Journal of Database Management
Volume 34 • Issue 1

Semi-Supervised Event Extraction Incorporated With Topic Event Frame

Gongqing Wu, Hefei University of Technology, China*
Zhuochun Miao, Hefei University of Technology, China
Shengjie Hu, Hefei University of Technology, China
Yinghuan Wang, Hefei University of Technology, China
Zan Zhang, Hefei University of Technology, China*
Xianyu Bao, Shenzhen Academy of Inspection and Quarantine, China

ABSTRACT

Supervised Meta-event extraction suffers from two limitations: (1) The extracted meta-events only contain local semantic information and do not present the core content of the text; (2) model performance is easily degraded because of labeled samples with insufficient number and poor quality. To overcome these limitations, this study presents an approach called frame-incorporated semi-supervised topic event extraction (FISTEE), which aims to extract topic events containing global semantic information. Inspired by the frame-based knowledge representation, a topic event frame is developed to integrate multiple meta-events into a topic event. Combined with the tri-training algorithm, a strategy for selecting unlabeled samples is designed to expand the training sets, and labeling models based on conditional random field (CRF) are constructed to label meta-events. The experimental results show that the event extraction performance of FISTEE is better than supervised learning-based approaches. Furthermore, the extracted topic events can present the core content of the text.

KEYWORDS

CRF, Meta-event Extraction, Semantic Information, Semi-supervised Learning, Sequence Labeling, Topic Event Extraction, Topic Event Frame, Tri-training

INTRODUCTION

Rapid advancements made in Internet technologies have resulted in massive volumes of data in the form of text digitization. In the light of ever-increasing textual data, a technology that can automatically mine useful information from text is urgently needed. In this context, information extraction technology emerged at a historic moment and has been widely used (Fiori et al., 2014). Event extraction is the

DOI: 10.4018/JDM.318453

*Corresponding Author

most challenging operation in information extraction, which aims to automatically extract information that users are interested in from unstructured text and present it in the form of structured events (Ahn, 2006). Event extraction has given a huge impetus to the development of knowledge graph construction (Bu et al., 2021), text mining (Lyu & Liu, 2021), information retrieval (Feng et al., 2021), etc.

At present, event extraction can be divided into meta-event extraction and topic event extraction, where a meta-event only describes simple actions or state changes, whereas a topic event describes the developmental processes of things. Event extraction broadly involves two subtasks: trigger extraction and event argument extraction, where a trigger refers to a keyword that can clearly express the occurrence of an event, and an event argument refers to the related descriptions such as time, place, and participant of the event. An event can be detected, and its type can be determined by identifying the trigger. Each event type is provided with a unique representation frame, and each relevant entity in the sentence determines whether it is an event argument based on the frame, and if so, its argument role can be determined.

Traditional meta-event extraction approaches mainly adopt pattern matching and machine learning. The former refers to the detection and extraction of meta-events under the guidance of meta-event templates, which show effective performance in specific fields. However, building meta-event templates is time-consuming and laborious; furthermore, building a general meta-event template is difficult. The latter is modeled as a multi-classification task or sequence labeling task, after which the extracted features are used as model inputs to complete the meta-event extraction. However, training models using supervised learning strategy requires large volumes of labeled samples and considering that these labeled samples are generally manufactured by experts, their manufacturing cost is high. When the quantity of labeled samples is small and the categories are unbalanced, the extraction performance of the models decreases. To overcome this limitation, researchers have proposed the adoption of semi-supervised learning strategy (Zhou & Li, 2010) that utilizes a small number of labeled samples and a large number of unlabeled samples to train models. Tri-training (Zhou & Li, 2005) is a classical semi-supervised learning algorithm that adopts bootstrapping to train three classifiers, makes them work together, and expands the training set by constantly introducing new training samples from unlabeled sample set to obtain three classifiers with excellent performance. Because unlabeled samples are cheap and easy to obtain, the use of semi-supervised strategy to train high-performance event extraction models is a current research hotspot.

Compared with sentence-level meta-events, document-level topic events contain richer global semantic information, including multi-facet meta-events, which can present the core content of the text from a global perspective. However, the description information of topic events is scattered in the text, and the existing meta-event extraction approaches cannot meet the demand of topic event extraction, which is a complicated procedure. The difficulty lies in determining all topic-related meta-events within the scope of the document and merging and extracting these meta-events. At present, the event frame or ontology is usually applied in some topic event extraction work to represent each component of the topic event and the relations between them, which has achieved superior results in specific fields. Nevertheless, the existing topic event extraction technologies are not mature enough; especially the intra-textual semantic understanding and cross-textual event extraction need further research.

Supervision-based meta-event extraction approaches suffer from two limitations: one is that the semantic information of meta-events is limited to the sentence level, thus not presenting the core content of the text from a global perspective; the other is that supervised learning requires a large number of labeled samples, and the model performance is generally degraded because of the poor quality and insufficient number of labeled samples. To this end, we present an approach called **frame-incorporated semi-supervised topic event extraction** (FISTEE), which aims to extract topic events containing global semantic information.

The main contributions of the present study are as follows:

1. We design a topic event frame inspired by frame-based knowledge representation, which organizes the meta-events representing different facets of the text to form a topic event.
2. We propose to improve the performance of the meta-event extraction by training the sequence labeling model with not only the human-labeled samples but also the samples automatically selected by tri-training.
3. We generate a trigger table by adopting the entropy-based feature selection algorithm and use triggers to filter irrelevant information to promote the efficiency of the meta-event extraction during the extraction phase.
4. We conduct experiments on real-world datasets to validate the effectiveness of the proposed approach.

The remainder of this study is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed approach. Section 4 presents the experimental results. Finally, Section 5 provides the conclusions and future work.

RELATED WORK

In this section, we introduce related works on event extraction, including meta-event extraction and topic event extraction. In addition, we summarize the related research on tri-training.

Meta-Event Extraction

Meta-event extraction can be divided into two types: pattern matching-based meta-event extraction (Riloff, 1993; Yangarber et al., 2000; Cao et al., 2015; Valenzuela-Escárcega et al., 2015; Li et al., 2016; Cao et al., 2018; Tadesse et al., 2020) and machine learning-based meta-event extraction (Chen & Ji, 2009; Li et al., 2013; He et al., 2014; Chen et al., 2015; Liu et al., 2016; C. Zhang et al., 2016; Z. Zhang et al., 2016; Lin et al., 2018; Liu & Nguyen, 2018; Kodelja et al., 2019; Ahmad et al., 2020; Zhou et al., 2021).

Earlier approaches for meta-event extraction mainly relied on pattern matching. Developed by Riloff (1993), the AutoSlog system, the earliest known event extraction system based on pattern matching, is used for curbing terrorism by extracting terrorist events. AutoSlog exploited a small number of linguistic patterns and a manually labeled corpus to obtain event patterns. Yangarber et al. (2000) designed the ExDisco system that does not need to annotate and pre-categorize the corpus and only needs to formulate a small number of seed patterns for learning automatically to obtain excellent matching patterns. Compared with AutoSlog, ExDisco greatly reduces manual intervention, and its event extraction performance is optimal. Valenzuela-Escárcega et al. (2015) defined a rule-based event extraction frame that is simple, powerful, robust, and fast. They used the frame to develop a grammar for the biochemical domain, which approached human performance. Cao et al. (2015) proposed a pattern expansion approach to import frequent patterns extracted from external corpora to boost event extraction performance. To solve the problem that some frequent expressions involving event triggers do not appear in the training corpus, Cao et al. (2018) introduced expert patterns from TABARI to boost event extraction performance. In addition, many other approaches have been proposed to facilitate automatic pattern construction by designing machine learning algorithms to learn new patterns based on a few seed patterns (Li et al., 2016; Tadesse et al., 2020).

Machine learning-based meta-event extraction can be divided into pipeline-based and joint learning-based models. The former regards trigger extraction and event argument extraction as different sequential processes, and models meta-event extraction as a task pipeline. Trigger extraction can determine the meta-events and their types, and the extraction performance of triggers directly determines the performance of subsequent argument extraction. Chen and Ji (2009) proposed training a trigger labeling model using text features such as lexical, syntactic, and semantic features. However, it cannot recognize unknown triggers that do not appear in the training set. To address this issue, He et al. (2014) used "Synonymy Thesaurus" to expand triggers to help obtain unknown triggers. Liu et al. (2016) designed two types of global information, event-event correlation and topic event correlation, to construct a probabilistic soft logic (PSL) model to assist in the decision-making of the output of each subtask model to determine the final extraction results. C. Zhang et al. (2016) proposed a CRF-based sequence labeling model to identify triggers from complex sentences that can consider the local context of text, achieving a good generalization performance. However, all pipeline-based approaches have a drawback; that is, the error of the previous subtask carries over to the next subtask. The joint learning-based approaches coordinate two subtasks, that is, the uncertain information in the previous subtask is transmitted to the next subtask, and the valuable information generated in the next subtask is allowed to be fed back to the previous subtask. Li et al. (2013) solved the problem of event extraction from the perspective of structured learning and designed a joint learning algorithm by combining sentence-level local and document-level global features. Specifically, they presented a joint frame based on structured prediction that extracts triggers and arguments together so that the local predictions can be mutually improved.

