



Knowledge interaction graph guided prompting for event causality identification

Ruijuan Hu¹ · Jian Li¹ · Haiyan Liu¹ · Guilin Qi² · Yuxin Zhang²

Accepted: 20 November 2024 / Published online: 13 December 2024
© The Author(s) 2024

Abstract

Event causality identification (ECI) aims to identify causality between event pairs in a text, and is commonly approached as a supervised classification task using pre-trained language models (PLMs). However, limitations in implicit causality identification and insufficient event-knowledge interaction pose significant challenges to ECI. To address these issues, we propose a novel **Knowledge Interaction Graph guided Prompt Tuning (KIGP)**, which leverages prompt tuning and knowledge interaction to fully exploit the potential of PLMs for ECI by integrating external knowledge. Specifically, to accurately capture implicit causality, we design the guidance mechanism and construct event-knowledge interaction graphs that enable external knowledge to enhance event representations through deep interaction between events and knowledge. Experimental results on two benchmark datasets demonstrate that our model outperforms existing approaches significantly.

Keywords Event causality identification · Prompt tuning · Event knowledge · Interaction graph

1 Introduction

Event causality identification (ECI) is an immensely challenging task in natural language understanding that involves predicting the causality relation for a pair of events in a document. This technique has a wide range of applications in machine reading comprehension [1, 2], question-and-answer reasoning [3, 4], and event prediction [5, 6]. As shown in Fig. 1, an ECI model needs to identify the causalities between the pairs of events mentioned in sentences S1 and S2: ① **earthquake** $\xrightarrow{\text{cause}}$ **collapsed** in S1. It is an explicit causality that has explicit connectives with causal words like because, since, or therefore, which can be easily identified. ② **shot** $\xrightarrow{\text{cause}}$ **killed** in S2. It is an implicit causality that has ambiguous connectives without causal words, which requires a deep understanding of the context's semantics for identification.

The previous ECI approaches relied on feature-based techniques [7–9]. However, recent advancements have delved into deep learning techniques [10, 11] which have significantly improved performance. Nevertheless, current methods primarily adopt the “*pre-train, fine-tuning*” paradigm, as illustrated in Fig. 2. While PLMs such as BERT [12], Roberta [13] act as fill-in-the-blank, ECI requires a classification layer for fine-tuning following these pre-training stages. This results in a model unable to fully exploit the potential of PLMs due to an existing gap between downstream tasks and PLMs. Prompt tuning [14, 15] aims to address this gap by adjusting the downstream tasks, utilizing task-specific templates to align their pre-training approach with that of PLMs. Thus, prompt tuning allows us to leverage the prior knowledge gained from PLMs to enhance our ECI task's effectiveness.

Although prompt tuning accommodates PLMs, it nonetheless faces two key challenges: (1) **Difficult implicit causalities identification**. PLMs primarily train on large unlabeled and unstructured data rich in generic high-frequency entities, common sense [16–18]. Task-specific knowledge involving long-tail entities, multivariate associative relationships, and complex causal logic like event knowledge poses significant difficulty for PLMs to comprehend. As such, even implementing a basic prompt tuning paradigm proves challenging due to the lack of event-specific knowledge in PLMs. (2)

Ruijuan Hu and Jian Li contributed equally to this work.

✉ Ruijuan Hu
huruijuan01@126.com

- ¹ PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, Henan, China
- ² School of Computer Science and Engineering, Southeast University, Nanjing 211100, Jiangsu, China

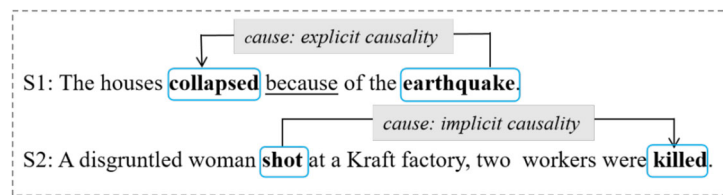


Fig. 1 Illustrations of original sentences with different causalities between event pairs

Insufficient event-knowledge interaction. Recent works have investigated external knowledge to enhance text understanding. Liu et al. [11] inserted external knowledge into the original text where the event was mentioned, potentially introducing noise and disrupting the semantics; Cao et al. [19] and Liu et al. [20] incorporated descriptive and relational knowledge by implementing a graph structure or knowledge representation learning, respectively, to facilitate causal reasoning. Nevertheless, external knowledge lacks sufficient interaction with the original text.

To tackle the challenges mentioned above, we propose a novel approach called **Knowledge Interaction Graph guided Prompt Tuning for Event Causality Identification (KIGP)**. (1) Implant external event knowledge to elicit PLMs. The event knowledge descriptions enhance the contextual understanding of events conceptually more straightforward and profound. In addition, it allows prompt tuning for better activation of PLMs' knowledge on event-event relationships, leading to more accurate identification of implicit causalities. (2) Construct interaction graphs to capture the interactions-between context, event mentions, and external knowledge. These graphs bridge the gap between external knowledge and causality by capturing potential semantic interaction.

Specifically, our approach comprises of three key steps: (1) Obtain the triples of event mentions from an external knowledge graph, i.e., ConceptNet, and linearize them into knowledge text. (2) Introduce an event pair-based template

and answer mapping verbalizer that uses prompt tuning to induce learning ability of PLMs and enhance implicit causality identification. (3) To facilitate better interaction between text, event mentions, and knowledge, we propose an interaction graph guidance mechanism by constructing interaction graphs that effectively guide the model for causality identification. We use graph convolutional network (GCN) to promote the feature representation of various nodes from a global perspective. Experimental results on two widely used datasets indicate that our model outperforms previous methods.

Our contributions are summarized as follows:

- (1) We propose a novel **Knowledge Interaction Graph guided Prompt Tuning for the Event Causality Identification** method, which effectively utilizes external event knowledge and prompt tuning to fully activate the potential of PLMs. To the best of our knowledge, this is the first work that combines GCN and prompt tuning for ECI task.
- (2) We design a guidance mechanism and construct knowledge interaction graphs that accurately guide external knowledge to effectively enrich the event representations through the deep interaction of text, events, and knowledge. These interaction graphs help to better capture implicit causality better and significantly improve our model's ability to solve the ECI task.

