Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Handling polysemous triggers and arguments in event extraction: an adaptive semantics learning strategy with reward–penalty mechanism*

Haili LI[1,3,4], Zhiliang TIAN[‡1], Xiaodong WANG[1], Yunyan ZHOU[2,4],
Shilong PAN[1], Jie ZHOU[1], Qiubo XU[3,4], Dongsheng LI[‡1]

[1]*National Key Laboratory of Parallel and Distributed Computing, College of Computer,*
National University of Defense Technology, Changsha 410073, China
[2]*Unit 63891 of PLA, Luoyang 471003, China*
[3]*Unit 63893 of PLA, Luoyang 471003, China*
[4]*State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System,*
Luoyang 471003, China

E-mail: {lihaili20,tianzhiliang,xdwang,panshilong18,zhoujie,lidongsheng}@nudt.edu.cn; 54zyy@sina.com; xuqb2005@163.com

Received Mar. 21, 2024; Revision accepted May 16, 2024; Crosschecked

Abstract: Event extraction (EE) is a complex natural language processing (NLP) task that aims at identifying and classifying triggers and arguments in raw text. The polysemy of triggers and arguments stands out as one of the key challenges affecting the precise extraction of events. The existing approaches commonly consider the semantics distribution of triggers and arguments to be balanced. However, the sample quantities of the different semantics in the same trigger or argument vary in real-world scenarios, leading to a biased semantic distribution. The bias introduces two challenges: (1) low-frequency semantics are difficult to identify and (2) high-frequency semantics are often mistakenly identified. To tackle these challenges, we propose an adaptive learning method with the reward–penalty mechanism for balancing the semantic distribution in polysemous triggers and arguments. The reward–penalty mechanism balances the semantic distribution by enlarging the gap between the target and nontarget semantics by rewarding correct classifications and penalizing incorrect classifications. Additionally, we propose the sentence-level event situation awareness (SA) mechanism to guide the encoder to accurately learn the knowledge of events mentioned in the sentence, thereby enhancing target event semantics in the distribution of polysemous triggers and arguments. Finally, for various semantics in different tasks, we propose task-specific semantics decoders to precisely identify the boundaries of the predicted triggers and arguments for the semantics. Our experimental results on ACE2005 and its variants, along with ERE benchmarks, demonstrate the superiority of our approach over single-task and multi-task EE baselines.

Key words: Event extraction; Polysemous triggers; Polysemous arguments; Semantic imbalance; Reward–Penalty mechanism

https://doi.org/10.1631/FITEE.2400220 **CLC number:** TP393.1

1 Introduction

Events, serving as carriers of information, possess significant research value due to their ele-

‡ Corresponding authors

* Project supported by the National Natural Science Foundation of China (No. 62306330 and No. 62106275), Young Elite Scientist Sponsorship Program by CAST (No. YESS20230367), Natural Science Foundation of Hunan Province (No. 2022JJ40558 and No. WDZC20235250103)

ORCID: Haili LI, http://orcid.org/ 0000-0002-7739-5317; Zhiliang TIAN, https://orcid.org/0000-0002-8906-5198

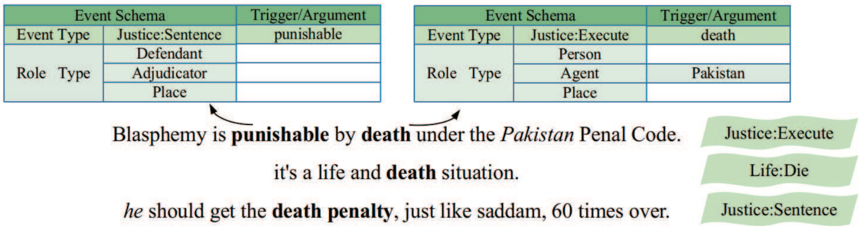


Fig. 1 The EE results of the sentence âĂIJBlasphemy is punishable by death under the Pakistan Penal Code.âĂİ and examples of all semantics for the triggers âĂIJdeathâĂİ and âĂIJpunishable.âĂİ Words in bold are triggers, while those that are italicized are arguments. Event âĂIJJustice:SentenceâĂİ is triggered by the trigger âĂIJpunishableâĂİ with no arguments playing any roles. Trigger âĂIJdeathâĂİ triggers the âĂIJJustice:ExecuteâĂİ event, where âĂIJPakistanâĂİ plays the âĂIJAgentâĂİ role in the event schema of âĂIJJustice:Execute.âĂİ EE, event extraction.

vated information content and rich semantic details. The accelerated evolution of the Internet and the emergence of numerous Internet applications have brought a large number of unstructured and fragmented text resources. How to quickly and accurately obtain structured target event information from these resources has always been a key and challenging problem for scholars engaged in the field of event extraction (EE). EE (Ahn, 2006; Chen et al., 2015; Lu et al., 2023) is the task of identifying and classifying triggers and arguments from unstructured text based on the predefined event schema, as shown in Fig. 1. EE enables users to obtain information in a timely and intuitive manner on who (doer), when (time), where (place), how (artifact), whom (recipient), and what (event) occurred. The extracted event can be widely used in downstream applications, such as event graph construction (Shu et al., 2021; Xu et al., 2022b), recommendation systems (Cui et al., 2023; Xia et al., 2023), decision aids (Anelli et al., 2022; You et al., 2023b), etc.

Many efforts have been devoted to EE. Earlier EE methodologies mainly relied on manually crafted multi-granularity features (Ji and Grishman, 2008; McClosky et al., 2011; Hong et al., 2011), which were labor-intensive. The emergence of deep learning techniques (Chen et al., 2015; Nguyen et al., 2016; Sha et al., 2018), capable of automatically learning features of tasks from ample annotated data, has overcome the limitations of manual feature design. Recently, pre-trained language models (PLMs) (Yang et al., 2019; Lin et al., 2020; Lu et al., 2021; Liu et al., 2022) with rich general language representations, such as BERT and RoBERTa, have become the backbone of EE systems, reducing the need for extensive annotated data. To address the challenges introduced by low-resource scenarios, including zero-shot and few-shot, prompts (Hsu et al., 2022; Wang et al., 2023b; Yao et al., 2023; Zhang et al., 2023b) with task-specific knowledge aid PLMs in comprehending the content and format of tasks, which require significant training. Large language models (LLMs) (Pang et al., 2023; Li et al., 2023; Ettinger et al., 2023) with exceptional text understanding and generation abilities require no training and are extensively applied across various natural language processing (NLP) tasks.

The polysemy of triggers and arguments poses significant challenges for EE (Feng et al., 2018; Ding et al., 2019). We take into consideration a trigger or argument of polysemy when it is associated with two or more semantics. We take polysemous triggers as an example for analyzing the challenges, as shown in Fig. 1. The polysemous trigger âĂIJdeathâĂİ triggers three distinct events: âĂIJLife:Die,âĂİ âĂIJJustice:ExecuteâĂİ and âĂIJJustice:Sentence,âĂİ whereas âĂIJpunishableâĂİ triggers only âĂIJJustice:Sentence.âĂİ Event types' semantics are finite, discrete, and predefined, which define the output space for EE tasks. The output space of polysemous triggers and arguments is a complex space composed of multiple semantic subspaces. Mapping âĂIJdeathâĂİ to âĂIJJustice:ExecuteâĂİ among its three relevant semantics is more challenging than mapping âĂIJpunishableâĂİ to âĂIJJustice:Sentence.âĂİ

Polysemy increases the complexity of event detection (ED) tasks, making it difficult for models to determine the exact meaning represented by polysemous triggers and arguments. Specifically, the challenges introduced by polysemy to ED tasks manifest in two main aspects: (1) Semantic ambiguity. Polysemy makes it difficult to distinguish the semantics of triggers and arguments, mainly manifested in their possible multiple different semantics. It requires complex semantic disambiguation for ED models to map triggers and arguments to predefined types. (2) Context dependency. Polysemy affects the semantics of triggers and arguments by the context, allowing the same word to represent different semantics in different sentences. Consequently, the semantics of triggers and arguments may become ambiguous across different contexts, increasing the complexity of understanding and modeling their semantics for models.

To tackle the aforementioned challenges, the existing EE studies have employed various methods to enhance the semantics of polysemous triggers and arguments, including context knowledge (Chen et al., 2015; Lu et al., 2023), knowledge enhancement (Du and Ji, 2022), multi-task learning (Ping et al., 2023), and prompt-based approaches (Yao et al., 2023; Zhang et al., 2023b), which treat the multiple semantics in polysemous triggers and arguments as balanced semantics. However, the semantic distribution is imbalanced. This imbalance poses some challenges for the semantic modeling of polysemous triggers and arguments, as well as the identification of their boundaries, mainly in the following three aspects. (1) Biased semantic distribution. By analyzing the samples of polysemous triggers and arguments in the dataset, we find significant differences in the distribution of sample numbers for different semantics. We observe in Fig. 2a that the number of samples with the semantic âĂIJLife:DieâĂİ is much higher than the number with the semantics âĂIJJustice:ExecuteâĂİ and âĂIJJustice:Sentence.âĂİ This imbalanced semantic distribution results in the model paying more attention to the high-frequency semantics in polysemous triggers and arguments during the learning process, while neglecting the acquisition of low-frequency semantics. Consequently, this further leads to the omission of low-frequency semantics and the erroneous identification of high-frequency semantics. (2) Misidentification of relevant and irrelevant semantics. The probability of relevant and irrelevant semantics is greater than the target semantic, denoted as a false positive (FP) that the nontarget semantics is identified, as illustrated in Fig. 2b. (3) Difficulty in boundary identification. For multi-token triggers and arguments, the polysemy of subtokens presents a substantial challenge in accurately identifying the boundaries of triggers and arguments.

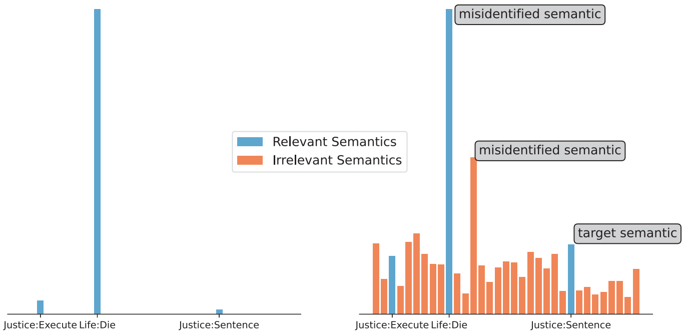
To tackle these challenges, we introduce an adaptive semantics learning method for handling the imbalanced semantics in polysemous triggers and arguments using the reward–penalty mechanism, denoted as RPEE. Firstly, we leverage the reward–penalty mechanism to balance the biased distribution of semantics by weakening the high-frequency semantics and amplifying the low-frequency semantics, and improve the classification accuracy by enlarging the gap between target semantics and nontarget semantics. This approach dynamically adapts the different semantics of polysemous triggers and arguments, based on the semantic probability distribution and the model's classification outcomes, which differs from the traditional methods that adjust category weights (Yang et al., 2019; Nam et al., 2022). Additionally, we utilize the sentence's event semantics to enhance the semantics of triggers and arguments, intending to reduce FPs for irrelevant semantics and nontarget relevant semantics. To ensure the accuracy of the sentence's event semantics for avoiding the error propagation, the proposed sentence-level event situation awareness (SA) mechanism utilizes a sentence event classification task for precisely modeling the sentence's event semantics. Finally, we develop task-specific decoders to identify all candidate spans for triggers and arguments in the sentence, classifying their types for different semantics using task-specific thresholds. Experimental results demonstrate that our method effectively mitigates the imbalanced semantic distribution of polysemous triggers and arguments.

Our contributions are fourfold:

- To solve the imbalanced semantic distribution of polysemous triggers and arguments, we introduce a semantic-adjustment method to minimize the gap between relevant semantics by utilizing the reward–penalty mechanism.

- We devise a reward–penalty mechanism to mitigate the biased distribution of semantics by dynamically adjusting different semantics in polysemous triggers and arguments.