The above-mentioned meta-event extraction approaches are more or less involved in feature engineering, which either rely on existing prior knowledge to design complex features or use NLP tools to extract relevant features. Therefore, they are not only laborious and inefficient, but also generate incorrect information, thereby resulting in a decreased event extraction performance. Compared with traditional machine learning, modern machine learning, namely, deep learning, adopts an end-to-end modeling scheme and avoids feature engineering. Therefore, in recent years, researchers have proposed various deep learning-based meta-event extraction approaches. These approaches can be divided into sequence-based methods and graph-based methods. For the sequence-based methods, Chen et al. (2015) developed a dynamic multi-pooling convolutional neural network (DMCNN) that automatically extracts lexical-level and sentence-level features to obtain the most important information for each part of the sentence. Z. Zhang et al. (2016) proposed a joint event extraction approach based on convolutional neural networks (CNNs) that can extract triggers and arguments simultaneously. Kodelja et al. (2019) constructed a representation of the global context using a bootstrapping approach and integrated the representation into a CNN model for event extraction. These methods input text as sequence data and can automatically learn feature representations. For the graph-based methods, recent work (Cui et al., 2020; Xie et al., 2022; Liu et al., 2021; Huang et al., 2020) employ Graph Convolution Networks and Graph Attention Networks using the dependency graph generated from syntactic dependency-parsers. These methods can capture long-distance dependencies between words related to event extraction. These types of meta-event extraction approaches can not only avoid feature engineering, but also obtain excellent event extraction performance.

Topic Event Extraction

Topic event extraction can be divided into ontology-based topic event extraction (Lee et al., 2003; Chua et al., 2012; Wu et al., 2020) and event frame-based topic event extraction (Wand & Weber, 2017; Storey, 2017; Wu et al., 2020).

As the name suggests, ontology-based topic event extraction is centered on the characteristics of the ontology. It first extracts meta-events and their related entity information from text according to the concepts specified by the ontology and then correlates the entity information of different meta-events based on the relations given by the ontology. It includes three steps: construction of the domain ontology, text semantic annotation, and event extraction. Lee et al. (2003) developed an ontology-based event extraction system that involved an ontology model with four layers, namely, domain, category, event, and extended concept layers. Their experimental results showed that the ontology model was effective for the extraction of Chinese meteorological news events. An event extraction system from a document is generally domain-dependent. To avoid this dependency as much as possible, Sahnoun et al. (2020) proposed applying an open information extraction approach for modeling any event type ontology representation. Their experimental results confirmed the effectiveness of this approach.

Event frame-based topic event extraction approaches show the same effectiveness as ontology-based topic event extraction approaches. A hierarchical and structured frame is designed in place of the ontology to guide topic event extraction. First proposed by Minsky (1974), frames have now become a common knowledge representation approach for describing the outline of related concepts. The basic idea is that the human brain stores a large number of typical scenarios, and when people face new scenarios, a basic empty knowledge structure called a frame is selected from the memory. Wu et al. (2020) classified meta-events according to the information presented by the meta-events in the topic and designed a frame model based on these meta-events, representing the topic events in the form of hierarchical meta-events. They developed CRF-based labeling models to extract topic events from various court verdicts.

Semi-Supervised & Distant-Supervised Event Extraction

In recent years, due to the lack of available training data, semi-supervised and distant supervised methods were gradually proposed.

Due to the successful application of distance supervision in relational extraction tasks, many researchers have also tried to apply distance supervision to the field of event extraction. Zheng et.al. (2019) uses Transformer and sequence annotations for sentence-level entity extraction to obtain the argument of the event and continuously adds event argument to the event table by constructing a directed acyclic graph to complete the extraction of events in the document-level entity extraction phase. Zhu et.al. (2021) developed DE-PNN, an encoder-decoder model for document-level event extraction, which is based on document-level encoding and multi-granular decoding, respectively. However, distance supervision requires aligning the background knowledge base with the accompanying natural language document corpus. For event extraction, such a data source required for distance supervision is often not readily available. Zhou and Zhong (2015) proposed a semi-supervised learning framework based on hidden topics for biomedical event extraction. In this framework, sentences in the unannotated corpus are elaborately and automatically assigned with event annotations based on their distances to these sentences in the annotated corpus. Zajec and Mladenić (2022) iteratively labeled unlabeled data using semi-supervised learning combined with probabilistic soft logic, inferring the pseudo-tokens of each instance from the predictions of multiple base learners. The proposed methodology is applied to Wikipedia pages about earthquakes and terrorist attacks in a cross-lingual setting.

Tri-Training

Co-training, first proposed by Blum and Mitchell (1998), requires the dataset to have two sufficient and redundant views. However, it is often difficult for actual datasets to meet these two conditions. Goldman and Zhou (2000) proposed a collaborative training algorithm that does not require two sufficient and redundant views. They utilized different decision tree algorithms to train two different classifiers, adopted cross-validation to label unlabeled samples, and combined the two learning approaches to form the final prediction. Owing to the extensive use of cross-validation, the algorithm has a high time complexity. Tri-training, which is also a semi-supervised learning algorithm based on collaborative training, does not require sufficient and redundant views or uses different learning algorithms. The difference among the training classifiers is guaranteed by using different sample subsets extracted from the original sample set. Because tri-training has no constraints on the attribute set and the learning algorithm used by the classifier, and does not require cross-validation, it has a wider application scope and higher efficiency.

Generally applied to classification tasks, tri-training is relatively rare in sequence labeling tasks. This is because when performing classification tasks, there is only one classification result, whereas when performing sequence labeling tasks, the labeling result is a sequence. When tri-training is combined with sequence labeling, it is necessary to obtain a consistent label sequence from the labeling results of multiple models as pseudo labels. Chen et al. (2006) developed the consistency calculation approach called *S2A1D* that calculates the consistency rate of labeled results of the two models for a sequence and selects the sequence with the highest consistency rate to add to the pseudo-label sample set. Chou et al. (2016) selected a sample from unlabeled sample set, labeled it with different models, and selected the top *m* label sequences with the highest probability from the labeling results of each model. If the label sequences generated by the two models are exactly the same, then the probability sum of the two is calculated, and the label sequence when the probability sum takes the maximum value is selected as the pseudo label.

FRAME-INCORPORATED SEMI-SUPERVISED TOPIC EVENT EXTRACTION

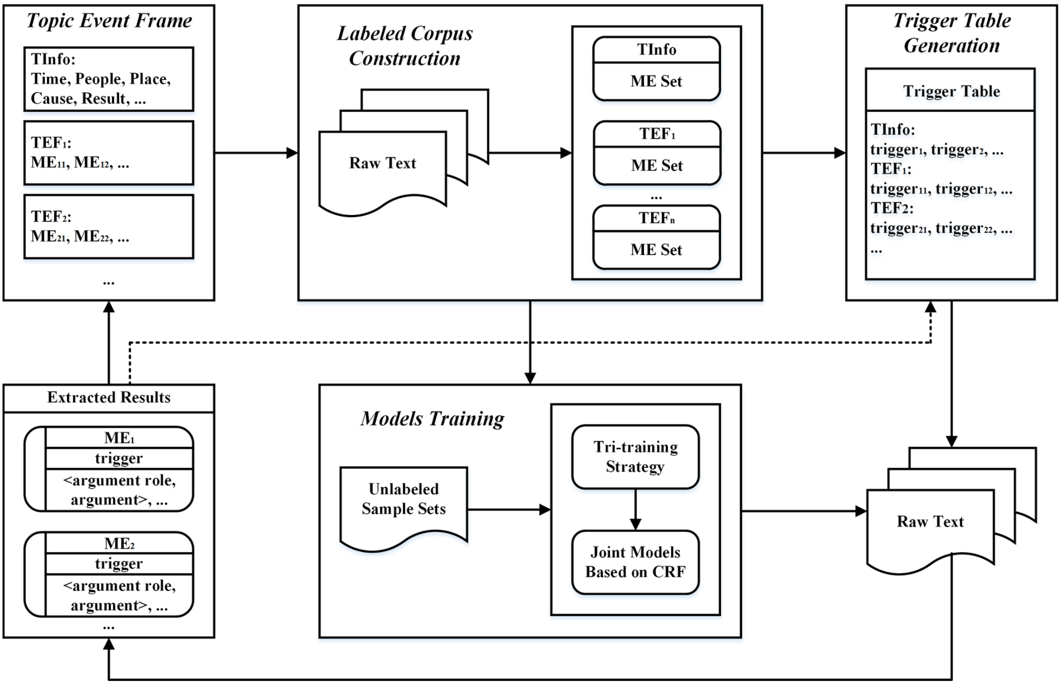
Here we present our design process for topic event extraction from text, which includes two core parts, as shown in Figure 1. The first part includes the steps of frame design, labeled corpus construction, and trigger table generation. The second part is a semi-supervised meta-event extraction based on tri-training.

Topic Event Frame

Meta-events only contain local semantic information and do not present the core content of the text from a global perspective. In contrast, a topic event is composed of multiple states and actions, including multiple meta-events related to text topics, and their global semantic information can effectively present the core content of the text. However, the description information of the topic event is usually scattered in the document; thus, most meta-event extraction approaches hardly meet the needs of topic event extraction. Inspired by the frame-based knowledge representation, we design a topic event frame that aims to integrate multiple meta-events representing different facets of information into a topic event, thereby representing a topic event in the form of meta-event sets. A topic event generally has the following characteristics:

1. **Separation:** A topic event often involves multiple facets of information. A facet refers to a type of meta-event set, and different facets are semantically separated.