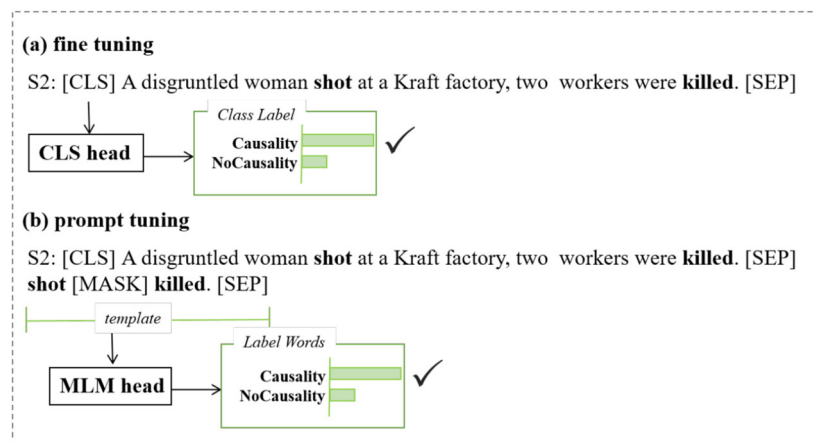


Fig. 2 Fine-tuning and prompt tuning for ECI

- (3) Experimental results demonstrate that our approach substantially outperforms the most recent state-of-the-art approach on two benchmark datasets, EventStoryLine and Causal-TimeBank, with an F1-score improvement of 6.3 and 2.9 percentage points, respectively.

2 Related work

Event causality identification The initial strategies for ECI involved feature-based methods that typically employed various features such as lexical and syntactic patterns [21–23], causality cues [7, 24, 25], and statistical information [6, 26]. Thereafter, supervised learning-based methods emerged, relying on a large amount of labeled data [27–30]. However, the scale of annotated datasets currently is relatively limited where so far the largest dataset EventStoryLine [31] only contains 258 documents, 4316 sentences, and 1770 causal event pairs. To address this problem, weakly supervised [8] and external knowledge-introducing approaches [11, 32–34] enhance the datasets and improve ECI performance. Advanced PLMs have achieved good performance in recent research [10]. Liu et al. [11] utilized a BERT-based model for mention masking generalization, and Zuo et al. [33] employed a pairwise learning framework to identify causalities by generating new samples. However, these existing methods are resolved by fine-tuning, which makes it difficult to identify implicit causalities.

Prompt tuning Since the emergence of GPT-3 [18], a new fine-tuning methodology named prompt tuning has gained attention. Unlike the “*pre-train, fine-tuning*” paradigm, prompt tuning adapts downstream tasks to PLMs and retrieves knowledge already stored in PLMs. It is widely applied to a large variety of tasks such as text classification [35], relationship extraction [36], event extraction [37, 38], and entity classification [39]. Researchers have made efforts to determine how to design prompt templates, and different methods have been proposed for automatic search of discrete prompts, gradient-guided search, and continuous prompts [40] including P-tuning [41] and Prefix-tuning [42] were successively proposed. Shen et al. [43] used a derivative prompt joint learning method to enhance the model’s ability to identify explicit and implicit causality. Recently, some studies have attempted to integrate external knowledge into prompt design. Tsinghua University proposed PTR [36], which implanted logic rules into prompt tuning, and KPT++ [44], which extended the verbalizer through external knowledge graphs, achieved large performance gains in task scenarios such as relationship extraction and text classification. Generally, external knowledge can be planted into input augmentation, architectural augmentation, and output regularity. Liu et al. [20] put forward a knowledge enhanced prompt tuning framework that exploited background knowl-

edge and relational information and adopted knowledge representation learning to capture implicit causalities further. However, previous works [45, 46] suggest that not all external knowledge will bring gains, and unselective implantation of external knowledge can sometimes introduce noise.

Graph convolutional network GCN [47] for graph structured data (non-Euclidean data) is widely used for node classification, graph classification, or link prediction. For example, textGCN treated documents and words as nodes, and exploited GCN to learn better node representations for text classification. RichGCN [48] first utilized an interaction graph for causality prediction, but it was prone to error accumulation due to the composition process using various existing NLP tools. ERGO [49] constructed event-relational graphs, where each node represented a pair of causal events, thereby converting ECI into a graph for node classification problems.

To efficiently select appropriate task-related knowledge and optimize learning knowledge representation, our model also introduces external knowledge, utilizes prompt tuning, and uses GCN to process the interaction graph. However, our approach differs from other related works in three ways: (1) To avoid error accumulation and considering that syntactic structures can be obtained directly through PLMs, we do not utilize off-the-shelf NLP tools to build the graph but design it based on the guidance mechanism. (2) Instead of straightly implementing node classification or relationship prediction of GCN, we take advantage of GCN’s powerful feature extraction in graph data to obtain the hidden layer features of nodes in the knowledge interaction graph; (3) we expect to adopt the feature representation that contains more in-depth interaction knowledge to precisely guide prompt and effectively stimulate the potential of PLMs. Note that, **we regard perceived event knowledge as a bridge between original texts and true causalities**. Simultaneously, we focus on addressing the issue of constructing interaction graphs by blending representations of integrated events and different knowledge.

3 Methodology

The overall framework of our approach KIGP is illustrated as Fig. 3. It contains three components: Document Encoder, Interaction Constructor, and Predictor. First, **Document Encoder** obtains words representations of the original text, event knowledge, and prompt for ECI. Second, the **Interaction Constructor** obtains the event representations aggregating knowledge through the graph structure to enhance the event representations in the prompt. The event knowledge interaction graph acquires representations ke_s , ke_t of event nodes by exploiting the knowledge encoder GCN, which aggregates the features of neighboring knowledge nodes in