- The proposed sentence event situation awareness (SESA) mechanism provides correct event constraints for triggers and arguments in the sentence. Additionally, the task-specific decoder



(a) Original semantic distribution of "death" (b) Semantic distribution of "death" after balancing

Fig. 2 Original semantic distribution vs. semantic distribution after balancing. The figure (a) shows the original imbalanced semantic distribution that the number of samples with the semantic "Life:Die" is significantly higher than the number of samples with the semantics "Justice:Execute" and "Justice:Sentence". The figure (b) shows the semantic distribution after balancing, in which the probabilities of both relevant and irrelevant semantics being detected are higher than that of the target semantic, leading to a false positive (FP) where the nontarget semantics are identified and a false negative (FN) where the target semantic is misidentified.

accurately identifies the boundaries of triggers and arguments comprised of an uncertain number of tokens.

- Extensive experiments demonstrate that RPEE outperforms the state-of-the-art EE methods, demonstrating strong robustness, generalization ability, and superior performance in handling polysemous triggers and arguments, even in complex scenarios where triggers and arguments comprise multiple tokens.

2 Related work

This section reviews EE approaches and summarizes models that accurately identify the boundaries of triggers and arguments.

2.1 EE

EE is a fundamental, crucial, and complex task in information extraction (IE) and NLP, focused on identifying triggers, event participants, and event types in text. Many efforts have been made from various perspectives to enhance EE performance. Researchers have utilized various neural networks like CNN (Chen et al., 2015; Zeng et al., 2016), recurrent neural network (RNN) (Nguyen et al., 2016; Sha et al., 2018), LSTM (Feng et al., 2018; Lou et al., 2021), and GNN (Liu et al., 2018; Cui et al., 2020) to capture event features. From the perspective of leveraging knowledge of tasks and datasets, Du and Cardie (2020), Liu et al. (2021), Yang et al. (2021), Du and Ji (2022), Lu et al. (2023), Wang et al. (2022) formalize EE as machine reading comprehension (MRC) or question answering (QA) tasks. Some researchers employ well-designed prompts (Hsu et al., 2022; Ma et al., 2023; Ping et al., 2023; Yao et al., 2023; Zhang et al., 2023b) to guide language models in extracting events. Some others (Ettinger et al., 2023; Hsu et al., 2023; Yang et al., 2023b) utilize NLP tools to make use of syntactic, syntax, and semantic knowledge in the data. From the perspective of leveraging external resources, Liu et al. (2022), Wang et al. (2023a), and Yao et al. (2023) tackle the challenge of data scarcity by generating instances. The methods mentioned above have made significant progress in EE. However, they tend to overlook the uneven distribution aspect of semantics in triggers and arguments, leading to numerous FPs that impact the overall performance of EE.

2.2 Boundary identification for EE

Identifying the boundaries of triggers and arguments is crucial for accurately extracting events, especially for those that consist of multiple tokens. In this section, we review relevant literature on EE, specifically focusing on sequence labeling and span-based approaches.

2.2.1 Sequence labeling EE

The sequence labeling EE models formalize EE as sequence labeling, aiming to model the semantic distribution of triggers and arguments. Various labeling schemes exist, including IO, BIO, BMES, and BIESO, where BIOMES stands for Beginning, Inside, Outside, Middle, End, and Single, respectively. Different methods use different labeling schemes. Liu et al. (2023a) and Guzman-Nateras et al. (2023) utilize the IO scheme for labeling. To better leverage the potential transferred knowledge between labels, (Nguyen et al., 2016), Sha et al. (2018), Lin et al. (2020), Cong et al. (2021), Xu et al. (2023b), Liu et al. (2022) and Wang et al. (2023) utilize RNNs or conditional random fields(CRF) and the BIO labeling scheme to model the boundaries of triggers and arguments. However, the sequence labeling method fails to handle nested triggers and arguments.

2.2.2 Span-based EE

In contrast to the methods rooted in sequence labeling, span-based modeling approaches aim to tackle intricate event structures, such as nested triggers and arguments. These approaches transform the EE task into a text span classification task, aiming to identify target triggers or arguments from all candidate spans and to classify each span's type. Depending on the modeling, span-based methods consist mainly of boundary location modeling and span representation modeling. Existing works (Yang et al., 2019; Du and Cardie, 2020; Yang et al., 2021; Xu et al., 2022a; He et al., 2023) utilize two task-specific classifiers to model the head and tail tokens of the span, respectively. The works (Dozat and Manning, 2017; You et al., 2022; Ping et al., 2023; You et al., 2023a) utilize the biaffine attention mechanism to jointly model the head and tail tokens of the span. Wadden et al. (2019) and Yang et al. (2023b) enumerate all spans, to model the joint representation of spans for multi-token triggers and arguments.

Sequence labeling EE methods model the semantic distribution of triggers and arguments but fail to handle the nested or overlapping ones. However, span-based approaches tackle the issue but struggle with the imbalanced semantic distribution of polysemous triggers and arguments. To address this, we formalize the EE task as a token-classification problem and propose a reward–penalty mechanism to dynamically adjust the imbalanced semantic distribution of polysemous triggers and arguments, thereby mitigating their bias. Additionally, we design task-specific decoders to model the boundaries of triggers and arguments, respectively.

3 Preliminaries

3.1 Task formulation

Following the definition by Ahn (2006), the process of EE consists of ED (Liu et al., 2023b; Wang et al., 2023b) and event argument extraction (EAE) (He et al., 2023; Yang et al., 2023a), aiming at extracting triggers and arguments from the given sentence, as well as mapping them to the predefined types, respectively. We formalize ED and EAE as multi-label classification tasks to address the polysemy of triggers and arguments. The type set of ED and EAE is denoted as $E = \{e_1, ..., e_M\} \cup \{e_0 = \text{“}NULL\text{”}\}$ and $R = \{r_1, ..., r_m\} \cup \{r_0 = \text{“}NULL\text{”}\}$, respectively, where âĂIJNULLâĂİ indicates that the token is neither triggers nor arguments.

For a given sentence $X = \{x_1, ..., x_n\}$, where n is the length of tokens, ED identifies all candidate triggers for each semantic, and presents results in the format of $\bigcup_{i=1}^{M} \left\{[e_i : \bigcup_{j=1}^{t} [(s_{ij}, e_{ij})]]\right\}$, where s_{ij} and e_{ij} represent the head and tail positions of the j-th candidate trigger for event type e_i, $e_i \in E$, t is the number of triggers for event e_i. According to the predefined event schema, the argument role set of e_i is $r^i = \left\{r_1^i, ..., r_a^i\right\}$, where $r_j^i \in R$. EAE recognizes all candidate arguments playing the role r_i, and the result is presented as $\left\{e_i : \bigcup_{j=1}^{a} \left\{r_j : \bigcup_{k=1}^{b} [(s_j^k, e_j^k)]\right\}\right\}$, where s_j^k and e_j^k represent the head and tail positions of the k-th candidate argument for role type r_j, respectively.

Definition 1 In the training set, suppose token x is labeled with a set of labels, denoted as $e_r = \{e_{x1}, ..., e_{xg}\}$, where $e_{xi} \in E$ and $g \leq M$. Here, e_r represents the relevant semantics for token x, while the remaining semantics $e_u = E - e_r$ constitute the irrelevant semantics of token x.

3.2 SA

SA (Endsley, 1988, 2001) perceives environment factors or events within the complex and dynamically changing information environment, comprehends their significance, and predicts their future states. SA is an intrinsic representation of the constantly changing external environment, which forms the fundamental basis for subsequent decision-making and performance.

Let $Ev = \{en_1, ..., en_n\}$ be the set of information in the environment, where en_i is the i-th kind of information, and Sa is the function representing the SA model. Consequently, $Sa(Ev) = \{s_1, ..., s_s\}$ signifies the comprehensive state of the environment Ev, with s_i representing the i-th state component.

With the help of situational information, individuals or systems can better comprehend and adapt to complex environments. SA is widely used in various fields such as cybersecurity (Onwubiko, 2020; Matey et al., 2022), power systems (Dwivedi et al., 2023), disease prevention (Shashikumar et al., 2021), and traffic security (Zhang et al., 2023a), and is also applied in specific tasks, such as emotion recognition (Akgun et al., 2023; Palash and Bhargava, 2023). To our knowledge, this paper is the first to introduce SA into EE, enhancing the understanding and adaptation to complex event environments.

3.3 Binary cross-entropy (BCE) loss

The BCE loss (Zheng et al., 2022; Xu et al., 2023a), commonly known as the sigmoid loss, employs the sigmoid function to compute probabilities and is commonly utilized by binary classification tasks and multi-label classification tasks. The sigmoid function independently calculates probabilities for each category, thereby preventing interference between different semantics. The formulation of BCE loss is:

$$\begin{aligned}\mathcal{L}(\boldsymbol{Y},\hat{\boldsymbol{Y}}) = -\frac{1}{c}\sum_{i=1}^{c}[y_i log(\delta(r_i))\\ + (1-y_i)log(1-\delta(r_i))])\end{aligned} \tag{1}$$

where $C = \{1, 2, .., c\}$ is the target set, $\boldsymbol{Y} = [y_1, ..., y_c]$, and $\hat{\boldsymbol{Y}} = [\delta(r_1), ..., \delta(r_c)]$ are the one-hot vector of the ground truth label and the predicted label vectors for the input x, respectively, $y_i \in \{0, 1\}$, r_i is the logits value of x on class i, $r_i \in [0, 1]$, and δ is the sigmoid function. Suppose that the ground-truth label for x is class r, the other classes are uniformly represented as $u = C \setminus \{r\}$, then the gradient of class r and $u_i \in u$ are:

$$\frac{\partial\mathcal{L}(\boldsymbol{Y},\hat{\boldsymbol{Y}})}{\partial r_r} = \frac{\delta(r_r) - 1}{c}, \frac{\partial\mathcal{L}(\boldsymbol{Y},\hat{\boldsymbol{Y}})}{\partial r_{u_i}} = \frac{\delta(r_{u_i})}{c} \tag{2}$$

4 Our approach

This paper presents a method to mitigate the biased semantic distribution of polysemous triggers and arguments using the reward–penalty mechanism. The overall framework, depicted in Fig. 3, consists of four main modules:

•*Reward–Penalty mechanism* dynamically adjusts the learning method of various semantics in polysemous triggers and arguments. It rewards well-learned semantics while penalizing erroneous ones with the semantic probability distribution and the model's classification outcomes.

•*The SESA mechanism* generates an accurate and comprehensive representation for all events mentioned in sentences.

•*Semantic-enhanced encoder* represents tokens with vectors and enhances semantics in tokens with all events mentioned in the sentence.

•*Task decoder* identifies all potential trigger and argument candidates in the sentence and classifies their types.

Our training procedure comprises three phases: pretraining the SESA module (Section 4.2), then training ED and EAE with their respective semantic-enhanced encoder (Section 4.3), and using the task decoder (Section 4.4). ED provides EAE with target role sets based on the predefined event schema. The semantic-enhanced encoder furnishes the task decoder with token representations augmented by the sentence event semantics. SESA ensures the accuracy of the sentence event semantics provided to the semantic-enhanced encoder. Finally, the task decoder identifies the boundaries of triggers and arguments based on the representations of tokens and the reward–penalty mechanism (Section 4.1) and classifies their types.

4.1 Reward–penalty mechanism

The reward–penalty mechanism dynamically adjusts different semantics of polysemous triggers and arguments by rewarding semantics that are correctly classified and penalizing erroneous ones. Subsequently, we provide a detailed analysis of the causes (Section 4.1.1), the desired effects (Section 4.1.2), and the implementation of the reward–penalty mechanism, covering both multi-factor (Section 4.1.3) and single-factor (Section 4.1.4) implementation methods.