Figure 1. The schematic diagram of the overall process of FISTEE



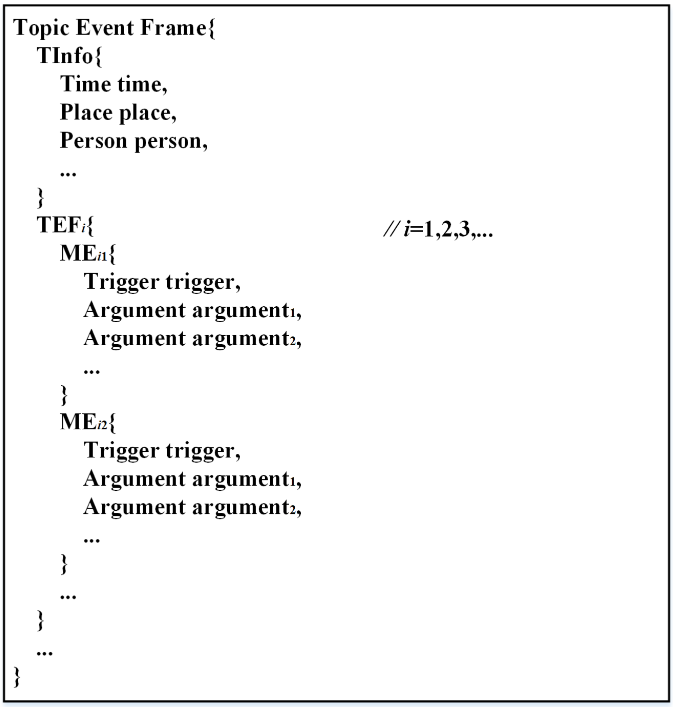
2. **Cohesion:** A topic event includes a topic core meta-event and other facet meta-events. The topic core meta-event describes the topic information, and all other facet meta-events are related to the topic through the topic information.

There exists a close relationship between a topic event and meta-event. Thus, this study designs a meta-event-based knowledge representation frame to describe a topic event. The frame regards a topic event as a multiple meta-event set. The triggers and event arguments in the meta-events are extracted to structure the meta-events, after which different types of meta-events are combined to hierarchically present a topic event. The meta-event types include topic information and facet meta-event types, which are formally defined as follows:

1. **Definition 1:** TInfo (topic information) is a general description of a topic event, including the most basic information of the topic event, such as time, location, and people.
2. **Definition 2:** TEF (topic event facet), TEF_i={ME_{i1}, ME_{i2},...},where TEF_i refers to a facet of the topic event, and "ME_{i1}, ME_{i2},..." in TEF_i denote the same type of meta-events, composed of triggers and event arguments.
3. **Definition 3:** TE (topic event) is described by topic information and multiple facets of the topic event, that is, TE={TInfo, TEF_1, TEF_2, …}.

The topic event frame based on the meta-events is shown in Figure 2. A topic event is a frame structure, and its slot values include topic information and information on various facets. Facet information is a type of meta-event set. Each meta-event itself is also a sub-frame, and its slot values are a trigger and event arguments.

Figure 2. The topic event representation frame



```
Topic Event Frame{
  TInfo{
    Time time,
    Place place,
    Person person,
    ...
  }
  TEFi{                              //i=1,2,3,...
    MEi1{
      Trigger trigger,
      Argument argument1,
      Argument argument2,
      ...
    }
    MEi2{
      Trigger trigger,
      Argument argument1,
      Argument argument2,
      ...
    }
    ...
  }
  ...
}
```

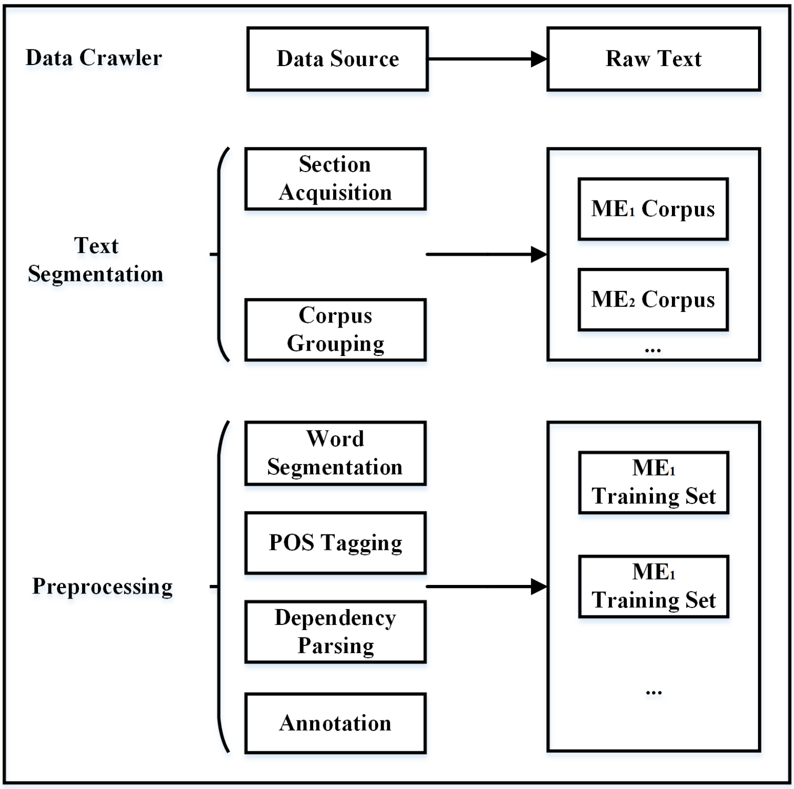
Labeled Corpus Construction

In this section, we manually construct a small amount of labeled corpus, which is part of the preparation for the subsequent meta-event extraction. The process is illustrated in Figure 3. First, we use the web crawler technology to obtain a large amount of unlabeled text from specific websites. Then, we preprocess the obtained text, including section acquisition, corpus grouping, word segmentation, part-of-speech (POS) tagging, and dependency parsing. Finally, we annotate various meta-event corpora after preprocessing and obtain labeled training sets of various meta-events.

Trigger Table Generation

Considering the co-occurrence of triggers and event arguments in meta-events, in the meta-event extraction phase, we locate the description information of meta-events existing in the text set using triggers to filter out irrelevant information, which improves the efficiency of event extraction. We utilize the entropy-based feature selection algorithm (Dash & Liu, 2000) to obtain triggers and regard trigger extraction as a clustering feature extraction problem. Because of the high computational complexity of this algorithm, to reduce the number of words involved in the calculation, we group the description sentences of each type of meta-event into a set, namely, $I = \{i_1, i_2, ..., i_n\}$, where each element of *I* represents a description sentence of a meta-event, and *n* is the number of sentences. As the POS of the trigger is a noun or a verb, we filter out other words of the POS except verbs and nouns in *I*. Let $W = \{w_1, w_2, ..., w_m\}$ denote the set of all words in *I*, where *m* is the number of words. We calculate the entropy value of *I* as *E* through Equation 1, where S_{ij} is the similarity function

Figure 3. The process of labeled corpus construction



between i_i and i_j, and $S_{ij} = \exp(-\alpha * D_{ij})$, D_{ij} is the Euclidean distance between i_i and i_j, α is a positive number, and its value is $-\ln 0.5 / \bar{D}$, and $\bar{D}$ is the average distance among all i_i.

$$E = -\sum_{i=1}^{n}\sum_{j=1}^{n}\left(S_{i,j}\log S_{i,j} + \left(1 - S_{i,j}\right)\log\left(1 - S_{i,j}\right)\right) \tag{1}$$

We remove each word that existed in *W* from *I* in turn, and then calculate the entropy *E* of *I* after removing the word according to Equation 1. Consequently, a set $\{E_1, E_2, \ldots, E_m\}$ is obtained. We select the top 10 words with the largest increase in the value of *E* as the candidate seed trigger. Repeating the above steps, we obtain candidate seed triggers for each type of meta-event. Then, we match the candidate seed triggers with the real triggers in the training set and determine the top three words as seed triggers from the matching results. Finally, we add them to the trigger table according to the meta-event types. In addition, in the meta-event extraction phase, we add the newly identified unknown triggers in labeled results to the trigger table to complete an iterative update of the trigger table. In the next meta-event extraction, we always utilize the updated trigger table to filter irrelevant information.

Semi-Supervised Event Extraction Based on Tri-Training

Tri-training Theory

Tri-training guarantees the difference among the training classifiers by using different sample subsets extracted from the original labeled sample set. The general steps of the tri-training algorithm are as follows. First, the labeled sample set denoted as *L* is sampled to generate three training subsets, which are used to train three classifiers, namely, h_i, h_j, and h_k. In each round, h_j and h_k ($j, k \neq i$) are used to label any sample *x* in unlabeled sample set denoted as *U*. If the labeled results of h_j and h_k are consistent, sample *x* and its category label *y* are combined as a new training sample of h_i, denoted as: $L_i = \{(x, y) : x \in U, y = h_j(x) = h_k(x)\}$. To ensure that the error rate of classifier h_i after iterative training is reduced, Equation 2 should be satisfied when the classifier h_i is trained in each round:

$$e_i \mid L_i \mid < e_i^{'} \mid L_i^{'} \mid \tag{2}$$

where e_i is the error rate of classifier h_i on L_i. Because L_i is selected from *U* through classifiers h_j and h_k, it is difficult to evaluate the error rate. Assuming that *U* and *L* are identically distributed, e_i can be determined using the classification error rate of h_j and h_k on *L*, as shown in Equation 3.

$$e_i = \frac{\left|\left\{x, y\right\} \in L, h_j(x) = h_k(x) \neq y\right|}{\left|\left\{x, y\right\} \in L, h_j(x) = h_k(x)\right|} \tag{3}$$

