that *Geng Li* is the object of **strangled**. This case shows that our method has achieved the desired effect.

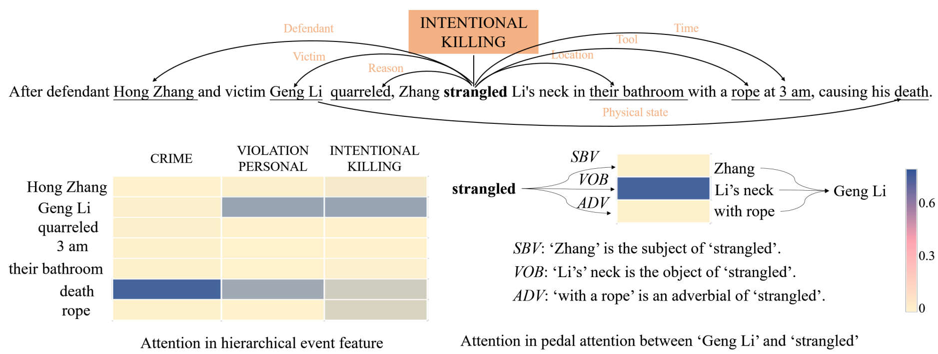


Figure 4: Heatmap of attention scores of hierarchical event feature and pedal attention of a randomly selected sentence.

6 Conclusion and Future Work

This paper analyzes the shortcomings of the traditional event structure in the legal field and defines a dynamic hierarchical event structure that can better express event information. We label a legal event extraction dataset that contains 11 types of events in two major legal-related events: *CRIME* and *LITIGATION*. Then we propose a pedal attention based joint hierarchical event extraction model, which can automatically extract lexical level and event-level features from plain texts. A novel pedal attention mechanism is introduced to capture lexical level semantic relation, and a hierarchical event feature is used to construct the event level feature. The experimental results prove the effectiveness of the proposed method. In the future, we will extend the dynamic event structure to other fields such as finance and biomedicine and apply the pedal attention mechanism in other natural language processing tasks as a general relational feature construction tool.

7 Acknowledgement

Research in this paper was partially supported by the National Key Research and Development Program of China under grants (2018YFC0830200, 2017YFB1002801), the Natural Science Foundation of China grants (U1736204), the Judicial Big Data Research Centre, School of Law at Southeast University.

References

Kevin D Ashley. 2017. *Artificial intelligence and legal analytics: new tools for law practice in the digital age*. Cambridge University Press.

Anderson Bertoldi, Rove Chishman, Sandro José Rigo, and Thaís Domênica Minghelli. 2014. Cognitive linguistic representation of legal events. In *Proceedings of COGNITIVE*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Silvia Ingolfo, Ivan Jureta, Alberto Siena, A Susi, A Perini, and John Mylopoulos. 2012. Legal compliance of roles and requirements.

Nikolaos Lagos, Frederique Segond, Stefania Castellani, and Jacki O'Neill. 2010. Event extraction for legal case building and reasoning. In *Proceedings of ICIIP*, pages 92–101. Springer.

Chuanyi Li, Yu Sheng, Jidong Ge, and Bin Luo. 2019a. Apply event extraction techniques to the judicial field. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing and ACM International Symposium on Wearable Computers*, pages 492–497.

Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. 2019b. Biomedical event extraction based on knowledge-driven tree-lstm. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1421–1430.

Wei Li, Dezhi Cheng, Lei He, Yuanzhuo Wang, and Xiaolong Jin. 2019c. Joint event extraction based on hierarchical event schemas from framenet. *IEEE Access*, 7:25001–25015.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. *arXiv preprint arXiv:1809.09078*.

K Tamsin Maxwell, Jon Oberlander, and Victor Lavrenko. 2009. Evaluation of semantic events for legal case retrieval. In *Proceedings of WSDM Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 39–41.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of ACL*, pages 300–309.

James Pustejovsky. 2013. Dynamic event structure and habitat theory. In *Proceedings of International Conference on Generative Approaches to the Lexicon*, pages 1–10.

Lei Sha, Jing Liu, Chin-Yew Lin, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Rbpb: Regularization-based pattern balancing method for event extraction. In *Proceedings of ACL*, volume 1, pages 1224–1234.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Proceedings of AAAI*.

Can Tian, Yawei Zhao, and Liang Ren. 2019. A chinese event relation extraction model based on bert. In *Proceedings of (ICAIBD)*, pages 271–276. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Advances in neural information processing systems*, pages 5998–6008.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. Hmeae: Hierarchical modular event argument extraction. In *Proceedings of EMNLP-IJCNLP*, pages 5781–5787.

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7:173111–173137.

Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. *arXiv preprint arXiv:1609.03632*.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of ACL*, pages 5284–5294.

Chen Yubo, Xu Liheng, Liu Kang, Zeng Daojian, Zhao Jun, et al. 2015. Event extraction via dynamic multi-pooling convolutional neural networks.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

A Appendix A. The Definition of Dynamic Hierarchical Event Structure

A.1 The Definition of Hierarchical Event

Category		Type
CRIME	*ENDANGERING PUBLIC SECURITY*	*RECKLESS DRIVING*
		TRAFFIC OFFENSE
	PROPERTY INFRINGEMENT	*LARCENY*
		FRAUD
		ROBBERY
	VIOLATION PERSONAL	*INTENTION INJURY*
		INTENTION KILLING
LITIGATION		*ARREST*
		DETENTION
		BAIL
		TRIAL

Table A. Hierarchical Event

Table A shows the hierarchical relationship between legal events in the legal documents. The events in the legal documents can be divided into criminal events (*CRIME*) and litigation-related events (*LITIGATION*). The details of the two event categories are as follows:

A.1.1 CRIME

CRIME. A crime committed by a perpetrator that violates the provisions of the criminal law and constitutes a crime. According to the definition and classification of criminal law in mainland China [6], we selected three representative crimes: endangering public safety, property infringement, and violation of personal rights for extraction. **ENDANGERING PUBLIC SECURITY** *ENDANGERING PUBLIC SECURITY*. A general crime, which objectively manifests as various acts that endanger public safety.
We selected the following two typical sub-types of crimes from *ENDANGERING PUBLIC SECURITY*:

- *RECKLESS DRIVING* refers to driving a car on the road: chasing racing, drunk driving, overloading, speeding, and other actions that endanger public safety.
- *TRAFFIC OFFENSE* refers to the criminal act of violating road traffic management laws and regulations, causing serious traffic accidents, causing serious injury or death, or causing heavy losses to public and private property being prosecuted for criminal responsibility according to law.

PROPERTY INFRINGEMENT *PROPERTY INFRINGEMENT*. A criminal act of seizing public and private property for the purpose of illegal possession or deliberately destroying public and private property.
We extract the following three common crimes in crimes against property *PROPERTY INFRINGEMENT*:

- *LARCENY* refers to the act of illegal possession, the theft of public and private property objects or multiple theft, household theft, theft with a weapon, and pickpocketing of public and private property.
- *FRAUD* refers to the act of deceiving large amounts of public and private property for the purpose of illegal possession, using fictitious facts or concealing the truth.
- *ROBBERY* is an act of illegal possession, using violence, coercion or other methods against the owner or custodian of property to steal public and private property forcefully.

VIOLATION PERSONAL *VIOLATION PERSONAL*. A general crime, which objectively manifests as various acts that endanger public safety.
We selected common intentional injurie and intentional killing as representatives of *VIOLATION PERSONAL*:

- *INTENTION INJURY* refers to the crime convicted of intentionally illegally harming the health of others.

[6] http://www.npc.gov.cn/wxzl/wxzl/2000-12/17/content_4680.htm

- *INTENTION KILLING* refers to the act of deliberately depriving others of their lives.

A.1.2 LITIGATION

LITIGATION. A series of actions taken against the defendant's criminal behavior, including: arrest, detention, bail, and trial.

- *ARREST* refers to a case where a public security organ, a people's procuratorate, or a people's court deprives a criminal suspect or defendant of an act that impedes criminal proceedings, evades investigation, prosecution, trial, or social danger, and deprives him of his personal freedom under the law and is detained. Kind of coercive measures.
- *DETENTION* refers to a compulsory state that restricts the personal freedom of a criminal suspect or defendant detained or arrested within a certain period of time.
- *BAIL* release refers to the criminal defendant detained in the judiciary providing security and granting release.
- *TRIAL* means hearing a case and giving a judgment. It is an important part of law enforcement power.

A.2 The Definition of Hierarchical Argument

Category	Type
ENTITY ARGUMENT	PARTY
	ITEM
	TIME
	LOCATION
	ORGANIZATION
	TERM
	CRIMINAL CHARGE
	BEHAVIOR
STATE ARGUMENT	PHYSICAL STATE
	BEHAVIORAL STATE
ATTRIBUTE ARGUMENT	ITEM ATTRIBUTE
	PARTY ATTRIBUTE
	CRIME ATTRIBUTE
	PENALTY/ENFORCE ATTRIBUTE

Table B. Dynamic Argment

Table B shows all entity argument types involved in the hierarchical event. According to the hierarchical relationship of arguments in the event, these arguments can be divided into three categories: entity argument, state argument, and attribute argument. The details of the three entity categories are as follows:

A.2.1 ENTITY ARGUMENT

ENTITY ARGUMENT. Something that can exist independently, as the basis of all attributes and the origin of all things.