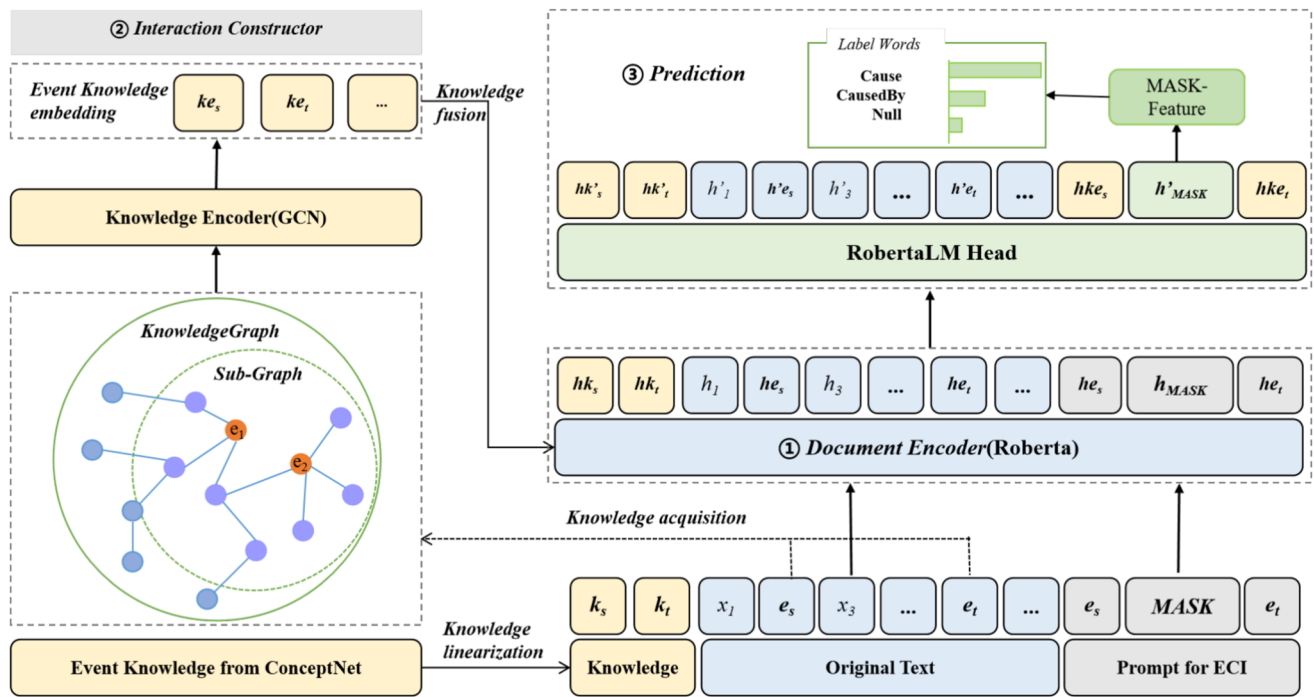


Fig. 3 Overall framework of our approach for ECI(KIGP). ① The input of Document Encoder contains three parts: the event knowledge text obtained from the external knowledge graph ConceptNet, the original text and prompt for ECI, and the output is the token representations corresponding to the three parts. ② The interaction constructor obtains the

event knowledge representations through the constructed event knowledge interaction graph by GCN encoding, and then fuses them with the event representations in the prompt. ③ The fused representations are sent to Roberta, and the causality is predicted according to the MASK-feature of RobertaLM head

the graph. Fuse them with the event representations he_s and he_t in the prompt to obtain new representations hke_s and hke_t that contain event semantic and relational knowledge. Finally, feeding the fused representations into Roberta and combining with the prompt, the causality classification results are predicted by MASK-Feature in **Predictor** based on RobertaLM head's vocabulary probability distribution.

3.1 Question definition

We convert the ECI task into a classification problem using the masked language model (MLM) head to make predictions. In contrast to most previous binary classification problems (Causality, NoCausality), we adopt ternary classification and further refine Causality into Cause and CausedBy. Given a sentence $X = \{x_1, x_2, \dots, x_l\}$ and an event pair $\langle e_s, e_t \rangle$ in S , where l is the number of tokens. \mathcal{Y} is the set of causal labels denoting whether there is a causality between event pairs. We set $\mathcal{Y} = \{Cause, CausedBy, Null\}$, which respectively indicate that e_s causes e_t , e_s is caused by e_t and there is no causality between $\langle e_s, e_t \rangle$. The goal of KIGP is to predict the causal label $y \in \mathcal{Y}$.

We design ECI Template as $\mathcal{T}_{ECI}(X)$ and splice it after the sentence X to make X' as the input of MLM. Then induce the model to generate label words associated with given labels. Specifically, splice $[CLS]$ and $[SEP]$ to the beginning and

end of X respectively, and add $[SEP]$ to the end of $\mathcal{T}_{ECI}(X)$. X' containing one $[MASK]$ token in $\mathcal{T}_{ECI}(X)$ that needs MLM to predict label words is:

$$X' = [CLS]X[SEP]\mathcal{T}_{ECI}(X)[SEP] \quad (1)$$

When X' is fed into MLM, the model can obtain the probability distribution $p([MASK]|X')$ of the candidate class, as:

$$p(y|X') = p([MASK] = m|X') \quad (2)$$

where m represents the m^{th} label token of class y .

3.2 Document encoder

We choose an prominent MLM named **Roberta** [13] as the document encoder to encode the input sequence and output prediction results. Encode each word in the input sequence X' which includes Original Text, Event Knowledge, and ECI prompt template $\mathcal{T}_{ECI}(X)$ as a sequence of representations. The encoded result sequence is $H = [h_{CLS}, H_X, h_{SEP}, H_{prompt}, h_{SEP}]$, where $H_X = [h_1, h_2, \dots, h_n]$ and $H_{prompt} = [h_{e_s}, h_{MASK}, h_{e_t}]$. This module involves event knowledge acquisition and prompt template design.

3.2.1 Event knowledge acquisition and linearization

A knowledge graph involving a large amount of common sense, entity knowledge, and semantic relations is undoubtedly the best choice for external knowledge. ConceptNet [50] is a knowledge graph rich in concepts and semantic relations, with more than 8 million nodes, 21 million edges, and 34 core relations. For ECI task, we require in-depth knowledge of event descriptions to supplement or activate the potential of PLMs, as well as to give a better prompt for prompt learning. Therefore, we retrieve the definitions of events mentioned in the original text along with the 16 semantic relations pertinent to ECI in ConceptNet: *CapableOf*, *Causes*, *CauseDesire*, *UsedFor*, *HasA*, *PartOf*, *Entails*, *Desires*, *HasContext*, *HasSubevent*, *HasPrerequisite*, *ReceivesAction*, *IsA*, *HasProperty*, *MannerOf*, and *CreatedBy*. Other knowledge sources such as WordNet can also be used as external knowledge.

Specifically, we first retrieve the nodes of event mentions e_s , e_t in the original text from Knowledge Graph (ConceptNet). Noting that most of the event mentions words involve the plural, past tense or participle forms of words, we perform word form reduction on them. Then match the Sub-Graph of the 16 semantic relations and associated nodes related to the event mentions. Part of the knowledge related to “shot” and “killed” gleaned from ConceptNet is shown in Fig. 4, **shooting** \xrightarrow{IsA} **homicide**, **kill** \xrightarrow{Causes} **death**, etc. We observe that an event mentioned corresponds to various relational knowledge and that there may also be several explanatory

terms within each relationship. As an illustration, the “HasSubevent” relation for the event mention “kill” includes the explanation items “HasSubevent shoot”, “HasSubEvent someone or something dies”, etc. In order to enhance the event representation, we add each explanation item connected to each event mention to a knowledge list, which is a more thorough and in-depth description of the event. These triples are finally linearized into text using *EventText*. The semantic relation words “IsA”, “HasSubevent”, etc., are changed into plain language descriptions such as “is a” and “has subevent”, etc., to make the knowledge description more natural and fluid. *EventText* is implanted into the input sequence in a spliced manner.