Considering that when $\mid L_i^{'} \mid$ is too large, Equation 2 no longer holds. To ensure that Equation 2 holds, it is necessary to select at most *u* samples from L_i, where *u* is calculated using Equation 4. The new sample set is shown in Equation 5. To ensure that the $\mid L_i \mid$ obtained by $Subsample(L_i, u)$ is still larger than $\mid L_i^{'} \mid$, $\mid L_i^{'} \mid$ should satisfy Equation 6. Assuming that the initial classification error rate $e_i^{'} = 0.5$, the initial value of $| L_i^{'} |$ can be calculated using Equation 7, and the value of L_i in each round can be calculated using Equations 4 and 5. The last step of each round is to combine L and L_i retrain h_i. The above process is iterated until Equation 2 no longer holds.

$$u = \left\lceil \frac{e_i^{'} \left| L_i^{'} \right|}{e_i} - 1 \right\rceil \tag{4}$$

$$L_i = \begin{cases} Subsample(L_i, u) & \text{Equation 2 no longer holds} \\ L_i & \text{otherwise} \end{cases} \tag{5}$$

$$| L_i^{'} | > \frac{e_i}{e_i^{'} - e_i} \tag{6}$$

$$| L_i^{'} | = \left\lfloor \frac{e_i}{0.5 - e_i} + 1 \right\rfloor \tag{7}$$

Algorithms

We treat the identification of triggers and event arguments from meta-events as a sequence labeling task, introduce tri-training into the sequence labeling process, and propose a semi-supervised event extraction approach based on tri-training and CRF. This approach is divided into two phases: the training phase (Tri-Training-CRFs, as shown in Algorithm 1) and the testing phase (Testing-CoLabeling, as shown in Algorithm 2).

```
**Algorithm 1:** Tri-training-CRFs
**Input:** $L$: Labeled sample set; $U$: Unlabeled sample set; $Train(X)$: The learning algorithm; $BootStrap(X)$: The bootstrap algorithm; $Subsample(X, top\_k)$: The subsampling algorithm; $Error(X, h_1, ...)$: The simultaneous error measuring algorithm.
**Output:** $h_i$ ( $i \in \{1,2,3\}$ )
for $i \in \{1,2,3\}$ do
    $h_i \leftarrow Train(BootStrap(L))$
    $e_i^{'} \leftarrow 0.5$
    $L_i^{'} \leftarrow \varnothing$
end for
repeat
  for $i \in \{1,2,3\}$ do
      $L_i \leftarrow \varnothing$
      $update_i \leftarrow False$
      $e_i \leftarrow Error(L, h_j, h_k)$, ( $j,k \neq i$ )
      if $e_i < e_i^{'}$ then
        $L_i \leftarrow$ Training-CoLabeling($U, h_j, h_k$)
        if $\left| L_i^{'} \right| = 0$ then
            $\left| L_i^{'} \right| \leftarrow \left\lfloor \frac{e_i}{e_i^{'} - e_i} + 1 \right\rfloor$
        end if
```

```
16              if $|L_i'| < |L_i|$ then
17                if $e_i |L_i| < e_i' |L_i'|$ then
18                  $update_i \leftarrow True$
19                else if $|L_i'| > \frac{e_i}{e_i' - e_i}$ then
20                  $u \leftarrow \left\lceil \frac{e_i' L_i'}{e_i} - 1 \right\rceil$
21                  $L_i \leftarrow Subsample(L_i, u)$
22                  $update_i \leftarrow True$
23                end if
24              end if
25            end if
26          end for
27          for $i \in \{1,2,3\}$ do
28            if $update_i = True$ then
29              $h_i \leftarrow Train(L \cup L_i)$
30              $e_i' \leftarrow e_i$
31              $L_i' \leftarrow L_i$
32            end if
33          end for
34        until $update_i = False$ , ( $i \in \{1,2,3\}$ )
```

Step 1: Training the initial sequence labeling model: Use the bootstrap algorithm to obtain three different training subsets from *L*, and train three initial sequence labeling models separately based on CRF (refer to lines 1–5 in Algorithm 1).

Step 2: Obtaining a new training sample set: After the model is trained in the previous round, whether to continue iterative training in the next round must first meet two conditions simultaneously: 1) $e_i < e_i'$ and 2) $|L_i'| < |L_i|$. When the first two conditions are met, the third condition $e_i |L_i| < e_i' |L_i'|$ is also satisfied. If the condition is met, the model will continue to be trained once in this round. While the condition is not met, but $|L_i'| > \frac{e_i}{e_i' - e_i}$ is met, then we set $u \leftarrow \left\lceil \frac{e_i' L_i'}{e_i} - 1 \right\rceil$ and utilize the subsample algorithm to extract the first *u* samples from $| L_i |$ as the new training sample set for this round, where L_i and L_i' represent the new training sample sets obtained in this round and the previous round, respectively, and e_i and e_i' represent the error rate of h_i on L_i in this round and the previous round, respectively (refer to lines 7–26 in Algorithm 1).

Step 3: Co-training the sequence labeling models: When a new training sample set L_i is obtained in each round, *L* and L_i are combined to retrain the sequence labeling model of the meta-events (refer to lines 27–33 in Algorithm 1).

The time complexity of CRF in the training phase is $O(N_f * S * N_l^2)$, and the time complexity in the test phase is $O(S * N_l^2)$, where N_f is the number of features at each position of the sequence to be labeled, S is the length of the sequence to be labeled, and N_l is the number of category labels. In Tri-training-CRFs, each round uses any two CRF models to label *U*, select samples from *U*, add them to the training set of the third CRF model, and then iteratively train the third CRF model once. Therefore, the time complexity of each round of Tri-training-CRFs is divided into labeling time $O(\left|U\right| * S * N_l^2)$ and training time $O(\left|L \cup L_i\right| * N_f * S * N_l^2)$, where $\left|U\right|$ and $\left|L \cup L_i\right|$ are the number of samples in the sets. In the actual scenario, the magnitude of $\left|U\right|$ is very large, and the magnitude of $\left|L \cup L_i\right| * N_f$ is smaller than $\left|U\right|$. We assume that the algorithm stops at *N* iterations, and hence, the overall time complexity of Tri-training-CRFs is $O(N* \mid U \mid *S * N_l^2)$.

In the Tri-training-CRFs algorithm, to ensure that the new training samples selected in each round have a high degree of confidence, we design a reasonable strategy for selecting unlabeled samples, namely, Training_CoLabeling, as shown in Function 1 (refer to line 12 in Algorithm 1).

```
Function 1: Training-CoLabeling
Input: U: Unlabeled sample set; $h_j$, $h_k$ : CRF labeling model, ($j,k \neq i$)
Output: $L_i$
1     $P_{jk} \leftarrow \varnothing$
2     $\theta \leftarrow 0.5$
3     $PSum \leftarrow \varnothing$
4     $NewIns \leftarrow \varnothing$
5     $index \leftarrow 0$
6     $position \leftarrow \varnothing$
7     for each x in U do
8        $Y_j \leftarrow h_j(x)$
9        $Y_k \leftarrow h_k(x)$
10       $Y \leftarrow Y_k \cap Y_j$
11       if $Y \neq \varnothing$ then
12          for $y \in Y$ do
13             $p_{jk} \leftarrow P_j(y \mid x)+P_k(y \mid x)$, ( $j \neq k$ )
14             $P_{jk}.add(\mathrm{y}, p_{jk})$
15          end for
16          $y, psum \leftarrow \max(P_{jk})$
17          if $psum \geq 2 * \theta$ then
18             $PSum.add(index, psum)$
19             $NewIns.add(index, (x, y))$
20             $index \leftarrow index + 1$
21          end if
22       end if
23    end for
24    $position \leftarrow sort(PSum)$
25    $L_i \leftarrow transfer(NewIns, position)$
26    return $L_i$, ( $i \neq j \neq k$ )
```

Step 1: Selecting the consistent label sequence: Use h_j and h_k to label each sample *x* in *U* to obtain Y_j and Y_k, and take the intersection of the two to obtain Y, where, Y_j, Y_k, and Y represent the sets that are made up of multiple label sequences corresponding to *x* (refer to lines 7–10 in Function 1).

Step 2: Selecting the samples whose probability sum meets the threshold conditions: For each label sequence *y* in Y, calculate the sum of $P_j(y \mid x)$ and $P_k(y \mid x)$ as p_{jk}. If the maximum value of p_{jk}

satisfies the given threshold condition, we take the label sequence obtained when p_{jk} takes the maximum value as the label *y* of the sample *x*, and add (*x*, *y*) to the set *NewIns*, where $P_j(y \mid x)$ and $P_k(y \mid x)$, respectively, denote the conditional probability of the sample *x* being labeled by h_j and h_k as the label sequence *y* (refer to lines 11–22 in Function 1).

Step 3: Sorting by probability sum descending order: For each instance in *NewIns*, we sort the instances in descending order according to the value of $psum$ (refer to lines 24–25 in Function 1).