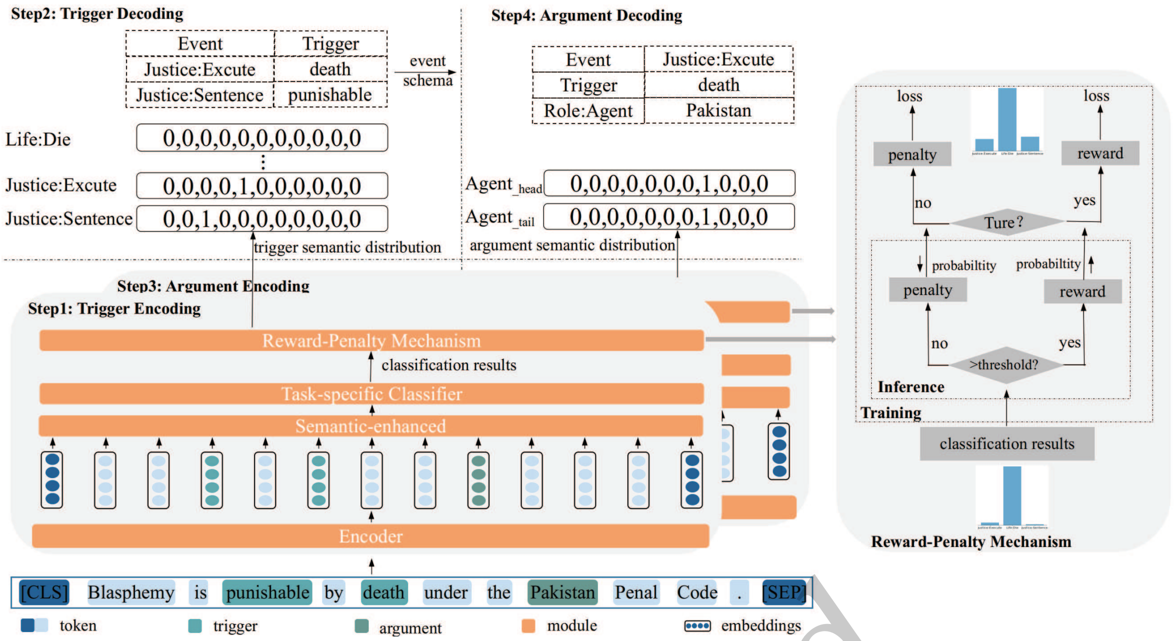


Fig. 3 The overview of our joint EE model. The encoder converts tokens into high-dimensional vectors. Before this, the sentence-level event SA mechanism fine-tunes the encoder to guarantee the precise representation of the sentence's events. The encoder then uses this representation S to enhance token semantics during encoding. Subsequently, our model calculates the probability distribution $P(x_i)$ and employs the reward–penalty mechanism to amplify the correct semantics and diminish the incorrect ones, widening the gap between them. Finally, using $P(x_i)$, the trigger decoder and argument decoder use task-specific thresholds to identify and classify all candidates. EE, event extraction.

4.1.1 Motivational analysis of the reward–penalty mechanism

We analyze the misclassifications introduced by the imbalanced sample quantities from the following perspectives.

FP of irrelevant semantics. Let n_p and n_h denote the number of samples in the dataset annotated with label p and h, respectively, where $n_h > n_p$, the weight updating process is:

$$w_{new} = w_{old} - \sum_{i=1}^{n} \frac{\delta(r_i) - 1}{c} \tag{3}$$

When $w_{old-p} = w_{old-h}$ and $\delta r_p = \delta r_h$, then $w_{new-h} > w_{new-p}$. For a test sample of class p, the trained model tends to categorize it as h, resulting in an FP.

FN of low-frequency semantics. Suppose that the two classes rh and rl of token x_i have k_1 and k_2 samples, respectively, with $k_1 > k_2$, $\{rh, rl\} \in C$. The accumulated gradients of rh and rl are:

$$\sum_{i=1}^{k_1} \frac{\partial \mathcal{L}(\boldsymbol{Y}, \hat{\boldsymbol{Y}})}{\partial r} = \sum_{i=1}^{k_1} \frac{\delta(r_i) - 1}{c} \tag{4}$$

When $\delta(r_{rh}) = \delta(r_{rl})$, the accumulated gradient value of class rh is greater than class rl, resulting in a biased semantic distribution for x and class rl being overwhelmed by class rh, as well as an FN for class rl.

4.1.2 Purposes of reward–penalty

The reward–penalty mechanism aims to mitigate these challenges by boosting the gradient gap between the relevant and irrelevant semantics in Eq. (3), and simultaneously diminishing the accumulated gradient gaps between semantics within the set of relevant semantics in Eq. (4). The gradient adjustment for semantics is detailed as follows:

$$\frac{\partial \mathcal{L}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}_{\boldsymbol{\mathcal{F}}})}{\partial r} = \mathcal{K}_{r,u} \frac{\partial \mathcal{L}(\boldsymbol{Y}, \hat{\boldsymbol{Y}})}{\partial r} \tag{5}$$

where $\mathcal{F}(\cdot) \in [0, 1]$ is the activation function that realizes the reward–penalty mechanism, $\mathcal{K}_{r,u} = \mathcal{P}_{r,u} \cdot \mathcal{R}_{r,u}$ is the reward–penalty factor consisting of the reward factor $\mathcal{P}_{r,u}$ and the penalty factor $\mathcal{R}_{r,u}$.

4.1.3 Multi-factor reward–penalty mechanism

The multi-factor reward–penalty mechanism accurately adjusts various semantics based on classification outcomes and the probability distribution of tokens. It alleviates the bias in the semantic distribution by amplifying the probability of low-frequency semantics while diminishing that of high-frequency semantics and irrelevant semantics. We elaborate on the implementation process of the reward and penalty mechanisms, respectively, delineating the specific problems they target.

Reward mechanism utilizes a reward factor $\mathcal{R}_{r,u}$ and the sample distribution to adjust the weight of the correctly classified semantics' gradients, including the true positive (TP) high-frequency semantics rh and the true negative (TN) irrelevant semantics u. We adjust the weight of semantics' gradients as follows:

$$\mathcal{R}_{r,u} = \begin{cases} (r_{rh})^{\mathcal{R}_r}; r_{rh} \geq t, a\ TP\ happens \\ (r_u)^{\mathcal{R}_u}; r_u \leq t, a\ TN\ happens \\ 1; r_{rh} < t\ or\ r_u > t \end{cases} \tag{6}$$

where $\mathcal{R}_r > 0$ and $\mathcal{R}_u > 0$ are hyper-parameters. When a sample of rh is correctly classified, the value of the reward factor $\mathcal{R}_{r,u}$ decreases, resulting in a gradient enlargement for rh, which narrows the gap in Eq. (4) between the accumulated gradients values of high-frequency and low-frequency semantics. Additionally, the weight of u diminishes to widen the gap between relevant and irrelevant semantics, making them easier to be distinguished.

Penalty mechanism employs a penalty factor $\mathcal{P}_{r,u}$ and the model's classification outcomes to handle misclassified semantics, including unidentified target semantics and misidentified irrelevant semantics. The adjustment process is as follows:

$$\mathcal{P}_{r,u} = \begin{cases} (\frac{t}{\mathcal{F}(r_{rl})})^{\mathcal{P}_r}; \mathcal{F}(r_{rl}) < t, a\ FN\ happens \\ (\frac{\mathcal{F}(r_u)}{t})^{\mathcal{P}_u}; \mathcal{F}(r_u) > t, a\ FP\ happens \\ 1; \mathcal{F}(r_l) \geq t\ or\ \mathcal{F}(r_u) \leq t \end{cases} \tag{7}$$

where $\mathcal{P}_r > 0$ and $\mathcal{P}_u > 0$ are hyper-parameters adjusting the punishment and t is the classification threshold. When a small sample of token x is labeled rl and an FN happens, the penalty mechanism decreases the gradient of rl, as illustrated in Eq. (5), thereby amplifying its weight, as described in Eq. (3). In the case of an FP, according to Eq. (7), $P_{r,u}$ increases, leading to an increase in the gradient of irrelevant semantics and a consequent decrease of its weight.

4.1.4 Single-factor reward–penalty mechanism

The numerous hyper-parameters of each factor in the multi-factor reward–penalty mechanism pose significant challenges to the accurately modeling of the semantic distribution of polysemous triggers and arguments. Therefore, we propose a single-factor reward–penalty mechanism with one hyper-parameter to improve feasibility and usability by sacrificing some accuracy. We meticulously design the following reward–penalty function to implement the reward–penalty mechanism.

$$pr(x, \mathcal{K}_{r,u}) = \frac{1}{1 + e^{(-x) \times \mathcal{K}_{r,u}}} \tag{8}$$

where x is the logits score of the input on a specified class, $\mathcal{K}_{r,u} > 1$ is an integer.

Table 1 Changes in loss value.

		Prediction	
		Positive	Negative
Gold	Positive	-	+
	Negative	+	-

âĂIJ+âĂİ and âĂIJ-âĂİ denote the increase and decrease in the loss value using the function $pr(x, \mathcal{K}_{r,u})$ compared to using the function $\delta(x)$, respectively.

Subsequently, we will elaborate on how the function $pr(x, \mathcal{K}_{r,u})$ implements the reward–penalty mechanism from the model training and inference perspective.

Training process. The reward–penalty function $pr(x, \mathcal{K}_{r,u})$ directs the model training by modifying the loss based on the model's classification outcomes. In the case of incorrect classifications, the reward–penalty function $pr(x, \mathcal{K}_{r,u})$ boosts the loss so as to encourage the model to learn more about the gold label of the input. Conversely, correct classifications result in $pr(x, \mathcal{K}_{r,u})$ decreasing the loss, which helps avoid over-fitting and reduces the gap between high-frequency and low-frequency semantics. Table 1 illustrates the impact on the loss value.

Specifically, when token x labeled with class i is correctly classified and $r_i > 0$, $pr(r_i, \mathcal{K}_{r,u}) > \delta(ri)$, the heightened probability serves as a reward for correct categorization. From Eq. (5), we observe that $loss_{pr} < loss_s$, where $loss_{pr}$ and $loss_s$ employ $pr(r_i, \mathcal{K}r, u)$ and $\delta(r_i)$ as the activation functions, respectively. The decrease in loss implies a reduced need for parameter tuning, further preventing the weight of class i from becoming excessively large. This illustrates that $pr(r_i, \mathcal{K}_{r,u})$ effectively balances the semantic distribution.

In case token x labeled with class r is unidentified, $r_r < t$ or $pr(r_r, \mathcal{K}_{r,u}) < pr(r_u, \mathcal{K}_{r,u})$, an FN happens. With $pr(r_r, \mathcal{K}_{r,u}) < \delta(r_r) < t$, $loss_{pr} > loss_s$, the increased loss can be seen as a penalty and leads the model to learn more about class r.

When token x do not have samples labeled with class u but $pr(r_r,\mathcal{K}_{r,u}) < pr(r_u, \mathcal{K}_{r,u})$, it indicates that u is misclassified. $loss_{pr} > loss_s$, the increased loss guides the model to reduce the weight of u to avoid the FP.

Inference process. The reward–penalty function $pr(x, \mathcal{K}_{r,u})$ simplifies the threshold setting by enlarging the gap between the target and the nontarget semantics, improving the accuracy of identifying the target semantic.

Typically, t is set to 0.5 and varies across tasks. Assuming the token x_i labeled with p is correctly recognized, then $pr(r_p, \mathcal{K}_{r,u}) \geq t$. At the same time, it can be observed from $pr(r_p, \mathcal{K}_{r,u}) > \delta(r_p)$ that $pr(x, \mathcal{K}_{r,u})$ rewards the target semantic p of x_i by enlarging its probability. Conversely, if the nontarget semantic u of x_i is misclassified, $pr(r_u, \mathcal{K}_{r,u}) < t$, and $pr(r_u, \mathcal{K}_{r,u}) < \delta(r_u)$. The reward–penalty function $pr(x, \mathcal{K}_{r,u})$ punishes the nontarget semantic u by diminishing its probability. Simultaneously, by comparing changes in the probability of various semantics utilizing different activation functions, we know $pr(r_p, \mathcal{K}_{r,u}) > \delta(r_p) > \delta(r_u) > pr(r_u, \mathcal{K}_{r,u})$, and $pr(r_p, \mathcal{K}_{r,u}) - pr(r_u, \mathcal{K}_{r,u}) > \delta(r_p) - \delta(r_u)$. A wider boundary implies an easier setting for the threshold and a more accurate recognition of the target semantic.

4.2 SESA mechanism

SESA generates the joint representation of all events $E_{SESA} = \{e_{s1}, ..., e_{sw}\}$ mentioned in the sentence X, where $e_{si} \in E$, to enhance event semantics of tokens in X. To generate an accurate representation, inspired by Gururangan et al. (2020), we utilize a sentence-level event classification task and the same training dataset as the EE task to fine-tune SESA. Due to the rich general knowledge in PLMs (Devlin et al., 2019; Lewis et al., 2020), we use BERT (Devlin et al., 2019) as the backbone.