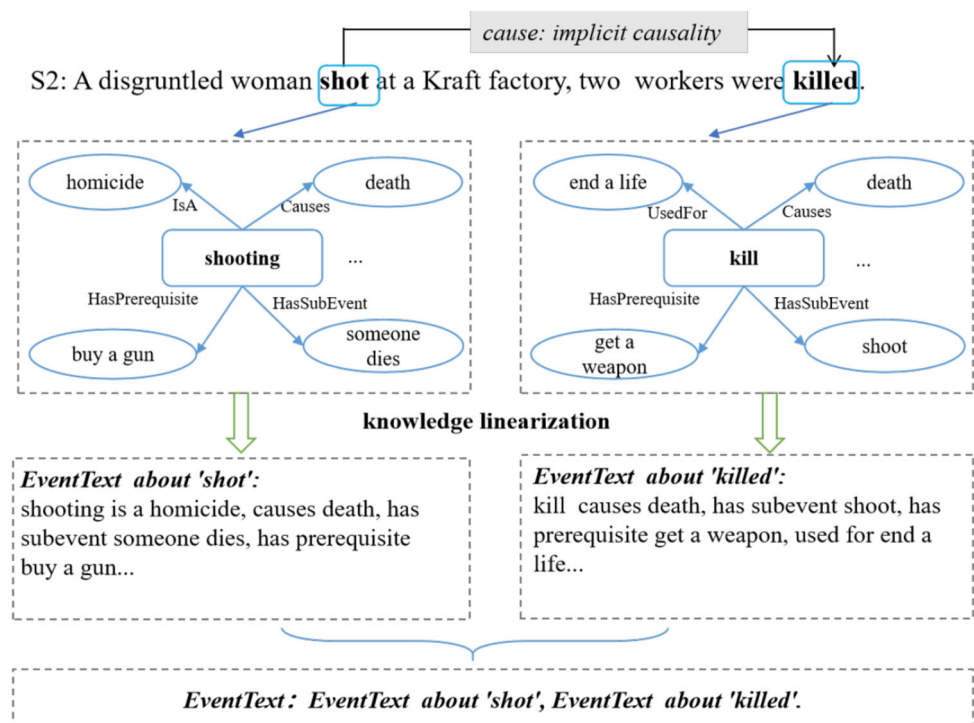
3.2.2 Design for ECI prompt

Prompt tuning converts downstream tasks into a form consistent with the pre-trained target by introducing task-specific templates, where it is critical to know how to construct templates and verbalizers. We design a prompt for ECI as $\mathcal{T}_{ECI}(X)$.

$$\mathcal{T}_{ECI}(X) : \text{In this sentence,} \\ \langle t1 \rangle \langle 'e'_s \rangle \langle t2 \rangle \langle t5 \rangle [\text{MASK}] \langle t6 \rangle \langle t3 \rangle \langle 'e'_t \rangle \langle t4 \rangle. \quad (3)$$

Some learnable tokens are applied to the template to dynamically accommodate the training of PLMs (e.g. $\langle t1 \rangle$ and $\langle t2 \rangle$ denote the position of $'e'_s$, $\langle t3 \rangle$ and $\langle t4 \rangle$

Fig. 4 Event knowledge from ConceptNet related to event mentions



denote the position of ' e_t '). The $[MASK]$ token in $\mathcal{T}_{ECI}(X)$ requires the label words to be filled ($\langle t5 \rangle$ and $\langle t6 \rangle$ denote the position of $[MASK]$).

For ECI task, the label words V are in PLM vocabulary. However, due to the large space of PLM vocabulary, it is possible that some words do not reflect the causality well, so we follow the previous work with setting virtual words for causal verbalizer. The label words V are denoted by the three virtual words $\{Cause, CausedBy, Null\}$. These virtual words are also learnable tokens, $Cause$ and $CausedBy$ facilitate the model to learn the direct features of causality, and the verbalizer directly uses these three label words corresponding to the causal labels. The probability distribution of the label words in the $[MASK]$ position of MLM is used as the probability distribution of the causal labels.

3.3 Interaction constructor

The representation of each word is obtained from the document encoder, and the classification results can be straightforwardly acquired after model training. Considering that there are close correlations with text, events, and knowledge, these associations can focus on event semantics and concept knowledge, which provide richer semantic features for causality identification. Therefore, we propose an interaction guidance mechanism and design an interaction constructor. Based on the guidance mechanism, the interaction constructor can effectively guide external knowledge to enhance the representation of relevant nodes. By constructing interaction graphs with texts, events, and knowledge, the hidden interaction representation of each node is generated by GCN, which has powerful feature extraction capabilities.

3.3.1 Guidance mechanism

There are two types of guidance mechanisms: guiding original text (*got*) and guiding events in the prompt (*get*). As shown in Fig. 5, the *got* mechanism intends to use external event description knowledge to enhance the semantic comprehension of the original text, so it bridges external knowledge

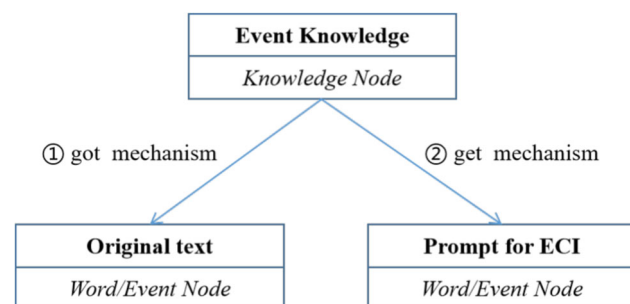


Fig. 5 Guide mechanism for interaction graph

with textual event mentions; while the *get* mechanism seeks to reinforce the reasoning ability of event relations in the prompt template, so it bridges external knowledge with event mentions in the prompt template. An example of guidance mechanisms can be found in Fig. 6 in Section 3.3.2. The blue arrow represents the *got* mechanism, which guides the establishment of connections between knowledge nodes and event nodes in the original text. For example, knowledge nodes k_s are connected to “shot” and k_t are connected to “killed”. The red arrow indicates the *get* mechanism, which guides the establishment of connections between knowledge nodes and event nodes in the prompt. For example, knowledge nodes k_s are connected to e_s , and k_t are connected to e_t .

3.3.2 Interaction graph construction

Constructing the nodes and edges in the interaction graph based on the guidance mechanism is essential for learning effective event representations for ECI.

Nodes in interaction graph Given a document $D = \{w_1, w_2, \dots, w_i\}$ (where w_i is a word in the document), we construct one graph for each document separately. The nodes in the graph should be able to capture the document content relevant to the source event e_s and the target event e_t to predict causality. We consider three node types in our work.

- ① Word Nodes, i.e., the contextual words of the document D .
- ② Event Nodes, i.e., the event mentions in the document D or the ECI prompt $\mathcal{T}_{ECI}(X)$, noted as $E = \{e_1, e_2, \dots, e_l\}$, where l is the number of knowledge.
- ③ Knowledge Nodes, i.e., external knowledge related to the event mentions, noted as $K = \{k_1, k_2, \dots, k_m\}$, where m is the number of knowledge. Thus, the set of nodes is $N = \{D \cup E \cup K\} = \{x_1, x_2, \dots, x_n\}$, where n is the number of nodes ($n = i + l + m$).