In the meta-event extraction phase, we first utilize the triggers in the trigger table to filter the irrelevant information in the text and preprocess the description sentences of each type of meta-event. Then, we use the three sequence labeling models constructed in the training phase to label the preprocessed text at the same time. The process of determining the label for each unlabeled sequence is presented in Algorithm 2.

```
Algorithm 2: Testing-CoLabeling
Input: T: The testing samples set; $h_i$, $i \in \{1,2,3\}$: CRF labeling model.
Output: $RES$
$P_{123} \leftarrow \varnothing$
$P_{ij} \leftarrow \varnothing$
$P_i \leftarrow \varnothing$
$\theta \leftarrow 0.5$
$RES \leftarrow \varnothing$
for each x in T do
  for $i \in \{1,2,3\}$ do
    $Y_i \leftarrow h_i(x)$
  end for
  $Y \leftarrow Y_1 \cap Y_2 \cap Y_3$
  if $Y \neq \varnothing$ then
    for $y \in Y$ do
      $p_{123} \leftarrow P_1(y \mid x)+P_2(y \mid x)+P_3(y \mid x)$
      $P_{123}.add(y, p_{123})$
    end for
    $y, p_{\max} \leftarrow \max(P_{123})$
    if $p_{\max} \geq 3*\theta$ then
      $RES \leftarrow RES \cup (x,y)$
      continue
    end if
    for $i, j \in \{1,2,3\}, i \neq j$ do
      for $y \in Y$ do
        $p_{ij} \leftarrow P_i(y \mid x)+P_j(y \mid x)$, ($i \neq j$)
        $P_{ij}.add(\mathrm{y}, p_{ij})$
      end for
      $y_{ij}, p_{ij} \leftarrow \max(P_{ij})$
    end for
    $y, p_{\max} \leftarrow \max(p_{12}, p_{13}, p_{23})$
    if $p_{\max} \geq 2*\theta$ then
      $RES \leftarrow RES \cup (x,y)$
```

```
31              continue
32            end if
33            for $i \in \{1,2,3\}$ do
34              for $y \in Y$ do
35                $p_i \leftarrow P_i(y \mid x)$
36                $P_i.add(\mathrm{y}, p_i)$
37              end for
38              $y_i, p_i \leftarrow \max(P_i)$
39            end for
40            $y, p_{\max} \leftarrow \max(p_1, p_2, p_3)$
41            $RES \leftarrow RES \cup (x, y)$
42          else
43            $RES \leftarrow RES \cup (x, nil)$
44          end if
45        end for
46        return $RES$
```

Step 1: Co-labeling: We simultaneously label each sample x in the test sample set T with $h_i, i \in \{1,2,3\}$, and take the intersection of labeled results to obtain Y (refer to Lines 6-10 in Algorithm 2).

Step 2: Determining the label of the unlabeled sample: For each label sequence y in Y, calculate the sum of $P_1(y \mid x)$, $P_2(y \mid x)$, and $P_3(y \mid x)$ as p_{123}. If the maximum value of p_{123} satisfies the threshold condition, the label of sample x is the label sequence y obtained when p_{123} takes the maximum value (refer to lines 12–20 in Algorithm 2). Otherwise, the sum of $P_i(y \mid x)$ and $P_j(y \mid x)$ is calculated as p_{ij} ($i \neq j$). If the maximum value of p_{ij} satisfies the threshold condition, the label of sample x is the label sequence y obtained when p_{ij} takes the maximum value (refer to lines 21–32 in Algorithm 2). If the above threshold conditions are not met, the label of the sample x is the label sequence y obtained when p_i ($i \in \{1,2,3\}$) takes the maximum value (refer to lines 33–41 in Algorithm 2).

Through the above steps, the time complexity of Testing-CoLabeling is $O(\left|T\right| * S * N_l^2)$, where $\left|T\right|$ is the number of samples in the test set, S is the length of the sequence to be labeled, and N_l is the number of category labels.

EXPERIMENTS

In this section, we analyze the experimental results of FISTEE. First, we introduce the dataset. We then elaborate on the experimental settings and measurements. Next, we present the comparison approaches and compare the experimental results in detail. Finally, we summarize the error-extraction process.

Experimental Dataset

In this study, we utilize court verdicts in the legal field as an experimental corpus. A court verdict is a type of long text data that roughly includes five parts: basic information, legal roles, indictment, case information, and judgment. The content of the case information is lengthy, complex, and diverse, and it usually contains factual information on multiple aspects of the case. Thus, we consider the topic events of extracting case information as an example to verify the effectiveness of the proposed algorithm.

As there is currently no publicly available training corpus of court verdicts, we crawl the court verdicts on various motor vehicle accidents, issued by courts of different regions in China, from the "China Judgment Online[1]". We randomly selected 1,800 court verdicts, used regular expressions to match full-text information, and obtained descriptive sentences of case information as experimental data. Then, we divided them into training and test corpora in a ratio of 16:2. After conducting a statistical analysis of all the corpora, we find that the case information contains four aspects: "Topic information," "Liability," "Insurance," and "Disability. Under the guidance of the topic event frame, we regard them as the four facets of the case topic event, and each facet contains a type of meta-event set.

Through the above analysis, we define a unique event representation frame for each type of meta-event, as shown in Table 1. To obtain the labeled corpus and unlabeled corpus of each type of meta-event, we process the case information as follows: First, we label the description sentences in the training corpus manually regarding the meta-event types in Table 1 as labels. Then, we extracted the labeled description sentences and classified them according to the category of labels. Next, we use the "LTP" (Che et al., 2010) to perform word segmentation, POS tagging, and dependency parsing on the training corpus to obtain the feature vector set of the training corpus. Subsequently, we divide the feature vector set into two subsets in a ratio of 1:1, and choose one subset denoted as U. Finally, we manually label another subset to obtain a labeled sample set denoted as L_1, and label U to obtain another labeled sample set denoted as L_2. The label sets of the various meta-events are listed in Table 2. In addition, we adopted an entropy-based feature selection algorithm to generate the trigger table. In the meta-event extraction phase, we utilize triggers to filter irrelevant contents in the case information to improve the efficiency of the meta-event extraction.

Experimental Settings and Measurement

In this study, we use "CRF++" to train the sequence labeling models. We need to specify the feature template and set the value of the hyper-parameter c. Hyper-parameter c is used to balance the degree to which the model fits the training samples. The larger the value of c, the higher the degree of fitting. For c, we set six different values: 1, 1.5, 2, 2.5, 3, and 4. For feature templates, we formulated five feature templates by analyzing the features contained in the meta-event training set. Among them, Template01 contained only word and POS features, and the context window was 3. Template02 and Template03 add dependency features based on Template01, and their context windows are 3 and 5, respectively. Template04 combines Template01 and Template02 to form multiple feature templates, whereas Template05 adds the POS joint dependency feature on the basis of Template04 to form a multiple cross-feature template.

Table 1. Frames of each type of meta-event

Meta-event Type	Event Trigger	< Argument Role, Argument>
主题信息 Topic information	撞击/相撞 Hit/Strike	<时间, > <肇事者, > <肇事车辆, > <相关人物, > <相关车辆, > <原因, > <结果, > <Time, > <Perpetrator, > < Vehicle causing the accident, > <Related person, > <Related vehicle, > <Reason, > <Result, >
主体责任 Liability	认定/划分 Affirm/Divide	<全部责任, > <主要责任, > <同等责任, > <次要责任, > <无责任, > <Full liability, > <Primary liability, > <Equal liability, > <Secondary liability, > <No liability, >
伤残等级 Disability	评定/鉴定 Assess/Identify	<*级伤残*处, > <*Level Disability*, >
投保类型 Insurance	投保/购买 Insure/ Purchase	<交强险, > <商业险, > <不计免赔商业险, > <计免赔商业险, > <Compulsory traffic insurance, > <Commercial insurance, > <Excluding deductible commercial insurance, > <Calculating deductible commercial insurance, >

To obtain the optimal sequence labeling model, we combined different feature templates and parameters to conduct experiments using ten-fold cross-validation. Based on the extraction results of meta-events obtained by various models, we determined the feature templates and parameters. The experimental results of ten-fold cross-validation on the training set of each type of meta-event showed that the labeling results of the models were optimal when the value of *c* was 1.5, and the template was Template05.

In this study, the extraction results of meta-events are evaluated in terms of precision (*P*), recall (*R*), and F_1, as shown in Equations 8, 9, and 10.

$$P = \frac{N_r}{N_r + N_e} \tag{8}$$

$$R = \frac{N_r}{N_{num}} \tag{9}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{10}$$

where N_r and N_e are the number of case text extracted correctly and incorrectly, and N_{num} is the number of case text in the meta-event standard set.

Comparison Approaches

To verify the effectiveness of our approach, we compared the performance of FISTEE with the following comparison approaches.

1. **BasicCRF:** BasicCRF ignores trigger information. Based on the CRF, the sequence labeling model of event arguments is trained on the training set L_1+L_2 and used to label a given case text. Arguments in meta-events are obtained by combining the words labeled "-A."
2. **SEE** (subject event extraction) (Wu et al., 2020): SEE regards the identification of triggers and event arguments in meta-events as a sequence labeling task. On a specific training set, a joint sequence labeling model of triggers and event arguments was trained based on the CRF.

In this study, we trained two SEE models. Specifically, we used the labeled training set L_1 to train a SEE model, namely, SEE(L_1). Similarly, we used the labeled training set L_1+L_2 to train another SEE model, namely, SEE(L_1+L_2).

Table 2. Label set of each type of meta-event

Object	Label set	
	Topic information	Other meta-events
Trigger	B-T/M-T/E-T	
Argument Role	B-Time/M-Time/E-Time B-P0/M-P0/E-P0 B-C0/M-C0/E-C0 B-P1/M-P1/E-P1 B-C1/M-C1/E-C1 B-RES/M-RES/E-RES	B-R/M-R/E-R
Argument	B-A/M-A/E-A	
Other	N	

Experimental Results Analysis

We chose Template05 as the feature template, and the value of *c* was set to 1.5. Experiments were performed on four meta-event test sets of the case information. The experimental results of our algorithm and the comparison approaches are shown in Tables 3–7.