Hence, the following discussion outlines the components of SESA, focusing on the learning and generation of sentence event representations.

Global encoder generates the global representation of all events mentioned in the input sentence with all tokens, to train and test SESA. The input sequence $X^{'}$ is constructed by adding the (CLS) token at the beginning of X, with all the corresponding masks in $ATTN\_MASK$ set to 1. The embedding of (CLS), denoted as $\boldsymbol{SG_{CLS}} = BERT(X^{'}, ATTN\_MASK)$, serves as the global representation of the sentence's events. During experiments, we observed that the embedding of (CLS) in the last layer hidden state of the BERT output outperforms its counterpart in the pooler_output.

Event encoder generates a single high-dimensional vector representing all triggers and arguments in the sentence, exclusively for training purposes. The input sentence of the event encoder aligns with that of the global encoder. After filtering out tokens irrelevant to events, the representation of the sentence's pure events is $\boldsymbol{SE_{CLS}} = BERT(X, EVENT\_MASK)$, where $EVENT\_MASK$ designates masks, corresponding to (CLS) and tokens in X labeled as triggers and arguments, with the value of 1. Experiments illustrate that the pure event representation enhances the global representation of the sentence's event when modeling, ensuring the accuracy of modeling all events mentioned in the sentence.

Event classifier identifies events with the joint representation of all events mentioned in the sentence S as follows. (1) Representation generation. During training, the output of the global encoder and the event encoder generates $\boldsymbol{S} = [\boldsymbol{SG_{CLS}}; \boldsymbol{SE_{CLS}}]$. However, during inference, only the global representation of all the events mentioned in the sentence is used, so $\boldsymbol{S} = \boldsymbol{SG_{CLS}}$. (2) Event identification. The event classifier comprises a feed-forward network (FFN) and an activation function. The FFN consists of a single-layer network structure and an ReLU function. The classifier generates an event probability vector $\boldsymbol{P} = [p_1, ..., p_M]$, where p_i denotes the probability of the occurrence of event type i. The threshold t is used to identify event types mentioned in X:

$$\hat{\boldsymbol{E_X}} = [e_1^{'}, ..., e_M^{'}], \underset{i\in[1,...,M]}{e_i^{'}} = \begin{cases} 0, & p_i < t \\ 1, & p_i \geq t \end{cases} \tag{9}$$
$$\boldsymbol{E_{SESA\_X}} = \hat{\boldsymbol{E_X}} \odot E = [e_{s1}, ..., e_{sw}]$$

where $e_i^{'} \in \{0, 1\}$, which indicates whether x_i triggers type e_i. When $e_{si}^{'} = 1$ and $e_{si} \in \boldsymbol{E_{SESA\_X}}$, it indicates that the sentence X triggers event type e_{si}. To streamline training, we utilize $pr(x, \mathcal{K}_{r,u})$ as the activation function and employ BCE loss for optimization.

$$\begin{aligned}\mathcal{L}(\boldsymbol{Y},\hat{\boldsymbol{Y}}) = -\textstyle\sum_{i=1}^{M}[g_i y_i log(pr(X))+ \\ (1-g_i)(1-y_i)log(1-pr(X))]\end{aligned} \tag{10}$$

where g_i denotes the weight of the event type e_i, $pr(X) = pr(X, \mathcal{K}_{r,u})$. To calculate g_i, we use the reciprocal for the ratio of the annotated samples of event type e_i to the total number of annotated samples in the dataset.

4.3 Semantic-enhanced encoder

The semantic-enhanced encoder transforms tokens in the input sentence X into real-valued word embedding. To mitigate the interference from non-target event semantics and relevant semantics, we encode the knowledge of all the events mentioned in X into the representation of each token, enhancing the token's event semantics. Specifically, we utilize the BERT fine-tuned by SESA to derive the event representation of X, denoted as $\boldsymbol{S_X} = SESA(X, ATTN\_MASK)$, where $S_X \in \mathcal{R}^{1\times d}$ and d represents the hidden layer dimension of BERT. We use the fine-tuned BERT to encode the vector representation of each token in X as $\{\boldsymbol{w_1}, ..., \boldsymbol{w_n}\} = BERT(x_1, ..., x_n)$, where $\boldsymbol{w_i} = [w_1^i, ..., w_d^i]$. The token's enhanced representation, incorporating the knowledge of the sentence's event, is expressed as $\boldsymbol{w_i^{'}} = [\boldsymbol{w_i}; \boldsymbol{S_X}]$.

4.4 Task decoder

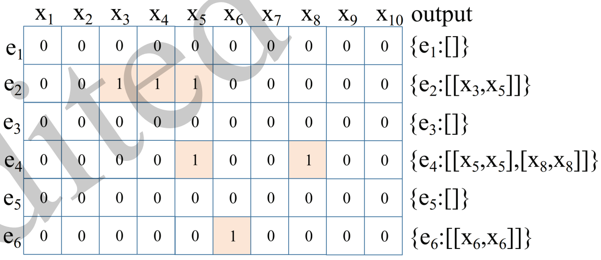
The task decoder recognizes the boundaries of candidate triggers and arguments in the sentence and classifies their types. Given that triggers and arguments are associated with distinct label sets and utilize different decoding modes, we develop task-specific decoders for triggers and arguments, respectively.

4.4.1 Trigger decoder

The trigger decoder consists of M semantic decoders, each responsible for recognizing its trigger boundaries, based on the probability distribution of tokens, and classifying their types. Before decoding, we utilize the event classifier described in Section 4.2 to obtain the probability distribution $p(x_i, S)$ of each token x_i. The decoding process includes the following steps.

(1)*Classification.* Commonly used methods identify and classify triggers using maximum probability leading to misclassification of low-frequency semantics. To address the issue, we employ a task-specific threshold value as the criterion for judgment. By applying t_t, based on Eq. (9) and the probability distribution of x_i, we derive the predicted type set for x_i as $E_{xi} = \{e_{xi1}, ..., e_{xit}\}$, where $e_{xii} \in E$ and $p_{xii} \geq t_t$.

(2)*Boundary identification.* We employ the semantic decoders and threshold to recognize the trigger's boundaries based on the predicted type set of tokens. Semantic decoder i identifies the boundaries of all triggers for type e_i, where trigger boundaries are determined by consecutive tokens in the sentence that triggers the same event type. The decoding process for the trigger decoder is shown in Fig. 4.



	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	output
e1	0	0	0	0	0	0	0	0	0	0	{e1:[]}
e2	0	0	1	1	1	0	0	0	0	0	{e2:[[x3,x5]]}
e3	0	0	0	0	0	0	0	0	0	0	{e3:[]}
e4	0	0	0	0	1	0	0	1	0	0	{e4:[[x5,x5],[x8,x8]]}
e5	0	0	0	0	0	0	0	0	0	0	{e5:[]}
e6	0	0	0	0	0	1	0	0	0	0	{e6:[[x6,x6]]}

Fig. 4 The process of the trigger decoding with the semantic decoders. 1 in (i, j) indicates that the predicted type for the token x_j is e_i, and 0 in (i, j) indicates that the token x_j does not trigger type e_i. Each row in this figure is a semantic decoder that identifies the boundaries of all candidates for the semantic. Each column is the predicted type set of the token.

4.4.2 Argument decoder

We design a head decoder and a tail decoder for each semantic in the EAE task and use the task-specific threshold t_r to identify roles for arguments, guiding the model in modeling the boundaries distribution of arguments.

(1)*Boundary modeling.* To accurately model boundaries of arguments, following previous studies (Yang et al., 2019; Du and Cardie, 2020; Yang et al., 2021), we adopt the head/tail labeling scheme to annotate boundaries of arguments. The probability distribution of token is x_i is $[p_{cr0}, ..., p_{crm}] = pr(FFN_c(\boldsymbol{w_i^{'}}), \mathcal{K}_{r,u})$, where $c \in \{head, tail\}$, FFN_{head} and FFN_{tail} are the head and tail role classifiers, respectively, and p_{cri} indicates the probability of x_i being the head or tail of the argument for role r_i.

(2)*Boundary identification.* We employ t_r to identify the role types that x_i plays, following Eq. (9). The widely used boundary identification method is the enumeration (Wadden et al., 2019; Du and Cardie, 2020), which enumerates all predicted head-tail position combinations and identifies target boundaries with the heuristic method. However, we adopt the heuristic matching principle proposed by Yang et al. (2019), which selects the tail closest to the head as the target argument. The process of identifying head positions and tail positions of each semantic is analogous. Taking the head position recognition of class i as an example, the number of head positions is determined by the number of candidate chunks. Subsequently, the token with the highest probability is selected as the head of the chunk. The detailed decoding process of the role decoder is illustrated in Fig. 5.

4.5 Training

We design task-specific loss functions for ED and EAE respectively to train the model, intending to learn the semantic distribution of polysemous triggers and arguments. Our model aims to intensify the learning of misjudged semantics by increasing the loss while diminishing the learning of correct semantics through loss reduction. To accomplish this, we employ BCE loss for training, based on the difference between the predicted probability distribution and the gold probability distribution. The loss functions for ED and EAE are formulated as follows.

$$\begin{aligned} \mathcal{L}_t &= -\sum_{j=1}^{M}[g_j y_j log(pr) + (1-g_j)(1-y_j)log(1-pr)] \\ \mathcal{L}_{role}^{c} &= -\sum_{j=1}^{m}[y_j log(pr_c) + (1-y_i)log(1-pr_c)] \end{aligned} \tag{11}$$

where $\mathcal{L}_t$ and $\mathcal{L}_{role}^{c}$ indicate the loss function for training the ED and EAE model, respectively, f is the task-specific classifier, $pr = pr(f(\boldsymbol{w_i'}), \mathcal{K}_{r,u})$, and $pr_c = pr(f_c(\boldsymbol{w_i'}), \mathcal{K}_{r,u})$.

5 Theoretical analysis of the reward–penalty mechanism

In this section, we theoretically analyze the effectiveness of the reward–penalty mechanism from the perspective of enlarging the gap between semantics and achieving balanced semantics learning.

5.1 Analysis from the perspective of enlarging the gap

Function $pr(x, \mathcal{K}_{r,u})$ augments the maximum gradient value for better training and enlarges the gap between the relevant and irrelevant semantics. The maximum gradient value of $pr(x, \mathcal{K}_{r,u})$ is $\mathcal{K}_{r,u}$ times larger than that of the sigmoid function, which effectively mitigates gradient vanishing during backpropagation, as depicted in the following equation.

$$\max(\frac{pr'(x, \mathcal{K}_{r,u})}{\delta'(x)}) = \mathcal{K}_{r,u} \tag{12}$$

It is evident that the larger the discrepancy between the probabilities of target and nontarget semantics, the simpler it is to set the threshold t for classification. As depicted in Fig. 6, the maximum value of $(pr'(x, \mathcal{K}_{r,u}) - \delta'(x))$ increases as $\mathcal{K}_{r,u}$ grows when $\mathcal{K}_{r,u}$ exceeds 1. However, the effective range of the reward–penalty mechanism, denoted as $[-x_e, x_e]$, decreases, where $\delta'(\pm x_e) = pr'(\pm x_e, \mathcal{K}_{r,u})$. When $x \in [-x_e, x_e]$ and $\Delta x \geq 0$, if $pr'(x, \mathcal{K}_{r,u}) \geq \delta'(x)$, then $(pr(x, \mathcal{K}_{r,u}) - pr((x - \Delta x), \mathcal{K}_{r,u})) \geq (\delta(x) - \delta(x - \Delta x))$. Thus, the reward–penalty function $pr(x, \mathcal{K}_{r,u})$ widens the gap between semantics, thereby reducing misclassifications.