The entity argument contains the following six entities: partiy, item, time, location, organization, term, criminal charges, and behavior.

- *PARTY* is a person who enters a lawsuit because of a dispute over the rights and interests in the substantive law or has a direct relationship with a specific legal fact and is bound by a court decision.
- *ITEM* refers to the items involved in the crime.
- *TIME* refers to the time of the event.
- *LOCATION* refers to the location where the crime occurred.
- *ORGANIZATION* refers to law enforcement agencies, procuratorates, and courts.
- *TERM* is an important basis for conviction and sentencing.
- *CRIMINAL CHARGE* is the name or title of the crime, and it is a high-level summary of the essential characteristics or main characteristics of the crime.

- *BEHAVIOR* refers to the appearance of activities that are controlled by ideas. This article refers to the actions taken by the parties before or after the crime.

A.2.2 STATE ARGUMENT

STATE ARGUMENT. The form that people or things show.

This article mainly concerns the personal status and behavior status of the parties.

- *PHYSICAL STATE* refers to the disability and mental state of the party.
- *BEHAVIORAL STATE* refers to the state of the party's behavior, such as epilepsy and drunkenness.

A.2.3 ATTRIBUTE ARGUMENT

ATTRIBUTE ARGUMENT. The abstract aspects of an object. The nature and relationship of a thing are called the attributes of the thing. The attributes in this article are in-depth descriptions of entity arguments and events, which mainly include item attributes, parties attributes, crime attributes, and penalty/enforce attributes.

- *ITEM ATTRIBUTE* is an abstract characterization of items, such as value, length, and diameter. The article attributes mainly relate to the value of the article.
- *PARTY ATTRIBUTE* refers to the attribute information of the party's ethnicity, age.
- *CRIME ATTRIBUTE* describes the social impact brought by criminal behavior, whether the defendant's criminal methods are cruel, and the reasons for the occurrence of criminal incidents.
- *PENALTY/ENFORCE ATTRIBUTE* is a more detailed description of the punishment of the defendant's criminal facts.

A.3 The Definition of Dynamic Event Structure

A.3.1 Child-argument of event

We define the collection of child-arguments for the event types without sub-types, the format of the collection is [*event type*]:[(*argument type*, *argument role*)] which means the collection of event types share a set of child-arguments.

- [*RECKLESS DRIVING*, *TRAFFIC OFFENSE*]: [(*PARTY*, *defendant/victim*), (*ITEM*, *tool/property*), (*TIME*, *time*), (*LOCATION*, *location*), (*STATE*, *driving state*), (*ATTRIBUTE*, *influence*)].
- [*LARCENY*, *FRAUD*, *ROBBERY*, *INTENTION INJURY*, *INTENTION KILLING*]: [(*PARTY*, *defendant/victim*), (*ITEM*, *tool/property*), (*TIME*, *time*), (*LOCATION*, *location*), (*ATTRIBUTE*, *influence/means/tense*)].
- [*ARREST*, *DETENTION*]: [(*PARTY*, *defendant*), (*ORGANIZATION*, *enforcement-organ /accusation*), (*TIME*, *time*), (*LOCATION*, *location*), (*CRIMINAL CHARGE*, *criminal-charge*)].
- [*BAIL*]: [(*PARTY*, *defendant/guarantor*), (*ORGANIZATION*, *enforcement/accusation*), (*TIME*, *time*), (*LOCATION*, *location*), (*CRIMINAL CHARGE*, *criminal charge*)].
- [*TRIAL*]: [(*PARTY*, *defendant/guarantor*), (*ORGANIZATION*, *judicial-organ*), (*TIME*, *time*), (*LOCATION*, *location*), (*CRIMINAL CHARGE*, *criminal-charge*), (*TERM*, *term*), (*BEHAVIOR*, *penalty-behavior/enforce-behavior*)].

A.3.2 Child-argument of argument

We define the collection of child-arguments for the argument types without sub-types, the format of the collection is *argument type*:[(*argument type*, *argument role*)].

- *PARTY*:[(*PHYSICAL STATE*, *disability/mental*), (*PARTY ATTRIBUTE*, *previous/subjective/age*), (*BEHAVIOR*, *before-crime/after-crime*), (*TIME*, *birth-time*)]
- *ITEM*:[(*ATTRIBUTE*, *value*)]
- *BEHAVIOR*:[(*PENALTY/ENFORCE ATTRIBUTE*, *amount/term*), (*ITEM*, *related-item*)]