1. Compared with BasicCRF(L_1+L_2), SEE(L_1), SEE(L_1+L_2), and FISTEE(L_1+U) are all joint sequence labeling models of triggers and event arguments. Their extraction results are improved in terms of P, R, and F_1. On the one hand, because a trigger contains rich contextual semantic information, it can promote the performance of the joint sequence labeling model; on the other hand, using triggers can filter irrelevant information in the text to reduce noise interference.
2. Compared with SEE (L_1), FISTEE(L_1+U) has better extraction results for the four meta-event test sets. The overall extraction results are improved by 17.2%, 16.8%, and 17% in P, R, and F_1, respectively, which demonstrates that FISTEE can improve the extraction performance by taking advantage of unlabeled samples. This is because SEE (L_1) only relies on the manually labeled training set to construct the sequence labeling models, whereas labeled training set is limited and may exist as a data-sparse problem, which leads to poor generalization performance of the models. To increase the coverage of labeled samples, FISTEE selects high-confidence pseudo-label samples via tri-training from U and adds them to L_1 when training the models, thereby improving the effectiveness of the models.
3. Compared with SEE (L_1+L_2), FISTEE(L_1+U) has better extraction results on the four meta-event test sets, and its overall extraction results are improved by 14.9%, 14.4%, and 14.7% in P, R, and F_1, respectively, indicating that FISTEE can improve the performance of meta-event extraction while reducing the manually labeled corpus. This is because SEE (L_1+L_2) uses all labeled training sets including L_2 to construct the sequence labeling models, which induces overfitting in the models and decreases the extraction performance, whereas FISTEE gradually selects samples with high confidence and adds them to the training sets until the models converge, thus avoiding overfitting of the models.

The Effect of Feature Template on the Performance of the Sequence Labeling Model

To observe the effect of the feature template on the performance of the sequence labeling model, we set the value of the hyper-parameter *c* to 1.5 and used different feature templates to construct the sequence labeling models based on the FISTEE algorithm. To intuitively compare the performance of the sequence labeling models constructed using different feature templates, we connect the values of the models' labeling results on P, R, and F_1 to form a broken line. Although the broken line itself has no special meaning, the broken line trend clearly presents the difference in the performance of the sequence labeling models. Figure 4(a-d) shows the comparative experimental results of different sequence labeling models on "Topic information," "Liability," "Disability," and "Insurance" meta-event test set, respectively. It can be seen from the broken lines' trend that the sequence labeling

Table 3. Experimental results of compared approaches on Topic information meta-event dataset

Approach	P	R	F_1
BasicCRF(L_1+L_2)	62.4	60.2	61.3
SEE(L_1)	65.1	56.2	60.4
SEE(L_1+L_2)	67.3	59.7	63.3
FISTEE(L_1+U)	70.5	62.1	66.0

Table 4. Experimental results of compared approaches on Liability meta-event dataset

Approach	P	R	F_1
BasicCRF(L_1+L_2)	84.6	82.7	83.6
SEE(L_1)	87.7	82.7	85.1
SEE(L_1+L_2)	87.3	83.2	85.2
FISTEE(L_1+U)	89.0	83.8	86.3

Table 5. Experimental results of compared approaches on Disability meta-event dataset

Approach	P	R	F_1
BasicCRF(L_1+L_2)	85.1	77.8	81.3
SEE(L_1)	97.0	79.3	87.3
SEE(L_1+L_2)	94.3	81.5	87.4
FISTEE(L_1+U)	98.6	87.8	92.9

Table 6. Experimental results of compared approaches on Insurance meta-event dataset

Approach	P	R	F_1
BasicCRF(L_1+L_2)	46.7	46.2	46.4
SEE(L_1)	83.3	64.0	72.4
SEE(L_1+L_2)	80.7	66.1	72.7
FISTEE(L_1+U)	91.8	84.9	88.2

Table 7. Experimental results of compared approaches on total dataset

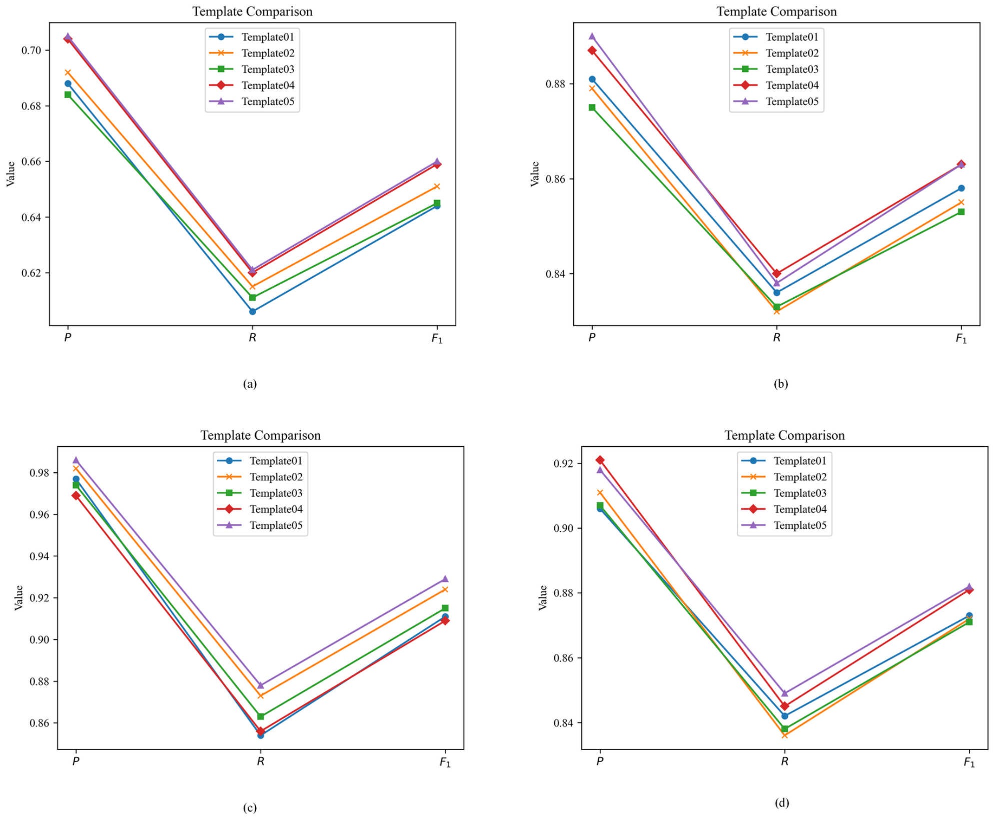
Approach	P	R	F_1
BasicCRF(L_1+L_2)	26.3	26.0	26.1
SEE(L_1)	33.1	32.6	32.9
SEE(L_1+L_2)	35.4	35.0	35.2
FISTEE(L_1+U)	50.3	49.4	49.9

models constructed using Template05 as the feature template have the best extraction results on the four meta-event test sets. Meanwhile, the values of P, R, and F_1 in the extraction results of each model were considerably different. Therefore, formulating an appropriate feature template plays a key role in constructing a sequence annotation model with excellent performance.

Parameter Sensitivity Analysis

In the experiments, we set different values for hyper-parameter c to observe how well the model fits the training samples. To investigate how hyper-parameter c influences the extraction performance, we changed the value of c with Template05 as the feature template. Similar to the previous set of comparative experiments, we conducted experiments on four meta-event test

Figure 4. Extraction performance with different feature templates on four meta-event test sets (c=1.5)



(a)

(b)

(c)

(d)

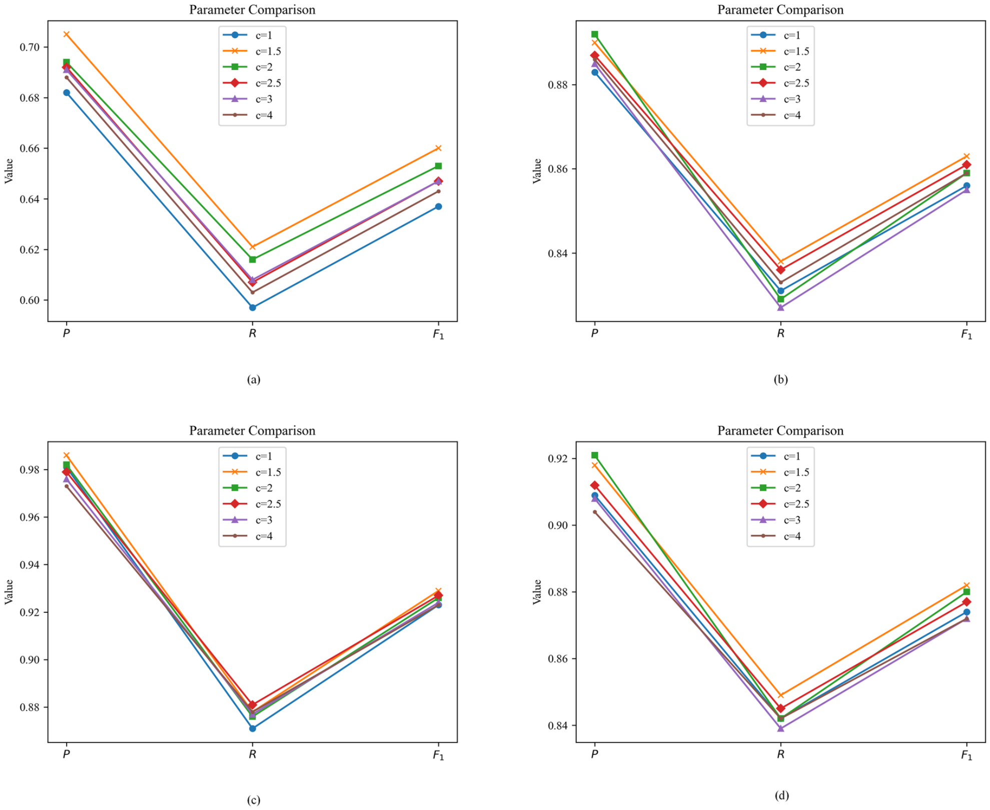
sets of the case information. Figure 5(a-d) depicts the extraction performance on the four meta-event test sets with different values of *c*, from which we can observe that the change in *c* has lesser influence on the performance compared with the change in the feature template. On the four meta-event test sets about "Topic information," "Liability," "Disability," and "Insurance," the models' labeling results show slight differences and the connecting lines of *P*, *R,* and F_1 are very close. When we set *c* to 1.5, the performance of FISTEE on the four meta-event test sets was slightly better than that of the other values. In general, the experimental results prove that FISTEE is stable under various values of *c*.