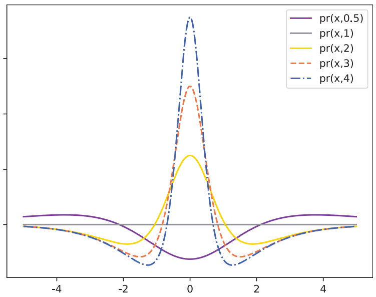
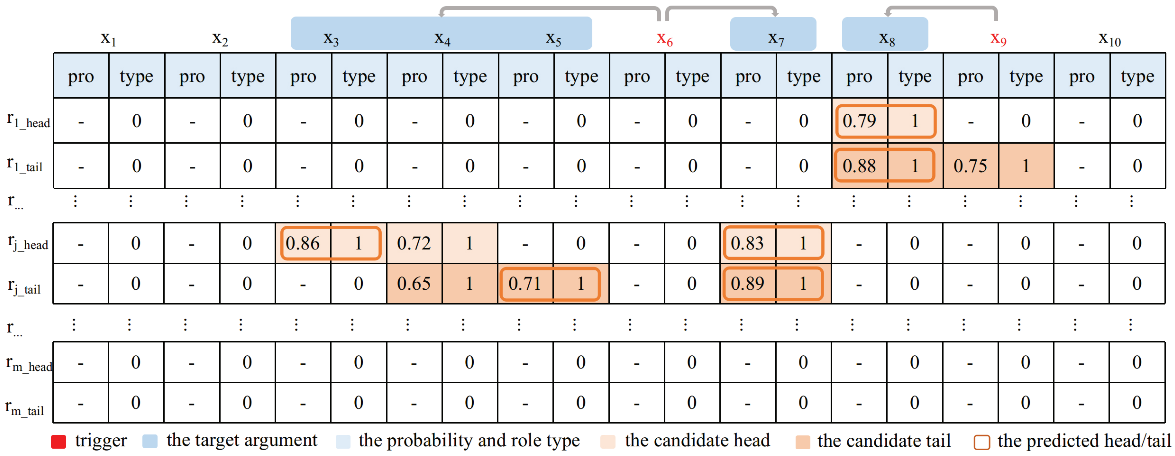


Fig. 6 Results of $(pr'(x, \mathcal{K}_{r,u}) - \delta'(x))$.

5.2 Analysis from the perspective of balanced learning semantics

The reward–penalty mechanism aims to rebalance the distribution of semantics by adjusting the training loss. This loss of a token's semantics comprises two components: the loss associated with target semantics and nontarget semantics, respectively, as illustrated in Eq. (13). To address misjudgments, the reward–penalty function $pr(x, \mathcal{K}_{r,u})$ fine-tunes the model with a penalty mechanism to enhance the understanding of the target semantics



	x_1		x_2		x_3		x_4		x_5		x_6		x_7		x_8		x_9		x_{10}	
	pro	type	pro	type	pro	type	pro	type	pro	type	pro	type	pro	type	pro	type	pro	type	pro	type
r_{1_head}	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.79	1	-	0	-	0
r_{1_tail}	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.88	1	0.75	1	-	0
$r_{...}$	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
r_{j_head}	-	0	-	0	0.86	1	0.72	1	-	0	-	0	0.83	1	-	0	-	0	-	0
r_{j_tail}	-	0	-	0	-	0	0.65	1	0.71	1	-	0	0.89	1	-	0	-	0	-	0
$r_{...}$	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
r_{m_head}	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
r_{m_tail}	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0

Fig. 5 The process of the role decoding, where âĂIJproâĂİ is the abbreviation of the probability indicating the probability of the token on the type, and the âĂIJtypeâĂİ value ($\in \{0, 1\}$) means whether the type is the predicted type of the token. The results of role decoding is $\{e_{x_6} : \{r_j : [[x_3, x_5], [x_7, x_7]]\}, e_{x_9} : \{r_1 : [[x_8, x_8]]\}\}$, where e_{x_i} is the event type for token x_i.

and decrease the learning of nontarget semantics, thereby widening the gap between target and nontarget semantics. Concurrently, the reward mechanism is employed to mitigate the model's learning of correctly classified semantics, aiming to reduce the gap between relevant semantics and achieve a balanced semantic distribution.

$$\mathcal{L}(Y_{i,j}, \hat{Y}_{i,j}) = -[\underbrace{y_{i,j} log(pr(f(x_i), \mathcal{K}_{r,u}))}_{the\ target\ semantic\ loss} + \underbrace{(1 - y_{i,j}) log(1 - pr(f(x_i), \mathcal{K}_{r,u}))]}_{the\ nontarget\ semantic\ loss} \tag{13}$$

where $Y_{i,j}$ and $\hat{Y}_{i,j}$ are the ground truth value and predicted probability value of token x_i on the class j, respectively, $y_{i,j} \in \{0, 1\}$.

The penalty mechanism guides the model to accurately learn misclassified semantics by enlarging the loss, as illustrated by the red region in Fig 7. Suppose token x_i labeled with class j with $p_{i,j} < 0$ and $t = 0.5$, x_i is unidentified as class j, resulting in an FN. Meanwhile, for token x_i not labeled with class b with $p_{i,b} > 0$, x_i is identified as class b, leading to an FP. Referring to Eq. (13) and Fig 7, it is apparent that when $\mathcal{K}_{r,u} > 1$, FPs and FNs lead to an increase in loss, which is equivalent to imposing a penalty.

The reward mechanism reduces the gap among relevant semantics by decreasing the loss, as the blue region depicted in Fig 7. When x_i is a sample for class j and is accurately classified as j, yielding a TP. Referring to Eq. (13) and Fig 7, it is observed that $loss_{pr} < loss_{\delta}$ in the case a TP occurs and $\mathcal{K}_{r,u} > 1$. The TP essentially leads to a diminished loss, akin to receiving a reward.

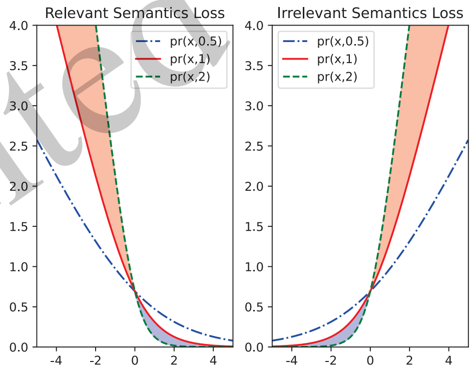


Fig. 7 Loss of relevant and irrelevant semantics

6 Experiments

6.1 Experimental setup

6.1.1 Datasets

We evaluate our model on two public EE benchmarks, the ACE2005 corpus [1] and the Rich Entities, Relations and Events corpus (ERE) (Song et al., 2015) [2], primarily using their English corpora, denoted as ACE2005-E+ and ERE-EN, respectively. Following Lin et al. (2020) and Hsu et al. (2022), ACE2005-E+ is spilt into three parts: the training

[1] `https://catalog.ldc.upenn.edu/LDC2006T06`

[2] Here we use datasets LDC2015E29,LDC2015E68, and LDC2015E78 in ERE.

set with 529 documents, the validation set with 30 other documents, and the test set with the remaining 40 newswire documents. Additionally, following the pre-processing in Lin et al. (2020), we obtain the variant ACE05-E. Both have been annotated with 33 event types and 22 argument roles. The pre-processing of ERE-EN follows Lin et al. (2020) involving 38 event types and 21 argument roles. Moreover, we add a âĂIJNULLâĂİ type for tokens without annotation. ACE05-E differs from ACE05-E+ and ERE-EN in that the latter are more complex datasets with multi-token triggers. Statistical results of ACE05-E, ACE05-E+, and ERE-EN are shown in Table 2.

Table 2 Datasets statistics results

Dataset	Split	Sents	Events	Roles
ACE05-E	Train	17,172	4202	4859
	Dev	923	450	605
	Test	832	403	576
ACE05-E+	Train	19,216	4419	6607
	Dev	901	468	759
	Test	676	424	689
ERE-EN	Train	14,736	6208	8924
	Dev	1209	525	730
	Test	1163	221	822

6.1.2 Evaluation metrics

We adhere to the criteria employed in previous studies (Wadden et al., 2019; Hsu et al., 2022). (1) *Trigger Identification* (Trig-I): a trigger is correctly identified only if its predicted span matches that of the gold trigger perfectly. (2) *Trigger Classification* (Trig-C): the event type of the trigger is correctly classified if only its predicted type matches that of the gold trigger. (3) *Argument Identification* (Arg-I): an argument is correctly identified only if its predicted span matches that of the gold argument. (4) *Argument Classification* (Arg-C): the role type of an argument is correctly classified only if its predicted role type and event type match that of the gold argument. Simultaneously, we utilize the widely used evaluation metrics, including Precision (P), Recall (R), and Micro F1 score (F1), to assess the performance.

6.1.3 Parameter settings

We conduct all experiments on one NVIDIA 3090 GPU, with a learning rate of 1e-5 and weight decay of 1e-5 for BERT, and a learning rate of 1e-4 and weight decay of 1e-2 for the other models. The batch size is 32. The epoch for EAE is 50 and the other is 30. The dropout rate is 0.5. We set our seed 42. The threshold for each task is t_e, t_t, t_r, respectively. We employ AdamW (Loshchilov and Hutter, 2019) to optimize the model, and the maximum gradient clipping is set to 5 to avoid over-fitting.

6.1.4 Baselines

Single-task EE models solely rely on event annotations for EE. In contrast, multi-task EE models perform EE with the help of named entity recognition, relation extraction, or entity annotations. Since not all event corpora extensively annotate entities and relationships, EE models relying solely on event annotations are more versatile.

Single-task EE models: (1) DMCNN (Chen et al., 2015) utilizes dynamic multi-pooling CNN to capture features of word-level and sentence-level; (2) BERT_QA (Du and Cardie, 2020) formalizes the EE task as a QA and designs task-specific question templates for the trigger extraction and argument extraction; (3) LEAR (Yang et al., 2021) enhances tokens' task semantics by encoding the label annotation into the token's representation; (4) TEXT2EVENT (Lu et al., 2021) uses a curriculum learning approach and constrained decoding to accomplish sequence-to-structure tasks in document-level EE; (5) DEGREE(Hsu et al., 2022) proposes an end-to-end event-generation model that generates events from predefined event type-specific templates; (6) GTEE-DYNPREF (Liu et al., 2022) is a template-based generative EE method, that adopts dynamic prefix-tuning technique; (7) DAEE (Wang et al., 2023a) enhances EE by utilizing reinforcement learning and event knowledge to generate high-quality data to augment EE; (8) DemoSG (Zhao et al., 2023) utilizes knowledge of the annotated data and label semantics to conduct EE in low resources and (9) ChatGPT-ICL (Han et al., 2023) is a prompt-based inference-only method, that conducts 14 subtasks of IE to evaluate the performance and robustness of ChatGPT.

Multi-task EE models: (1) DYGIE++ (Wadden et al., 2019) learns distant contextual features by using dynamic span graphs; (2) ONEIE (Lin et al., 2020) obtains optimal event graphs by using beam search and global features and (3) UniEX (Ping et al., 2023) proposes the triaffine attention mechanism to encode the schema of all tasks and their label semantics into token semantics to fully improve the comprehensive semantics.

6.2 Effectiveness

In this section, we conduct extensive experiments on the three datasets to assess the effectiveness of RPEE. Detailed content related to the case study is provided in Supplementary Materials Section 1. A comprehensive discussion of RPEE is presented in Section 2 of the Supplementary Materials.

6.2.1 Main results

Table 3 illustrates the experimental results of all baselines and our method on ACE05-E. When comparing the performance, we obtain the following findings: (1) Our method surpasses all baselines in terms of F1 score. This indicates the effectiveness of the proposed RPEE on the EE task. (2) In terms of Trig-C, compared with the state-of-the-art methods of both single-task and multi-task EE, our method achieves relative performance improvements of 5.2% and 3.7% on Trig-C F1, respectively. Our method does not use manually crafted prompts or complex language models or NLP tools, and achieves significant results with limited annotated data, showcasing strong scalability and generalization. (3) In terms of Arg-C, our model exhibits performance enhancements of 3.9% and 4.4% compared with single-task EE and multi-task EE models, respectively, highlighting the effectiveness of our approach in this task. (4) Concerning PLMs, our method not only outperforms the baselines that also use BERT-base but also outperforms the baselines that employ larger PLMs, such as BART-large, demonstrating the superiority of our approach in applications.

To further verify the scalability and robustness of our method, we conducted experiments on ACE05-E+ and ERE-EN, and present results in Table 4. Upon analysing the results, we derive two crucial conclusions.