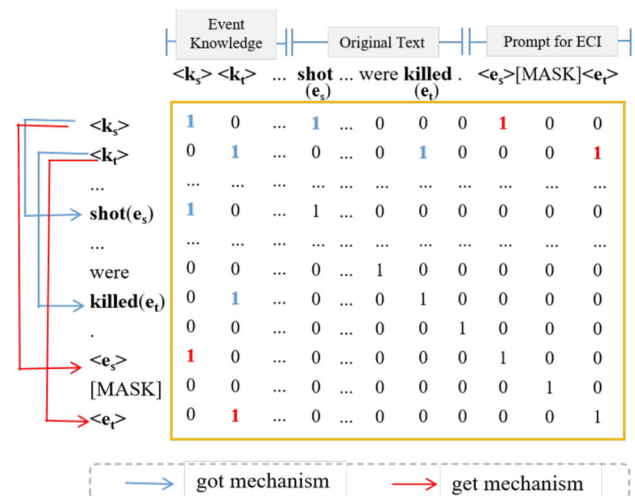


Fig. 6 Example of event knowledge interaction graph adjacency matrix

Edges in interaction graph After mapping the document into the three types of nodes, we construct the following two types of edges between nodes to establish the interaction graph based on the guidance mechanism.

① Event-Event Edge (E-E). The event pairs in a document will be scattered in different sentences. Our main purpose is to identify the causality between two events, therefore, Event-Event information is extremely valuable. We add edges between events e_s and e_t in a document.

② Event-Knowledge Edge (E-K). In order to complement the conceptual and semantic knowledge of events in a document, we construct edges between event nodes and external knowledge nodes.

Interaction graph feature extraction After defining the nodes and edges of the knowledge interaction graph, the adjacency matrix A is used to automatically construct the event knowledge interaction graph G , with the number of nodes n . Then A is the matrix of $n \times n$. According to the got mechanism, connect the knowledge nodes to the corresponding original event nodes, and set the corresponding position to 1. According to the get mechanism, connect the knowledge nodes with the event nodes in the prompt, with the corresponding position set to 1 and all other positions set to 0.

$$A_{ij} = \begin{cases} 1, & e_{ij} \in E - E, E - K \\ 0, & e_{ij} \notin E - E, E - K \end{cases} \quad (4)$$

$A_{ij} = 1$ indicates that node i and node j is connected to each other with an edge. We employ GCN for feature extraction to generate node representations in the interaction graph. Specifically, in our work, the GCN model uses the feature representation obtained by the document encoder $H^{(0)} = [h_{CLS}, H_K^{(0)}, h_{SEP}, H_D^{(0)}, h_{SEP}, H_{Prompt}^{(0)}, h_{SEP}]$ as the initial representation, where $H_K^{(0)} = [hk_s, hk_t]$, $H_D^{(0)} = [h_1, h_{e_s}, h_3, \dots, h_{e_t}, \dots, h_i]$ and $H_{Prompt}^{(0)} = [he_s, h_{MASK}, he_t]$. After l layers aggregation, the feature representation $H^{(l+1)}$ of the $(l+1)^{th}$ layer is:

$$H^{(l+1)} = ReLU(AH^{(l)}W^{(l)}) \quad (5)$$

Where $H^{(l)}$ and $H^{(l+1)}$ denote the feature vectors of nodes in the l^{th} , $(l+1)^{th}$ layer, respectively. $W^{(l)}$ denotes the weight matrix of the l^{th} layer, $ReLU$ is the activation function, and after G -layer GCN, $H^{(g)}$ is noted as $H^{(g)} = GCN(A, H^{(0)}, G)$ for convenience. The GCN model outputs the feature vectors of event nodes e_s and e_t as ke_s and ke_t , where ke_s and ke_t aggregates the features of neighboring knowledge nodes from the structure. The new feature vectors hke_s and hke_t are derived by fusing them with he_s and he_t in the prompt through splicing to enhance the semantic representation of events. The final fused feature representation $H^{(g)}$ captures the relationship

between word nodes and their neighboring nodes, denoted as: $H^{(g)} = [h'_{CLS}, H_K^{(g)}, h'_{SEP}, H_D^{(g)}, h'_{SEP}, H_{Prompt}^{(g)}, h'_{SEP}]$, where $H_K^{(g)} = [hk'_s, hk'_t]$, $H_D^{(g)} = [h'_1, h'_{e_s}, h'_3, \dots, h'_{e_t}, \dots, h'_i]$ and $H_{Prompt}^{(g)} = [h'_{e_s}, h'_{MASK}, h'_{e_t}]$. Therefore, it realizes the interaction of events and knowledge, and provides more extensive and more abstract deep features for causality prediction.

Assuming the input document is S2 as above, the input format is:

*[CLS] <k_s> shooting is a homicide, causes death...</k_s>. <k_t> kill causes death, has subevent shoot...</k_t>. [SEP] A disgruntled woman **shot** at a Kraft factory, two workers were **killed**. [SEP] In this sentence, shot [MASK] killed. [SEP]*

The two events “shot” and “killed” in the above text are denoted by e_s and e_t , respectively. The knowledge corresponding to the two events “shooting causes death...” and “kill has subevent shoot...” is denoted by k_s and k_t . The adjacency matrix of the event knowledge interaction graph constructed in the training process is shown in Fig. 6, where the word itself is 1 on the diagonal line, and k_s and k_t interact with e_s and e_t in the original text and prompt for ECI, respectively.

3.4 Predictor

The representation $H^{(g)}$ obtained by the GCN module of the interaction constructor has the intensive interaction feature, enhancing the event representations in the prompt. Then $H^{(g)}$ is further fed into RobertaLM Head to yield the MASK-Feature. The predictor will get the probability distribution of the candidate classes based on the MASK-Feature, and eventually, the causality labels $\mathcal{Y} = \{Cause, CausedBy, Null\}$ of the event pair $\langle e_s, e_t \rangle$ can be predicted.

4 Experiments

Our experiment aim to verify (1) whether external event knowledge can effectively potentiate the ability of PLMs to identify the implicit causality, and (2) whether knowledge interaction graphs can precisely guide models to enhance the performance of ECI.

4.1 Datasets and evaluation metrics

Our proposed method is evaluated on two widely used datasets, EventStoryLine (version 0.9) [31] and Causal-TimeBank [51]. EventStoryLine contains 258 documents, 5334 events, and 1770 causal event pairs. As the prior split in [9, 20], we group documents by topic and sort them by topic IDs. The last two topics are used as the development data, and documents in the remaining 20 topics are employed for 5-fold cross-validation. Causal-TimeBank contains 184

documents, 1813 events, and 318 causal event pairs. Following Zuo et al. [33, 34], we adopt the same data division as they did, using 10-fold cross-validation. For the evaluation, we used Precision (P), Recall (R), and F1-score (F1) as evaluation metrics.