Error Analysis

Through the statistics of the error-extraction results of meta-events, we find that there are three possible error results: the returned meta-event extraction results are empty, the returned meta-event extraction results are incomplete, and the returned meta-event extraction results are semantically duplicated. The main reasons for these three types of errors are as follows.

1. The description sentences of meta-events are not recognized, resulting in an empty extraction result. This is because some description sentences containing triggers were not added to the test sets. For example, in the "Topic information" meta-event, the test set does not contain the description sentence of the trigger word "collision," which leads to the extraction results of related meta-events being empty.

Figure 5. Extraction performance with different values of c on four meta-event test sets (Template05)



(a)

(b)

(c)

(d)

2. The sequence labeling model does not recognize the description words of event arguments, resulting in incomplete meta-event extraction results. For example, the vehicle type is not recognized in the "Topic information" meta-event and the person's name is not recognized in the "Liability" meta-event, which leads to the failure to extract a complete set of event arguments from the meta-event set.
3. The meta-event extraction results were duplicated semantically. For example, the event argument of the "traffic accident" extracted from the "Insurance" meta-event is the "vehicle causing the accident," but the event argument is also extracted as "the vehicle of a certain license plate." If the "vehicle causing the accident" is correctly extracted in the "Topic information," that is, the "vehicle causing the accident" and "the vehicle of a certain license plate" can be merged to solve the problem of semantic duplication. However, if errors 1 and 2 occur when extracting arguments from the "Topic information" meta-event, the "vehicle causing the accident" cannot be merged with its corresponding license plate number, which will cause the semantic duplication of the event arguments extracted from the meta-events.

THEORETICAL AND PRACTICAL CONTRIBUTIONS

In this section, we discuss the theoretical and practical contributions of our work.

In theory, we design a topic event frame, which organizes the meta-events representing different facets of the text to form a topic event. We improve event extraction performance with a semi-supervised approach via tri-training and automatically select new training samples using the tri-training algorithm in the sequence labeling model. At present, there is little research work on semi-supervised event extraction. (Zhou & Zhong, 2015) used sentence structure and hidden topic embedding in sentences to describe distances, and annotated sentences in an unannotated corpus based on the distance between sentences. Compared to our work, the predefined trigger table in (Zhou & Zhong, 2015) is fixed, which limits the performance of the model because massive unlabeled words are ignored in the trigger table. Moreover, when generating a new annotated sentence, the model (Zhou & Zhong, 2015) only considers the similarity of content and structure while overlooking the rich contextual information between the words, which may lead to a lack of semantic interpretation of the model. (Ferguson et al., 2018) proposed a method for self-training event extraction systems that mention the same event instance in parallel in news text. This method labeled each event cluster for assigning triggers' labels, which added new training samples to the dataset. However, to generate event clusters, this method only uses the same number of entities mentioned in different news in a day to calculate the weight to form an event cluster, which causes some sentences in the cluster to be unrelated to other sentences. Our work fully considers all the meta-events related to the topic in the document to avoid this situation. (Zajec & Mladenić, 2022) used semi-supervised method and integrates cross-language data into the learning process, enhancing the pseudo-annotation supported by probabilistic soft logic. Moreover, to avoid manually annotating data when extracting event argument, (Zajec & Mladenić, 2022) combined Wikipedia and Wikidata to obtain the labeled data. The method takes into account the subject, language, and argument in the annotation corpus, but unfortunately this work focuses only on argument extraction and ignores what we think is the most important and fundamental work of event extraction-event detection.

In practice, our event extraction method is oriented to the legal field and can help with issues such as predicting judgment of cases. Existing event extraction frameworks are mainly applied to the financial and medical fields (Shun et al., 2019), while there is very little work applying event extraction to the legal field. In recent years, the legal field has gradually promoted intelligent legal management, and there are difficulties in automatic conviction and sentencing, large-scale judgment documents, and the analysis of legal issues in legal forums. Event extraction extracts the fine-grained key events of a case and then makes legal judgments based on the extracted event information. Attempts (Shen et al., 2020; Li et al., 2020; Feng et al., 2022) were made to apply event extraction to the legal field using a supervised approach. However, due to the confidentiality of legal documents, it is difficult to annotate a large number of new legal documents. Therefore, we propose a semi-supervised event extraction method that can obtain high-confidence extraction results with only a small number of annotated documents, which is more suitable for the special field of justice.

CONCLUSION

In this study, inspired by frame-based knowledge representation, we design a topic event frame to integrate all the topic-related meta-events scattered in the document to form a topic event. We present a semi-supervised approach to improve the performance of topic event extraction models via tri-training. We propose a reasonable strategy to introduce tri-training into the sequence labeling task, which can select a certain number of samples with high confidence as new training samples from the unlabeled sample set. The selected samples are used together with the human-labeled samples to train the better sequence labeling models. The effectiveness of this approach was verified by conducting experiments. Furthermore, an extracted topic event is represented by different types of structured meta-events, thus presenting the core content of the text from a global perspective.

In future work, we plan to apply our approach to other fields, such as finance and education, sports, to further verify the effectiveness of our approach. Moreover, owing to the introduction of new training samples in iterative learning, our approach inevitably contains noise, which degrades the performance of the sequence labeling model. Therefore, we intend to design a data editing algorithm to identify the samples labeled by errors to optimize the training set and further improve the performance of the sequence labeling model.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China under Grant 2018YFC1603601, the Program for Innovative Research Team in University of the Ministry of Education under Grant IRT_17R32, the National Natural Science Foundation of China under Grant 91746209.

REFERENCES

Ahmad, W. U., Peng, N., & Chang, K. W. (2020). GATE: Graph attention transformer encoder for cross-lingual relation and event extraction. In *Proceedings of the 35th Conference on Artificial Intelligence* (pp. 12462-12470). AAAI.

Ahn, D. (2006). The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events* (pp. 1-8). ACM.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational learning theory* (pp. 92-100). ACM. doi:10.1145/279943.279962

Bu, C., Yu, X., Hong, Y., & Jiang, T. (2021). Low-quality error detection for noisy knowledge graphs. *Journal of Database Management*, *32*(4), 48–64. doi:10.4018/JDM.2021100104

Cao, K., Li, X., Fan, M., & Grishman, R. (2015). Improving event detection with active learning. In *Proceedings of the International Conference Recent Advances in Natural Language Processing* (pp. 72-77). INCOMA.

Che, W., Li, Z., & Liu, T. (2010). LTP: A Chinese language technology platform. In *Proceedings of the 23rd International Conference on Computational Linguistics, Demonstrations Volume* (pp. 13-16). ACM.

Chen, W., Zhang, Y., & Isahara, H. (2006). Chinese chunking with tri-training learning. In *Proceedings of the International Conference on Computer Processing of Oriental Languages* (pp. 466-473).

Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (*Volume 1*: Long Papers)* (pp. 167-176). ACL. doi:10.3115/v1/P15-1017

Chen, Z., & Ji, H. (2009). Language specific issue and feature exploration in Chinese event extraction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers* (pp. 209-212). ACM. doi:10.3115/1620853.1620910

Chou, C. L., Chang, C. H., & Huang, Y. Y. (2016). Boosted web named entity recognition via tri-training. [TALLIP]. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *16*(2), 1–23. doi:10.1145/2963100

Chua, C. E. H., Storey, V. C., & Chiang, R. H. (2012). Knowledge representation: A conceptual modeling approach. [JDM]. *Journal of Database Management*, *23*(1), 1–30. doi:10.4018/jdm.2012010101

Cui, S., Yu, B., Liu, T., Zhang, Z., Wang, X., & Shi, J. (2020). Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (pp. 2329-2339). ACL. doi:10.18653/v1/2020.findings-emnlp.211

Dash, M., & Liu, H. (2000). Feature selection for clustering. In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD 2000: Knowledge Discovery and Data Mining* (pp. 110-121). Springer. doi:10.1007/3-540-45571-X_13

Feng, X., Hu, Z., Liu, C., Ip, W. H., & Chen, H. (2021). Text-image retrieval with salient features. *Journal of Database Management*, *32*(4), 1–13. doi:10.4018/JDM.2021100101

Feng, Y., Li, C., & Ng, V. (2022). Legal judgment prediction via event extraction with constraints. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics* (pp. 648-664). ACL. doi:10.18653/v1/2022.acl-long.48

Ferguson, J., Lockard, C., Weld, D. S., & Hajishirzi, H. (2018). Semi-supervised event extraction with paraphrase clusters. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 359–364). ACL.