•*High scalability.* Our approach has demonstrated strong performance on ACE05-E+ and ERE-EN. For trigger and Arg-C, our method surpasses all baseline methods on Trig-C and Arg-C in terms of the F1 score. Notably, we observe a relative enhancement of 2.9% and 0.1% for F1 scores in Trig-C, and a relative improvement of 8.0% and 6.4% for F1 scores in Arg-C, respectively. These findings underscore the scalability and efficacy of our method, signifying its suitability for EE across diverse domains.

Table 4 Experimental results on ACE05-E+ and ERE-EN

Model	ACE05-E+		ERE-EN	
	Trig-C	Arg-C	Trig-C	Arg-C
ONEIE	72.8	54.8	59.1	50.5
LEAR*	71.4	-	57.0	-
TEXT2EVENT	71.8	54.4	59.4	48.3
DEGREE	70.9	56.3	57.1	49.6
GTEE-DYNPREF	74.3	54.7	66.9	55.1
DAEE	76.9	56.3	65.0	51.6
RPEE (Ours)	**79.1**	**60.8**	**67.0**	**58.7**

The highest scores are highlighted in bold, while sub-optimal scores are underlined. The symbol $*$ denotes results obtained by using the same dataset and data pre-processing outlined in this paper. Arg-C, argument classification; Trig-C, trigger classification.

•*Strong robustness.* Our method exhibits superior performance on ACE05-E+ compared to all baselines and even outperforms its performance on ACE05-E, as shown in Tables 3 and 4. Notably, baselines generally perform better on ACE05-E than ACE05-E+. The discrepancy is attributed to the presence of multi-tokens, posing a substantial challenge for models to precisely model trigger and argument boundaries. Our method adeptly leverages multi-token instances, enhancing model performance via the reward–penalty mechanism and the task-specific decoding strategy. Experimental results affirm the robustness of our approach in handling intricate scenarios involving multi-tokens.

6.2.2 Ablation study

This section focuses on a detailed analysis of the impact of each component on performance, with corresponding experimental results on ACE05-E+ presented in Table 5.

• **w/o SESA** indicates the variant without the SESA module. Significantly, there is a notable performance decrease compared with RPEE, affirming the effectiveness of the SESA proposed in Section 4.2. This discrepancy is attributed to the representation of the sentence's event that is well-learned during

Table 3 EE results on ACE05-E

Task	Model	PLMs	Trig-C	Arg-C
Multi-Task EE	DYGIE++ (Wadden et al., 2019)	BERT-base	73.6	52.5
	ONEIE (Lin et al., 2020)	BERT-base	74.7	56.8
	UniEX (Ping et al., 2023)	RoBERTa-large	74.1	53.9
Single-Task EE	DMCNN (Chen et al., 2015)	-	69.1	53.5
	BERT_QA (Du and Cardie, 2020)	BERT-base	72.4	53.3
	LEAR* (Yang et al., 2021)	BERT-base	72.2	-
	TEXT2EVENT (Lu et al., 2021)	T5-large	71.9	53.8
	DEGREE (Hsu et al., 2022)	BART-large	73.3	55.8
	GTEE-DYNPREF (Liu et al., 2022)	BART-large	72.6	55.8
	DAEE (Wang et al., 2023a)	BART-large	75.8	56.5
	DemoSG$_R$ (Zhao et al., 2023)	BART-large	73.4	56.0
	ChatGPT-ICL$_{5shot}$ (Han et al., 2023)	gpt-3.5-turbo	27.3	31.6
	RPEE (Ours)	BERT-base	**78.6**	**59.0**

The highest scores are highlighted in bold, while suboptimal scores are underlined. In PLMs, âĂIJ-âĂİ indicates the absence of PLM usage. The symbol $*$ denotes results obtained by using the same dataset and data pre-processing outlined in this paper. Arg-C, argument classification; EE, event extraction; PLMs, pre-trained language models; Trig-C, trigger classification.

Table 5 Ablation study on ACE05-E+

model	**Trig-I**			**Trig-C**			**Arg-I**			**Arg-C**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
RPEE	**85.22**	**79.18**	**82.09**	**81.63**	**76.79**	**79.14**	**73.95**	**64.91**	**69.13**	**62.31**	**59.37**	**60.80**
w/o SESA	84.46	*75.08*	79.49	81.19	72.77	76.75	68.11	60.20	63.91	55.89	56.10	56.00
w/o Re-weighting	84.19	77.47	80.69	79.54	73.87	76.60	70.71	62.41	66.30	54.97	56.97	55.95
w/o Event Encoder	*82.22*	76.01	*78.99*	*78.41*	73.31	75.77	66.76	63.57	65.12	52.84	57.78	55.20
w/o SA	82.86	75.26	78.88	78.82	*71.81*	*75.15*	72.56	*52.85*	*61.16*	59.56	*45.05*	*51.30*
w/o Reward–Penalty	82.82	76.15	79.35	78.71	72.66	75.56	*61.97*	63.73	62.83	*50.14*	57.86	53.72

The highest scores are highlighted in bold, while the lowest scores for all model variants are indicated in italics. Arg-C, argument classification; Arg-I, argument identification; SA, situation awareness; SESA, sentence event situation awareness; Trig-C, trigger classification; Trig-I, trigger identification.

the SESA pre-training, offering event constraints for triggers in the sentence.

- **w/o Event Encoder** denotes the absence of the event encoder in the SESA module. Compared with the âĂIJw/o SESAâĂİ variant, âĂIJw/o Event EncoderâĂİ learns the representation of the sentence's event, and the performance instead decreases. It illustrates that pre-training without pure event knowledge hinders the accurate acquisition of sentence event knowledge, resulting in a degraded model.

- **w/o Re-weighting** signifies ignoring the imbalanced distribution of events. The findings suggest that this uneven distribution impacts the learning of event semantics, thereby influencing the overall performance of EE.

- **w/o SA** designates the variant that omits the utilization of the representation of the sentence's event, resulting in the poorest performance among all the variants. It illustrates the significance of incorporating the representation of all the events mentioned in the sentence, as it effectively enhances the event semantics of tokens within the sentence. Enhanced representation offers vital event constraints for triggers. The findings emphasize the pivotal role of the SA in the overall model effectiveness.

- **w/o Reward–Penalty** indicates the variant without using the reward–penalty mechanism, displaying inferior performance compared to most variants. The declining performance underscores the crucial role of the reward–penalty mechanism in achieving a balanced modeling of the token's event distribution within the model.

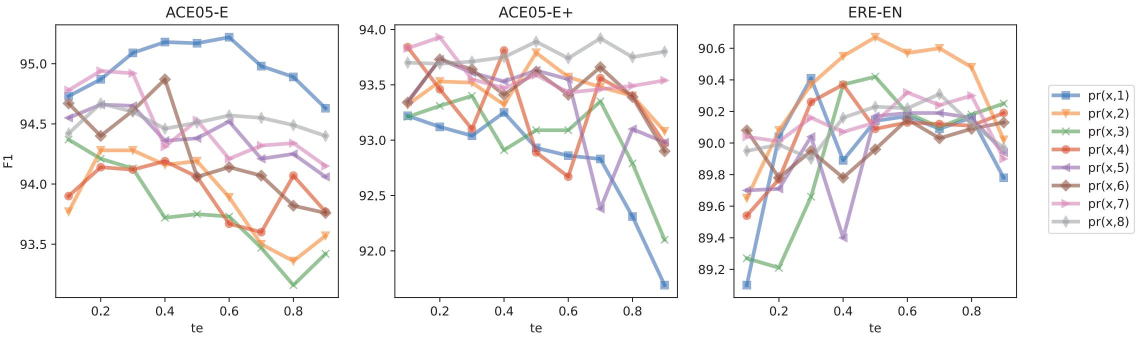


Fig. 8 Performance of SESA on ACE05-E, ACE05-E+, and ERE-EN. SESA, sentence event situation awareness.

6.2.3 Analysis of the SESA

In this section, we conduct experiments on ACE05-E, ACE05-E+, and ERE-EN to analyze the effectiveness of SESA and the impact of the event encoder, as shown in Fig. 8. Results reveal that SESA excels in the sentence event classification task, achieving remarkable F1 values surpassing 93, 92, and 89, respectively. It indicates that SESA has well learned the representation of all events mentioned in the sentence. When fixing $\mathcal{K}_{r,u}$, the performance varies with the adjustment of t_e. There exists an optimal t_e for $\mathcal{K}_{r,u}$ to achieve the best performance. Furthermore, experimental results suggest that utilizing the reward–penalty mechanism has a negligible impact on SESA's performance. To reduce the complexity of training, SESA adopts the identical configuration of the reward–penalty function with Trig-C.

To further assess the influence of pure event knowledge on modeling all the events mentioned in the sentence, we conducted experiments by excluding the event encoder module, with detailed results in Table 6. Experimental results emphasize that relying solely on the knowledge of all the tokens in the sentence for modeling the sentence's event yields unsatisfactory results, leading to severe errors in downstream tasks, as depicted in Section 6.2.2. The findings underscore the pivotal role of pure event knowledge in effectively modeling the sentence's event.

6.2.4 Sensitivity test

We analyze the impact of different configurations of key hyper-parameters in RPEE, specifically, $\mathcal{K}_{r,u}$ in the reward–penalty mechanism and t_t during the decoding process. These hyper-parameters are individually adjusted, while the remaining parameters remain consistent with the previously reported settings. Taking the ED task as the case study, we conduct experiments with various settings of $\mathcal{K}_{r,u}$ and t_t on ACE05-E, ACE05-E+, and ERE-EN. The plot in Fig. 9 illustrates the fluctuation of the P, R, and F1 score for Trig-I and Trig-C across different hyper-parameters setting on ACE05-E. It is evident from the figure that both R and F1 decrease as $\mathcal{K}_{r,u}$ or t_t increases. P increases with the increment in $\mathcal{K}_{r,u}$ and t_t, indicating a positive correlation between the hyper-parameters and precision. When $\mathcal{K}_{r,u} = 1$, $pr(x,1) = \delta(x)$ denotes the absence of the reward–penalty mechanism. The improvement in P, R, and F1 score demonstrates the efficacy of the reward–penalty mechanism.

Fig. 10 illustrates the variation in the Trig-C F1 score on ACE05-E+ and ERE-EN, respectively. Experimental results suggest that an increased value of $\mathcal{K}_{r,u}$ or t_t does not consistently improve the model's performance. Different tasks exhibit optimal performance under the specific $\mathcal{K}_{r,u}$ and t_t, demonstrating the flexibility and superiority of the reward–penalty mechanism.

6.2.5 Polysemy test

In this section, we conduct experiments to verify the efficiency of RPEE when dealing with polysemous triggers. To assess the effect on the polysemy and monosemy, we divide the test datasets of ACE05-E, ACE05-E+, and ERE-EN into monosemous and polysemous test datasets, based on whether sub-tokens of triggers have multiple semantics. Table 7 displays the performances of RPEE and its variants on the Trig-C F1 score. The performance of RPEE on the polysemous test dataset of ERE-EN outperforms the monosemous one, while for

Table 6 Ablation study of the event decoder for the SESA

Model	ACE05-E			ACE05-E+			ERE-EN		
	P	R	F1	P	R	F1	P	R	F1
RPEE	96.47	92.66	94.53	94.86	92.54	93.69	89.36	90.05	89.70
w/o Event Encoder	82.79	80.36	81.56	78.85	77.63	78.24	74.91	79.79	77.27
Δ	16.52	15.31	15.90	20.30	19.21	19.75	19.29	12.86	16.09

Δ signifies the relative performance gain obtained using pure event knowledge. SESA, sentence event situation awareness.