4.2 Experimental settings

In implementations, we use the pre-trained Roberta-base model of 768 dimensions for word embeddings as a document encoder. We set $1e-4$ for the learning rate of the Adam optimizer. Due to the sparsity of the positive example samples in the ECI datasets, the model training process uses negative sampling. We adopt a negative sampling rate of 0.5 for training our model, and the batch size for training is 16. We tune the hyper-parameters by grid search based on the development set performance and perform early stopping. In the interaction graph construction module, we use one layer for the GCN model ($G = 1$) and 2000 hidden units for GCN layers. The external knowledge acquisition is selected from the common sense knowledge graph ConceptNet 5.5.

4.3 BaseLines

We compare our model with the state-of-the-art (SOTA) models for ECI and we consider the following baselines:

Previous SOTA methods: **LSTM** [52] and **Seq** [53], a dependency path-based sequential model which is originally developed for temporal relation prediction; **LR+** [9] and **LIP** [9], document structure-based models; **RB** [51], a rule-based system; and **ML** [54], a feature-based model for ECI.

Using PLMs and introducing external knowledge methods: **LearnDA** [33], a data augmentation method by introducing knowledge bases to augment training data; **CauseRL** [32], a self-supervised learning method that learns context-specific causal patterns from external causal statements; **MM** [11], a BERT-based model with mention masking generalization. **KEPT** [20], a knowledge enhanced prompt tuning method incorporating background information and relational information.

Employing GCN methods: **RichGCN** [48], using GCN to capture interconnections in the document structure graph; **ERGO** [49], constructing an event relationship graph and utilizing GCN for node classification.

4.4 Main results

Tables 1 and 2 show the performances of our proposed approach and all benchmark models on EventStoryLine and Causal-TimeBank datasets, respectively.

- (1) In terms of overall performance, the proposed model KIGP outperforms all the existing baselines on both EventStoryLine and Causal-Timebank datasets, with

Table 1 Main results on EventStoryLine dataset (%)

Methods	P	R	F1
LSTM [52]	34.0	41.5	37.4
Seq [53]	32.7	44.9	37.8
LR+ [9]	37.0	45.2	40.7
LIP [9]	37.4	55.8	44.7
LearnDA [33]	42.2	69.8	52.6
CauSeRL [32]	41.9	69.0	52.1
MM [11]	41.9	62.5	50.1
KEPT [20]	50.0	68.8	57.9
RichGCN [48]	49.2	63.0	55.2
ERGO [49]	49.7	72.6	59.0
KIGP(ours)	62.4	68.5	65.3*

Bold denotes best F1. * denotes a significance test with $p=0.05$

6.3% and 2.9% improvement over the SOTA method ERGO, respectively. This indicates the effectiveness of our method for ECI task.

- (2) From the perspective of external knowledge and pre-training, LearnDA and CauSeRL show that introducing external knowledge can improve the prediction performance of causality compared to the approaches without external knowledge (RB, ML), but there is still a semantic gap between external knowledge and causality. The pre-training model MM is dedicated to stimulating the knowledge of PLMs themselves, and its performance is not as good as that of adding external knowledge, probably because PLMs do not have enough event-specific and causality knowledge for learning. KEPT capitalizes on background knowledge and relational information, and optimizes event representations and causality jointly with TransE to capture implicit relationships, which outperforms LearnDA and CauSeRL.
- (3) Our model adopts the “*PLMs, event knowledge, prompting*” paradigm to supplement PLMs with event-specific knowledge and use prompting to explore the potential

Table 2 Main results on Causal-TimeBank dataset (%)

Methods	P	R	F1
RB [51]	36.8	12.3	18.4
ML [54]	67.3	22.6	33.9
LearnDA [33]	41.9	68.0	51.9
CauSeRL [32]	43.6	68.1	53.2
MM [11]	36.6	55.6	44.1
KEPT [20]	48.2	60.0	53.5
RichGCN [48]	39.7	56.5	46.7
ERGO [49]	58.4	60.5	59.4
KIGP(ours)	61.3	63.4	62.3*

Bold denotes best F1. * denotes a significance test with $p=0.05$

semantics. The performance is improved by approximately 8% compared with CauSeRL and KEPT on both datasets. This demonstrates that external event-specific knowledge can effectively stimulate the ability of PLMs to recognize implicit causality.

- (4) From the perspective of interaction graph structure, compared with RichGCN and ERGO models using graph structure, we do not employ GCN to do node classification and relationship prediction, but directly adopt it to extract node features of event knowledge interaction graph. The F1-scores of our model are higher than RichGCN and ERGO on both datasets. The reason may be that our process of building event knowledge interaction graphs avoids introducing noise and causing error accumulation with the existing NLP tools. In addition, the powerful feature extraction capability of GCN can promote the hidden layer representation of nodes and precisely guide the model to understand the semantics to help causality prediction.

4.5 Ablation experiments

To analyze how each component in the proposed KIGP model contributes to the performance, we conduct ablation studies to turn off one at a time on the validation set as Tables 3 and 4 show.

- (1) **w/o intergcen**, to verify the effectiveness of the interaction graph module, we remove the interaction graph and use only Roberta encoder to generate hidden layer representations H instead of representations $H^{(g)}$ that go through the GCN layer for predicting causality. Without the interaction between text, events, and knowledge for bootstrapping, the performance decreased by 2.1% and 1.8% on the two datasets, respectively. This shows the importance of the event-knowledge interaction, where the features after the interaction play an crucial role in guiding causality reasoning. Our model can make in-depth interaction between event and knowledge, thus boosting the model performance.
- (2) **w/o eventkg**, to verify the effectiveness of external event knowledge, we remove the event knowledge text

Table 3 Performance of KIGP model with different components on EventStoryLine dataset(%)

Methods	P	R	F1	Δ
KIGP(ours)	62.4	68.5	65.3	–
w/o intergcen	61.2	65.4	63.2	–2.1
w/o eventkg	60.7	64.2	62.4	–2.9
w/o prmauto	61.8	66.5	64.1	–1.2
w/o prmec	59.7	63.8	61.7	–3.6

Δ means the points lower than KIGP

Table 4 Performance of KIGP model with different components on Causal-TimeBank dataset(%)

Methods	P	R	F1	Δ
KIGP(ours)	61.3	63.4	62.3	–
w/o intergcen	58.4	62.8	60.5	–1.8
w/o eventkg	57.6	62.4	59.9	–2.4
w/o prmauto	60.1	62.3	61.2	–1.1
w/o prmec	56.5	61.9	59.1	–3.2

Δ indicates the percentage points lower than KIGP

EventText which is acquired from ConceptNet from the input of document encoder, and simultaneously, the interaction graph module with GCN loses its function. As a result, the performance of our model drops 2.9% and 2.4% in terms of F1-score on the two datasets, respectively. The result indicates that external event-specific knowledge contains useful clues between events that facilitates the ability of PLMs to understand the semantics of text regarding event relationships.