Fiori, A., Grand, A., Bruno, G., Brundu, F. G., Schioppa, D., & Bertotti, A. (2014). Information extraction from microarray data: A survey of data mining techniques. *Journal of Database Management*, *25*(1), 29–58. doi:10.4018/jdm.2014010102

Goldman, S. A., & Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning* (pp. 327-334). University of Washington.

He, R., Zhang, Y., Li, T., & Hu, Q. (2014). Improved conditional random fields model with multi-trigger embedding for Chinese event extraction. *World Wide Web (Bussum)*, *17*(5), 1029–1049. doi:10.1007/s11280-013-0231-7

Huang, K. H., Yang, M., & Peng, N. (2020). Biomedical event extraction on graph edge-conditioned attention networks with hierarchical knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (pp. 1277-1285). ACL.

Kodelja, D., Besançon, R., & Ferret, O. (2019). Exploiting a more global context for event detection through bootstrapping. In *Proceedings of the European Conference on Information Retrieval* (pp. 763-770). Springer. doi:10.1007/978-3-030-15712-8_51

Lee, C. S., Chen, Y. J., & Jian, Z. W. (2003). Ontology-based fuzzy event extraction agent for Chinese e-news summarization. *Expert Systems with Applications*, *25*(3), 431–447. doi:10.1016/S0957-4174(03)00062-9

Li, P., Zhou, G., & Zhu, Q. (2016). Minimally supervised Chinese event extraction from multiple views. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *16*(2), 1–16. doi:10.1145/2994600

Li, Q., Ji, H., & Huang, L. (2013). Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (pp. 73-82). ACL.

Li, Q., Zhang, Q., Yao, J., & Zhang, Y. (2020). Event extraction for criminal legal text. In *Proceedings of the 2020 IEEE International Conference on Knowledge Graph* (pp. 573-580). IEEE. doi:10.1109/ICBK50248.2020.00086

Lin, H., Lu, Y., Han, X., & Sun, L. (2018). Nugget proposal networks for Chinese event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 1565-1574). ACL. doi:10.18653/v1/P18-1145

Liu, A., Xu, N., & Liu, H. (2021). Self-attention graph residual convolutional networks for event detection with dependency relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 302-311). ACL. doi:10.18653/v1/2021.findings-emnlp.28

Liu, S., Liu, K., He, S., & Zhao, J. (2016). A probabilistic soft logic based approach to exploiting latent and global information in event classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 2993-2999). AAAI. doi:10.1609/aaai.v30i1.10375

Liu, W., & Nguyen, T. H. (2018). Similar but not the same: Word sense disambiguation improves event detection via neural representation matching. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4822-4828). ACL. doi:10.18653/v1/D18-1517

Lyu, S., & Liu, J. (2021). Convolutional recurrent neural networks for text classification. *Journal of Database Management*, *32*(4), 65–82. doi:10.4018/JDM.2021100105

Minsky, M. (1974). A framework for representing knowledge. In *Proceedings of the Psychology of Computer Vision*. (pp. 211-277). MIT.

Riloff, E. (1993, July). Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 811-816). AAAI.

Sahnoun, S., Elloumi, S., & Ben Yahia, S. (2020). Event detection based on open information extraction and ontology. *Journal of Information and Telecommunication*, *4*(3), 383–403. doi:10.1080/24751839.2020.1763007

Shen, S., Qi, G., Li, Z., Bi, S., & Wang, L. (2020). Hierarchical Chinese legal event extraction via pedal attention mechanism. In *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 100-113). doi:10.18653/v1/2020.coling-main.9

Storey, V. C. (2017). Conceptual modeling meets domain ontology development: A reconciliation. [JDM]. *Journal of Database Management*, *28*(1), 18–30. doi:10.4018/JDM.2017010102

Tsegaye, R., & Qaqqabaa, K. (2020). Event extraction from unstructured Amharic text. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 2103-2109). ACL.

Valenzuela-Escárcega, M. A., Hahn-Powell, G., Surdeanu, M., & Hicks, T. (2015). A domain-independent rule-based framework for event extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing* (pp. 127-132). ACL. doi:10.3115/v1/P15-4022

Wand, Y., & Weber, R. (2017). Thirty years later: Some reflections on ontological analysis in conceptual modeling. [JDM]. *Journal of Database Management*, *28*(1), 1–17. doi:10.4018/JDM.2017010101

Wu, G., Hu, S., Wang, Y., Zhang, Z., & Bao, X. (2020). Subject event extraction from Chinese court verdict case via frame-filling. In *Proceedings of the 11th International Conference on Knowledge Graph (ICKG)* (pp. 12-19). IEEE. doi:10.1109/ICBK50248.2020.00012

Xie, Z., & Tu, Y. (2022). A graph convolutional network with adaptive graph generation and channel selection for event detection. In *Proceedings of the 36th Conference on Artificial Intelligence* (pp. 11522-11529). AAAI. doi:10.1609/aaai.v36i10.21405

Yangarber, R., Grishman, R., Tapanainen, P., & Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th International Conference on Computational Linguisti*cs (pp. 940-946). ACM. doi:10.3115/992730.992782

Zajec, P., & Mladenić, D. (2022). Using semi-supervised learning and Wikipedia to train an event argument extraction system. *Informatica (Vilnius)*, *46*(1), 121–128.

Zhang, C., Hong, S., & Zhang, P. (2016). The research on event extraction of Chinese news based on subject elements. In *Proceedings of the 15th International Conference on Computer and Information Science* (pp. 1-5). IEEE. doi:10.1109/ICIS.2016.7550911

Zhang, Z., Xu, W., & Chen, Q. (2016). Joint event extraction based on skip-window convolutional neural networks. In *Proceedings of the Natural Language Understanding and Intelligent Applications* (pp. 324-334). Springer. doi:10.1007/978-3-319-50496-4_27

Zheng, S., Cao, W., Xu, W., & Bian, J. (2019). Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 337–346). ACL. doi:10.18653/v1/D19-1032

Zhou, D., & Zhong, D. (2015). A semi-supervised learning framework for biomedical event extraction based on hidden topics. *Artificial Intelligence in Medicine*, *64*(1), 51–58. doi:10.1016/j.artmed.2015.03.004 PMID:25863986

Zhou, Y., Chen, Y., Zhao, J., Wu, Y., Xu, J., & Li, J. (2021). What the role is vs. What plays the role: semi-supervised event argument extraction via dual question answering. In *Proceedings of the 35th Conference on Artificial Intelligence* (pp. 14638-14646). AAAI. doi:10.1609/aaai.v35i16.17720

Zhou, Z. H., & Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, *17*(11), 1529–1541. doi:10.1109/TKDE.2005.186

Zhou, Z. H., & Li, M. (2010). Semi-supervised learning by disagreement. *Knowledge and Information Systems*, *24*(3), 415–439. doi:10.1007/s10115-009-0209-z

Zhu, T., Qu, X., Chen, W., Wang, Z., Huai, B., Yuan, N. J., & Zhang, M. (2021). Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence* (pp. 4552–4558). Arxiv.

ENDNOTES

[1] https://supremepeoplescourtmonitor.com/tag/china-judgments-online/

Gongqing Wu is a Professor of Computer Science at Hefei University of Technology (China). He received the B.S. degree in Computer Science from Anhui Normal University (China), the M.S. degree in Computer Science from University of Science and Technology of China (USTC), and the Ph.D. degree from Hefei University of Technology (HFUT). His research interests include data mining and Web intelligence. He has published more than 70 referred research papers. He is the recipient of a Best Paper Award at IEEE International Conference on Tools with Artificial Intelligence (ICTAI) 2011, and a Best Paper Award at IEEE/WIC/ACM International Conference on Web Intelligence (WI) 2012.

Zhuochun Miao received the B.S. degree in Mathematics and Applied Mathematics from Hefei University of Technology, Anhui, China, in 2021. He is currently pursuing the M.S. degree at Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China) and School of Computer Science and Information Engineering at Hefei University of Technology. His research interests are in the field of data mining and natural language processing.

Shengjie Hu received the B.S. degree in Information and Computing Science from Anhui University of Science and Technology, Anhui, China, in 2018. He is currently pursuing the M.S. degree at Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education and School of Computer Science and Information Engineering at the Hefei University of Technology. His research interests are in the field of data mining and pattern recognition.

Yinghuan Wang received the B.S. degree from Anhui University of Finance and Economics, Anhui, China, in 2016, and the M.S. degree in computer science and information engineering from Hefei University of Technology, Anhui, China, in 2019. During the postgraduate, her research interests are in the field of data mining, information extraction and retrieval, and natural language processing. She is currently a software development engineer at the Pudong Development Bank R&D Center, China.

Zan Zhang received the B.S. degree from Northeast Electric Power University, China, in 2009, and the M.S. degree from Hefei University of Technology, China, in 2012. Since 2013, he has been a Ph.D. candidate in School of Computer Science and Information Engineering, the Hefei University of Technology, China. He has been a visiting Ph.D. student in the School of Information Technology and Mathematical Sciences, the University of South Australia, Australia. He is currently a lecturer at Hefei University of Technology. His research interests include data mining, knowledge engineering and artificial intelligence.

Xianyu Bao received the B.S. and Ph.D. degrees in electronic information engineering from Hefei University of Technology (HFUT), Hefei, China. He is currently a research fellow with the High-tech Department of Shenzhen Academy of Inspection and Quarantine. His current research interests include big data technology, monitoring and early warning.