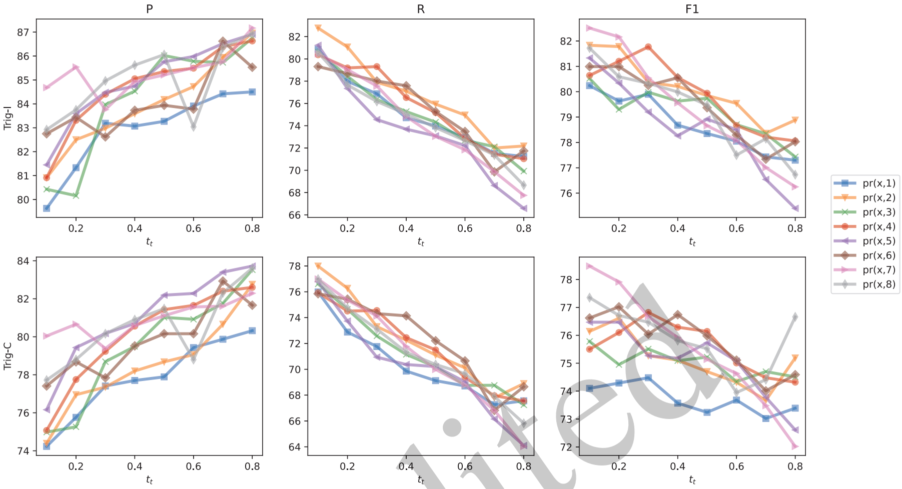


Fig. 9 Performance of Trig-I and Trig-C on ACE05-E with different settings of two hyper-parameters ($\mathcal{K}_{r,u}$ and t_t). Trig-I, trigger identification.

Table 7 Experimental results on the monosemous and polysemous triggers

	Trig-I			Trig-C		
	P	R	F1	P	R	F1
	ACE05-E					
RPEE_full	85.67	80.12	82.80	80.84	76.42	78.57
RPEE_monosemous_triggers	87.23	77.04	81.82	83.89	75.03	79.21
RPEE_polysemous_triggers	93.25	81.12	86.77	81.09	73.89	77.32
w/o reward_penalty RPEE_monosemous_triggers	74.40	78.30	76.30	74.21	78.10	76.11
w/o reward_penalty RPEE_polysemous_triggers	80.68	81.43	81.06	68.33	70.51	69.40
	ACE05-E+					
RPEE_full	85.22	79.18	82.09	81.63	76.79	79.14
RPEE_monosemous_triggers	83.20	77.30	80.14	82.09	76.66	79.28
RPEE_polysemous_triggers	88.88	81.08	84.80	78.66	75.40	76.99
w/o reward_penalty RPEE_monosemous_triggers	74.33	82.99	78.42	71.96	82.00	76.65
w/o reward_penalty RPEE_polysemous_triggers	74.70	83.95	79.06	63.49	77.58	69.83
	ERE-EN					
RPEE_full	75.24	78.14	76.62	63.78	70.51	66.97
RPEE_monosemous_triggers	74.45	74.09	62.04	62.04	65.13	63.55
RPEE_polysemous_triggers	74.34	86.30	79.88	60.42	77.25	67.81
w/o reward_penalty RPEE_monosemous_triggers	66.67	66.16	66.42	55.98	57.77	56.86
w/o reward_penalty RPEE_polysemous_triggers	71.24	80.85	75.74	53.19	69.41	60.23

Trig-C, trigger classification; Trig-I, trigger identification.

ACE05-E and ACE05-E+ the opposite is true. The reason is that the number of monosemous and polysemous test datasets for ERE-EN is nearly equal, whereas, for ACE05-E and ACE05-E+, the monosemous dataset is 50% larger than the polysemous dataset. It demonstrates that RPEE can effec-

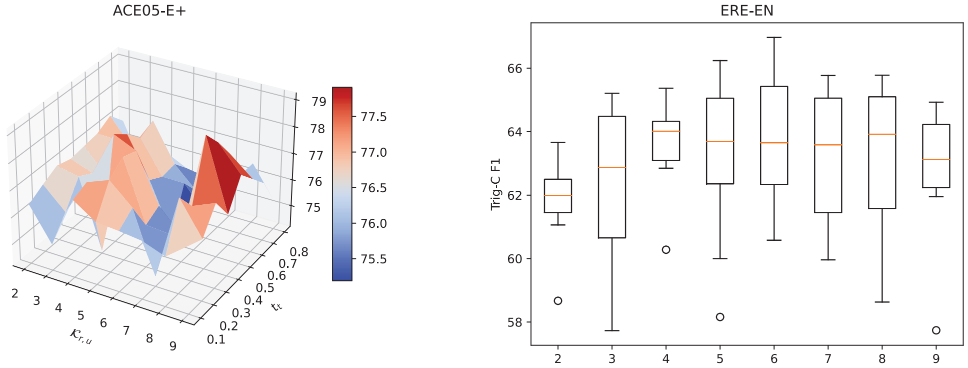


Fig. 10 Trig-C F1 results on ACE05-E+ and ERE-EN with varying $\mathcal{K}_{r,u}$ and t_t.

tively handle triggers with multiple semantics. When utilizing the reward–penalty mechanism, there is a relative improvement of 11.4%, 1.3%, and 12.6% for the polysemous datasets of ACE05-E, ACE05-E+, and ERE-EN, respectively. It indicates that the reward–penalty mechanism effectively addresses the challenges posed by polysemy. Hence, we can confidently conclude that our approach can handle datasets with polysemous triggers and arguments, showcasing strong robustness.

6.2.6 EAE with gold triggers

We conducted comparative experiments using gold triggers on ACE05-E, ACE05-E+, and ERE-EN to explore the potentiality of our model. As depicted in Table 8, it achieves relative F1 gains of 4.4%, 6.7%, and 7.8% for Arg-C on ACE05-E, ACE05-E+, and ERE-EN, respectively. It demonstrates that our approach effectively handles EAE tasks, irrespective of using predicted triggers or gold triggers. Additionally, we observe a decrease in the EAE performance when neglecting the reward–penalty mechanism, further affirming the reliability and effectiveness of our designed reward–penalty mechanism.

7 Conclusions

In this paper, we propose an adaptive semantics learning strategy to mitigate the bias in the semantics distribution of polysemous triggers and arguments. We design a reward–penalty mechanism to enlarge the gap between the relevant semantics and irrelevant semantics and diminish the gap between relevant semantics by rewarding the corrected classified semantics and punishing the misclassified semantics. The sentence-level event semantics, pretrained by using a sentence-level event SA mechanism to ensure accuracy, are integrated into token representations to narrow the target event scope of triggers. The model identifies the boundaries of triggers and arguments and classifies their types using task-specific semantic decoders. Our experiments show our model's strengths in robustness, scalability, and generalization ability in complex scenarios. In the future, we will extend our model to low-resource scenarios.

Contributors

Hai-li LI designed the research. Hai-li LI, Yun-yan ZHOU, and Jie ZHOU processed the data. Hai-li LI drafted the manuscript. Zhi-liang TIAN, Yun-yan ZHOU, and Qiu-bo XU helped organized the manuscript. Hai-li LI, Zhi-liang TIAN, Dong-sheng LI, Xiao-dong WANG, and Shi-long PAN revised and finalized the paper.

Compliance with ethics guidelines

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ahn D, 2006. The stages of event extraction. Proc Workshop on Annotating and Reasoning about Time and Events, p.1-8.
https://doi.org/10.3115/1629235.1629236

Akgun SA, Ghafurian M, Crowley M, et al., 2023. Using affect as a communication modality to improve human-robot communication in robot-assisted search and rescue scenarios. *IEEE Trans Affect Comput*, 14(4):3013-3030.
https://doi.org/10.1109/TAFFC.2022.3221922

Anelli VW, Di Noia T, Di Sciascio E, et al., 2022. Inferring user decision-making processes in recommender systems

Table 8 Performance of EAE using gold triggers on ACE05-E, ACE05-E+, and ERE-EN

Model	ACE05-E		ACE05-E+		ERE-EN	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
RPEE	66.43	58.96	69.13	60.80	69.47	58.57
with gold_triggers	65.98	61.53	72.59	64.86	69.09	63.13
with gold_triggers + w/o reward–penalty	67.50	55.83	72.10	63.89	68.53	62.35

Arg-C, argument classification; Arg-I, argument identification; EAE, event argument extraction.

with knowledge graphs. Proc 30th Italian Symp on Advanced Database Systems, p.505-513.

Chen YB, Xu LH, Liu K, et al., 2015. Event extraction via dynamic multi-pooling convolutional neural networks. Proc 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int Joint Conf on Natural Language Processing, p.167-176. https://doi.org/10.3115/V1/P15-1017

Cong X, Cui SY, Yu BW, et al., 2021. Few-shot event detection with prototypical amortized conditional random field. Proc Findings of the Association for Computational Linguistics, p.28-40. https://doi.org/10.18653/V1/2021.FINDINGS-ACL.3

Cui SY, Yu BW, Liu TW, et al., 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. Proc Findings of the Association for Computational Linguistics, p.2329-2339 https://doi.org/10.18653/v1/2020.findings-emnlp.211

Cui ZJ, Yuan ZM, Wu YF, et al., 2023. Intelligent recommendation for departments based on medical knowledge graph. *IEEE Access*, 11:25372-25385 https://doi.org/10.1109/ACCESS.2023.3254303

Devlin J, Chang MW, Lee K, et al., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.4171-4186., https://doi.org/10.18653/v1/n19-1423

Ding N, Li ZR, Liu ZY, et al., 2019. Event detection with trigger-aware lattice neural network. Proc Conf on Empirical Methods in Natural Language Processing and the 9th Int Joint Conf on Natural Language Processing, p.347-356., https://doi.org/10.18653/v1/D19-1033

Dozat T, Manning CD, 2017. Deep biaffine attention for neural dependency parsing. Proc 5th Int Conf on Learning Representations, p.1-8.

Du XY, Cardie C, 2020. Event extraction by answering (almost) natural questions. Proc Conf on Empirical Methods in Natural Language Processing, p.671-683., https://doi.org/10.18653/v1/2020.emnlp-main.49

Du XY, Ji H, 2022. Retrieval-augmented generative question answering for event argument extraction. Proc Conf on Empirical Methods in Natural Language Processing, p.4649-4666., https://doi.org/10.18653/v1/2022.emnlp-main.307

Dwivedi D, Yemula PK, Pal M, 2023. Dynamopmu: A physics informed anomaly detection, clustering, and prediction method using nonlinear dynamics on μ PMU measurements. *IEEE Trans on Instrum Meas*, 72:1-9 https://doi.org/10.1109/TIM.2023.3327481

Endsley MR, 1988. Design and evaluation for situation awareness enhancement. *Proc Hum Factors Ergon Soc Annu Meet*, 32(2):97-101 https://doi.org/10.1177/154193128803200221

Endsley MR, 2001. Designing for situation awareness in complex systems. Proc 2nd Int Workshop on Symbiosis of Humans, Artifacts and Environment, p.1-14.

Ettinger A, Hwang J, Pyatkin V, et al., 2023. âĂIJYou are an expert linguistic annotatorâĂİ: limits of LLMs as analyzers of abstract meaning representation. Proc Findings of the Association for Computational Linguistics, p.8250-8263. https://doi.org/10.18653/v1/2023.findings-emnlp.553

Feng XC, Qin B, Liu T, 2018. A language-independent neural network for event detection. *Sci China Inf Sci*, 61(9):092106 https://doi.org/10.1007/S11432-017-9359-X

Gururangan S, Marasović A, Swayamdipta S, et al., 2020. Don't stop pretraining: adapt language models to domains and tasks. Proc 58th Annual Meeting of the Association for Computational Linguistics, p.8342-8360., https://doi.org/10.18653/v1/2020.acl-main.740

Guzman-Nateras L, Dernoncourt F, Nguyen T, 2023. Hybrid knowledge transfer for improved cross-lingual event detection via hierarchical sample selection. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.5414-5427., https://doi.org/10.18653/v1/2023.acl-long.296

Han RD, Peng T, Yang CH, et al., 2023. Is information extraction solved by chatgpt? An analysis of performance, evaluation criteria, robustness and errors. https://doi.org/10.48550/arXiv.2305.14450

He YX, Hu JY, Tang BZ, 2023. Revisiting event argument extraction: can EAE models learn better when being aware of event co-occurrences? Proc 61st Annual Meeting of the Association for Computational Linguistics, p.12542-12556., https://doi.org/10.18653/v1/2023.acl-long.701

Hong Y, Zhang JF, Ma B, et al., 2011. Using cross-entity inference to improve event extraction. Proc 49th Annual Meeting of the Association for Computational Linguistics, p.1127-1136.