- (3) **w/o prmauto**, to verify the validity of the automatic prompt, we remove the learnable tokens $\langle t_1 \rangle \langle t_2 \rangle \dots \langle t_6 \rangle$ in the template and use only “manual” prompt like $\mathcal{T}_{ECI}(X) : \text{In this sentence, } 'e_s' [MASK] 'e_t'$. The experimental results show that the performance of the “manual” prompt is 1.2% and 1.1% lower than that of the “manual+automatic” prompt using learnable tokens on the two datasets, which suggests that the learnable tokens indeed learn some contextual semantic information through the model training, which is helpful for causality prediction.
- (4) **w/o prmec**, to demonstrate the necessity of the prompt template module, the prompt $\mathcal{T}_{ECI}(X)$ is removed and ECI degenerates into a basic fine-tuning paradigm. We only serve the original text and event knowledge as Roberta’s input, resulting in a significant drop in performance (3.6% and 3.2%). This illustrates that the $[MASK]$ form of the prompt better caters to MLM’s cloze task and stimulates its learning ability. The precise prompt for ECI promotes a more accurate understanding and prediction of causalities.

Through ablation experiments, we observe that all the components contribute to model performance and both external knowledge interaction graph with GCN are beneficial and functional for ECI.

4.6 Impact of knowledge number and position

The number of knowledge It is observed that the number of each event knowledge triple obtained from ConceptNet varies within the range of 0 and 20. We make statistics on the number of relevant event knowledge in EventStoryLine

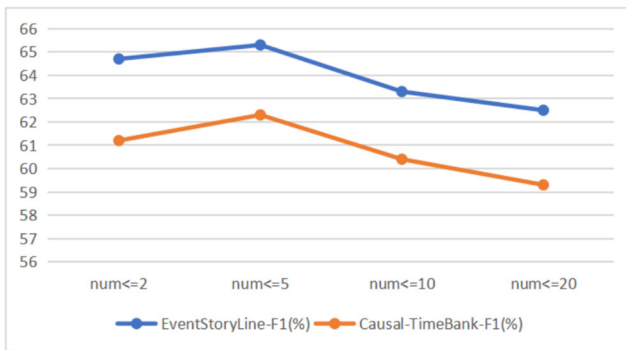


Fig. 7 Comparison of model performance corresponding to different number ranges in EventStoryLine and Causal-TimeBank datasets

and Causal-TimeBank datasets, and find that most of the event knowledge is within 5 items. Experiments with different numbers (2, 5, 10, and unrestricted) of event knowledge and the results are shown in Fig. 6. It can be found that the model performance does not keep improving with the increase of the number of knowledge, but the best performance is captured by limiting the number of knowledge to less than 5. More than 6 or unrestricted knowledge may generate knowledge noise, confuse the semantics and affect the understanding of the original text by PLMs (Fig. 7).

Knowledge positions Three forms of knowledge-enhanced event text as input to the document encoder are validated:

preposition, postposition, and interpolation. Preposition is to place the linearized knowledge *EventText* in front of *Original Text*, denoted as:

$$X = [EventText, Original Text] \quad (6)$$

Postposition is to place the linearized knowledge *EventText* behind *Original Text*, denoted as:

$$X = [Original Text, EventText] \quad (7)$$

Interpolation is to insert the linearized knowledge *EventText* directly into the position where the event is mentioned in the original text and to insert the relevant knowledge k_1, k_2 corresponded to the event mentioned in the text directly behind e_1, e_2 . An experimental comparison of the three forms of knowledge positions, as shown in Fig. 8, reveals that the accuracy rate of knowledge preposition is higher than that of knowledge postposition, and knowledge interpolation is the least effective. Intuitively, although the interpolation form can help the model improve the understanding of the event itself, it will make the gap between two events mentioned in the text too wide, reducing the flow of the original text and making it difficult for the model to determine the causality of the events.

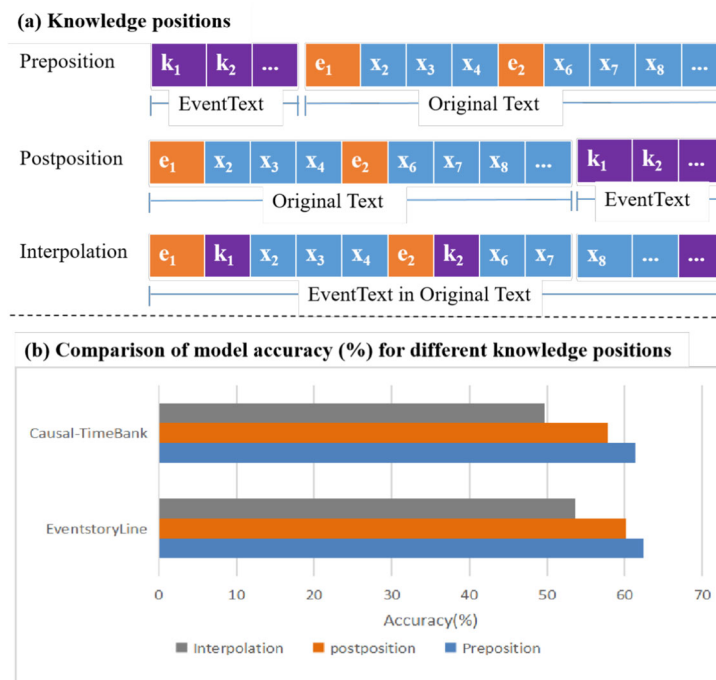


Fig. 8 (a) Three forms of knowledge position, x_i indicating words in the original text (blue), e_i indicating event mentions (orange), and k_i indicating event knowledge (purple); (b) Comparison of model accuracy (%) for different knowledge

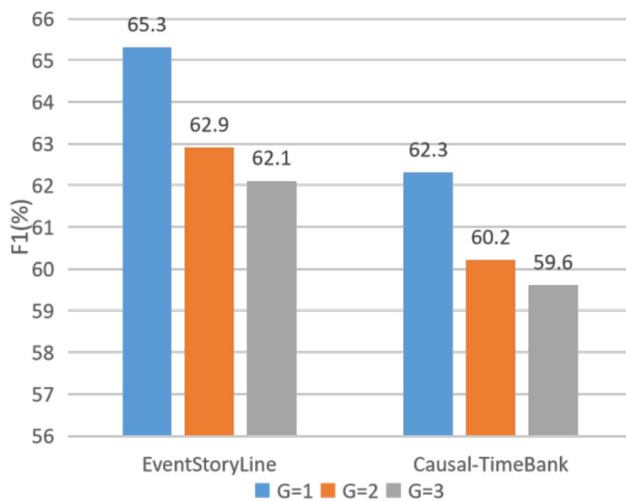


Fig. 9 Variation of model's F1-score corresponding to different GCN's layers of the interaction graph

4.7 Effect of structure and GCN layers of the interaction graph

The structure of the interaction graph constructed under the guidance of got and get mechanisms will change due to changes in event knowledge, original text, and prompt templates. With the original text and event knowledge unchanged, the structure of interaction graphs is different when “manual” prompt and “manual+automatic” prompt with learnable tokens are used. We compare the effects of the two in the ablation experiment, as shown in the row of w/o prmauto in Tables 3 and 4. The “manual+automatic” prompt with learnable tokens show better results than manual prompt, and also demonstrate the rationality of the interaction graph structure.