Hsu IH, Huang KH, Boschee E, et al., 2022. DEGREE: a data-efficient generation-based event extraction model. Proc Conf of the North American Chapter of the Association for Computational Linguistics, p.1890-1908., https://doi.org/10.18653/v1/2022.naacl-main.138

Hsu IH, Xie ZY, Huang KH, et al., 2023. AMPERE: AMR-aware prefix for generation-based event argument extraction model. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.10976-10993., https://doi.org/10.18653/v1/2023.acl-long.615

Ji H, Grishman R, 2008. Refining event extraction through cross-document inference. Proc ACL-08, p.254-262.

Lewis M, Liu YH, Goyal N, et al., 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Proc 58th Annual Meeting of the Association for Computational Linguistics, p.7871-7880. https://doi.org/10.18653/V1/2020.ACL-MAIN.703

Li S, Zhao RN, Li ML, et al., 2023. Open-domain hierarchical event schema induction by incremental prompting and verification. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.5677-5697., https://doi.org/10.18653/v1/2023.acl-long.312

Lin Y, Ji H, Huang F, et al., 2020. A joint neural model for information extraction with global features. Proc 58th Annual Meeting of the Association for Computational Linguistics, p.7999-8009., https://doi.org/10.18653/v1/2020.acl-main.713

Liu J, Chen YF, Xu JA, 2021. Machine reading comprehension as data augmentation: a case study on implicit event argument extraction. Proc Conf on Empirical Methods in Natural Language Processing, p.2716-2725., https://doi.org/10.18653/v1/2021.emnlp-main.214

Liu J, Chen YF, Xu JA, 2022. Saliency as evidence: event detection with trigger saliency attribution. Proc 60th Annual Meeting of the Association for Computational Linguistics, p.4573-4585., https://doi.org/10.18653/v1/2022.acl-long.313

Liu J, Sui DB, Liu K, et al., 2023a. Learning with partial annotations for event detection. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.508-523., https://doi.org/10.18653/v1/2023.acl-long.30

Liu J, Sui DB, Liu K, et al., 2023b. Learning with partial annotations for event detection. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.508-523, https://doi.org/10.18653/v1/2023.acl-long.30

Liu X, Luo ZC, Huang HY, 2018. Jointly multiple events extraction via attention-based graph information aggregation. Proc Conf on Empirical Methods in Natural Language Processing, p.1247-1256., https://doi.org/10.18653/v1/d18-1156

Liu X, Huang HY, Shi G, et al., 2022. Dynamic prefix-tuning for generative template-based event extraction. Proc 60th Annual Meeting of the Association for Computational Linguistics, p.5216-5228., https://doi.org/10.18653/v1/2022.acl-long.358

Loshchilov I, Hutter F, 2019. Decoupled weight decay regularization. Proc 7th Int Conf on Learning Representations, p.1-8.

Lou DF, Liao ZL, Deng SM, et al., 2021. MLBiNet: a cross-sentence collective event detection network. Proc 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int Joint Conf on Natural Language Processing, p.4829-4839., https://doi.org/10.18653/v1/2021.acl-long.373

Lu D, Ran SH, Tetreault JR, et al., 2023. Event extraction as question generation and answering. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.1666-1688. https://doi.org/10.18653/v1/2023.acl-short.143

Lu YJ, Lin HY, Xu J, et al., 2021. Text2event: controllable sequence-to-structure generation for end-to-end event extraction. Proc 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int Joint Conf on Natural Language Processing, p.2795-2806., https://doi.org/10.18653/v1/2021.acl-long.217

Ma MD, Taylor A, Wang W, et al., 2023. DICE: data-efficient clinical event extraction with generative models. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.15898-15917., https://doi.org/10.18653/v1/2023.acl-long.886

Matey AH, Danquah P, Koi-Akrofi GY, 2022. Predicting cyber-attack using cyber situational awareness: The case of independent power producers (IPPs). *Int J Adv Comput Sci Appl*, 13(1):700-709. https://doi.org/10.14569/IJACSA.2022.0130181

McClosky D, Surdeanu M, Manning CD, 2011. Event extraction as dependency parsing. Proc 49th Annual Meeting of the Association for Computational Linguistics:Human Language Technologies, p.1626-1635., https://doi.org/10.5555/2002472.2002667

Nam H, Kim SH, Park YH, 2022. Filteraugment: An acoustic environmental data augmentation method. Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, p.4308-4312., https://doi.org/10.1109/ICASSP43922.2022.9747680

Nguyen TH, Cho K, Grishman R, 2016. Joint event extraction via recurrent neural networks. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.300-309., https://doi.org/10.18653/v1/n16-1034

Onwubiko C, 2020. CyberOps: situational awareness in cybersecurity operations. *Int J Cyber Situational Aware*, 5(1):82-107. https://doi.org/10.22619/ijcsa.2020.100134

Palash M, Bhargava B, 2023. SAFER: situation aware facial emotion recognition https://doi.org/10.48550/ARXIV.2306.09372

Pang CX, Cao YX, Ding Q, et al., 2023. Guideline learning for in-context information extraction. Proc Conf on Empirical Methods in Natural Language Processing, p.15372-15389. https://doi.org/10.18653/v1/2023.emnlp-main.950

Ping Y, Lu JY, Gan RY, et al., 2023. UniEX: an effective and efficient framework for unified information extraction via a span-extractive perspective. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.16424-16440., https://doi.org/10.18653/v1/2023.acl-long.907

Sha L, Qian F, Chang BB, et al., 2018. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. Proc 32nd AAAI Conf on Artificial Intelligence, p.5916-5923., https://doi.org/10.1609/aaai.v32i1.12034

Shashikumar SP, Josef CS, Sharma A, et al., 2021. DeepAISE - an interpretable and recurrent neural survival model for early prediction of sepsis. *Artif Intell Med*, 113:102036 https://doi.org/10.1016/J.ARTMED.2021.102036

Shu XF, Yan J, Gao WR, et al., 2021. Research on military equipment entity recognition and knowledge graph construction method based on ALBERT-Bi-LSTM-CRF. Proc 4th Int Conf on Artificial Intelligence and Pattern Recognition, p.273-279.
https://doi.org/10.1145/3488933.3489030

Song ZY, Bies A, Strassel S, et al., 2015. From light to rich ERE: annotation of entities, relations, and events. Proc 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, p.89-98.
https://doi.org/10.3115/V1/W15-0812

Wadden D, Wennberg U, Luan Y, et al., 2019. Entity, relation, and event extraction with contextualized span representations. Proc Conf on Empirical Methods in Natural Language Processing and the 9th Int Joint Conf on Natural Language Processing, p.5783-5788.
https://doi.org/10.18653/V1/D19-1585

Wang B, Huang HY, Wei XC, et al., 2023a. Boosting event extraction with denoised structure-to-text augmentation. Proc Findings of the Association for Computational Linguistics, p.11267-11281.,
https://doi.org/10.18653/v1/2023.findings-acl.716

Wang SJ, Yu M, Huang LF, 2023b. The art of prompting: event detection based on type specific prompts. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.1286-1299.,
https://doi.org/10.18653/v1/2023.acl-short.111

Wang S, Yu M, Chang S, et al., 2022. Query and extract: refining event extraction as type-oriented binary decoding. Proc Findings of the Association for Computational Linguistics, p.169-182.,
https://doi.org/10.18653/v1/2022.findings-acl.16

Wang ZT, Wang XY, Hu W, 2023. Continual event extraction with semantic confusion rectification. Proc Conf on Empirical Methods in Natural Language Processing, p.11945-11955.
https://doi.org/10.18653/v1/2023.emnlp-main.732

Xia LQ, Liang YS, Leng JW, et al., 2023. Maintenance planning recommendation of complex industrial equipment based on knowledge graph and graph neural network. *Reliab Eng Syst Saf*, 232:109068
https://doi.org/10.1016/J.RESS.2022.109068

Xu RX, Wang PY, Liu TY, et al., 2022a. A two-stream AMR-enhanced model for document-level event argument extraction. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.5025-5036.,
https://doi.org/10.18653/v1/2022.naacl-main.370

Xu TY, Guo C, Du LX, et al., 2022b. A method for traditional Chinese medicine knowledge graph dynamic construction. Proc 5th Int Conf on Big Data Technologies, p.196-202.
https://doi.org/10.1145/3565291.3565323

Xu ZZ, Liu RK, Yang S, et al., 2023a. Learning imbalanced data with vision transformers. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.15793-15803.,
https://doi.org/10.1109/CVPR52729.2023.01516

Xu ZY, Lee JY, Huang LF, 2023b. Learning from a friend: improving event extraction via self-training with feedback from abstract meaning representation. Proc Findings of the Association for Computational Linguistics:, p.10421-10437.,
https://doi.org/10.18653/v1/2023.findings-acl.662

Yang P, Cong X, Sun ZY, et al., 2021. Enhanced language representation with label knowledge for span extraction. Proc Conf on Empirical Methods in Natural Language Processing, p.4623-4635.,
https://doi.org/10.18653/v1/2021.emnlp-main.379

Yang S, Feng DW, Qiao LB, et al., 2019. Exploring pre-trained language models for event extraction and generation. Proc 57th Annual Meeting of the Association for Computational Linguistics, p.5284-5294.,
https://doi.org/10.18653/v1/p19-1522

Yang XJ, Lu YJ, Petzold LR, 2023a. Few-shot document-level event argument extraction. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.8029-8046.,
https://doi.org/10.18653/v1/2023.acl-long.446

Yang YQ, Guo QP, Hu XK, et al., 2023b. An AMR-based link prediction approach for document-level event argument extraction. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.12876-12889.,
https://doi.org/10.18653/v1/2023.acl-long.720

Yao YZ, Mao SY, Zhang NY, et al., 2023. Schema-aware reference as prompt improves data-efficient knowledge graph construction. Proc 46th International ACM SIGIR Conf on Research and Development in Information Retrieval, p.911-921.,
https://doi.org/10.1145/3539618.3591763

You HL, Samuel D, Touileb S, et al., 2022. EventGraph: event extraction as semantic graph parsing. Proc 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text, p.7-15.,
https://doi.org/10.18653/v1/2022.case-1.2

You HL, Touileb S, Øvrelid L, 2023a. JSEEGraph: joint structured event extraction as graph parsing. Proc 12th Joint Conference on Lexical and Computational Semantics, p.115-127.,
https://doi.org/10.18653/v1/2023.starsem-1.11

You MS, Yin J, Wang H, et al., 2023b. A knowledge graph empowered online learning framework for access control decision-making. *World Wide Web*, 26(2):827-848.
https://doi.org/10.1007/S11280-022-01076-5

Zeng Y, Yang HH, Feng YS, et al., 2016. A convolution BiLSTM neural network model for Chinese event extraction. Proc 5th CCF Conf on Natural Language Processing and Chinese Computing, 10102:275-287.
https://doi.org/10.1007/978-3-319-50496-4_23

Zhang JR, Ilievski F, Ma KX, et al., 2023a. A study of situational reasoning for traffic understanding. Proc 29th ACM SIGKDD Conf on Knowledge Discovery and Data Mining, p.3262-3272.
https://doi.org/10.1145/3580305.3599246

Zhang KH, Shuang K, Yang XY, et al., 2023b. What is overlap knowledge in event argument extraction? APE: a cross-datasets transfer learning model for EAE. Proc 61st Annual Meeting of the Association for Computational Linguistics, p.393-409.,
https://doi.org/10.18653/v1/2023.acl-long.24

Zhao G, Gong XC, Yang XJ, et al., 2023. DemoSG: demonstration-enhanced schema-guided generation for low-resource event extraction. Proc Findings of the Association for Computational Linguistics, p.1805-1816. https://doi.org/10.18653/v1/2023.findings-emnlp.121

Zheng YZ, Pan SR, Lee VCS, et al., 2022. Rethinking and scaling up graph contrastive learning: an extremely efficient approach with group discrimination. Conf on Neural Information Processing Systems, p.1-17., http://papers.nips.cc/paper_files/paper/2022/hash/46027e3de0db3617a911f1a647def3bf-Abstract-Conference.html

List of supplementary materials

1 Case study

2 Discussions

Table S1 Case analysis of different types of semantic error classification for polysemous triggers