GCN is employed by the interaction graph module as feature extractor. Two layers of GCN are typically used in the text classification task to collect neighbor node features and achieve high performance. We experiment with the choice of GCN layers (G=1,2,3) for ECI task, and the model's F1 score on both datasets as shown in Fig. 9. It demonstrates that 1 layer is preferred over 2 and 3 layers. That is to say, the more GCN layers there are, the worse the effect. The interaction graph specifies text, event, and knowledge nodes with the intention of applying knowledge to improve the comprehension of occurrences, which may be the cause. knowledge nodes serve as the immediate neighbors of event nodes in the interaction graph, allowing knowledge features to be directly

aggregated by a single layer. If 2 or 3 layers are used, the range of aggregated nodes will further expand, which can easily confuse semantics and hinder the understanding of events.

4.8 The computation complexity involved in the process

Computational complexity is described from two aspects, one is spatial complexity, which refers to the storage space occupied by model training or prediction, mainly referring to the computer's graphics memory here. The second is time complexity, which refers to the time taken for model training or prediction, by counting the average time to run multiple Batches in each epoch. Since deep learning models are difficult to use traditional complexity representations, such as $O(n \cdot \log n)$, etc., we use a comparison-based complexity analysis method.

Our model is a “base+module” structure, the base model is Roberta, and the module is mainly GCN. Therefore, our model Robert+GCN is compared with the base model Roberta and analyzed in terms of time and spatial consumption. A comparison of the graphics memory consumption and time consumption for training and prediction of the two models on the EventStoryLine dataset is shown in Table 5. For convenience, maximum memory consumption for model training is denoted as Maxmt and average time for Batches spent on model training is denoted as Avgmt. Maximum memory consumption for model prediction is denoted as Maxmp and average time for Batches spent on model prediction is denoted as Avgmp.

It can be seen that with the Roberta+GCN model, there is no significant increase in the spatial complexity and time complexity compared to the Roberta model, and even the maximum memory consumption of Roberta+GCN is a little bit less than that of Roberta's when the model is predicted.

4.9 Case study

To visually demonstrate the effectiveness of KIGP, we do a case study to compare the identification results of both KIGP and RichGCN methods, as shown in Fig. 10.

In Case 1, RichGCN identifies *< war, bombs >* as a causal pair, but in fact, there is no causality between war and bombs, and the model may have misunderstood the words and confuses “war” with “warns” which are close to each

Table 5 A comparison of the graphics memory consumption and time consumption for training and prediction on the EventStoryLine dataset

Models	Maxmt(MB)	Avgmt(ms)	Maxmp(MB)	Avgmp(ms)
Robert+GCN	11052	205	2379	61
Robert	9245	204	2400	51

	<i>Examples</i>	$\langle e_i, e_j \rangle$	<i>GT</i>	<i>Rich</i>	<i>KIGP</i>
1	South Sudan warns of <u>war</u> after Sudan bombs refugee camp. Military aircraft from Sudan crossed the new international border with South Sudan and dropped <u>bombs</u> Thursday in and around a camp filled with refugees, officials said. A government official initially reported <u>deaths</u> , but an American activist who spoke to aid workers at the camp later said there were no casualties.	$\langle \text{war, bombs} \rangle$	✗	✓	✗
		$\langle \text{bombs, deaths} \rangle$	✓	✗	✓
2	Updated: July 02, 2013 15:50 IST A 6.1-magnitude <u>earthquake</u> which hit the Indonesian province of Aceh on Tues-3488 day <u>killed</u> a child, <u>injured</u> dozens and <u>destroyed</u> buildings, sparking panic in a region devastated by the quake-triggered tsunami of 2004.	$\langle \text{earthquake, injured} \rangle$	✓	✓	✓
		$\langle \text{earthquake, killed} \rangle$	✓	✓	✓
		$\langle \text{earthquake, destroyed} \rangle$	✓	✗	✓

Fig. 10 Case study. $\langle e_i, e_j \rangle$ indicates event pairs, *GT* means Ground Truth demonstrating the true relations between event pairs. *Rich* indicates the RichGCN method. Bold underlined words indicate events, ✓

indicates that the model identifies a causality between event pairs, and ✗ indicates that the model does not identify causality between event pairs

other. Because there is no explicit clue word in the text, RichGCN fails to identify the causality between “bombs” and “death”, an implicit causality that often requires common sense knowledge to be correctly inferred.

In Case 2, both RichGCN and KIGP can correctly determine that “earthquake cause injured” and “earthquake cause killed”, but for the causal event $\langle \text{earthquake, destroyed} \rangle$, RichGCN fails to identify, which indicates that adopting only document structure graph to capture the association between events from structural features may lack the comprehending of text semantics. However, KIGP accurately identifies “earthquake cause destroyed” by using structural features and also emphasizing semantic features. KIGP can correctly identify all causal pairs in the two cases, indicating that our proposed approach can facilitate the identification of implicit causality by incorporating external knowledge to interact with text and events, thus enhancing the effectiveness of the ECI model.

Finally, the experiments demonstrate that (1) the incorporation of external event knowledge for PLMs can promote the semantic analyzing of events and event relations in texts, and further improve the effect of implicit causality identification through prompt tuning, and (2) the interaction structure features extraction by the event knowledge interaction graph can guide the model to identify causality more precisely and strengthen ECI capability.

5 Conclusion and future work

This paper proposes a novel Knowledge Interaction Graph guided Prompt Tuning (KIGP) to leverage external event knowledge and interaction graph for ECI task. To improve the identification of implicit causalities, we incorporate external event knowledge and design the prompt to maximally activate the powerful learning capability of PLMs. To accurately guide the ECI models and augment the interaction between events and knowledge, we introduce a guidance

mechanism to construct interaction graphs capturing deeply hidden features to enhance event representations in the prompt. Experimental results on two widely used ECI datasets demonstrate that our approach outperforms existing SOTA methods, effectively addressing the challenges of implicit causality identification and event knowledge interaction to some extent. In future work, we will explore automatic prompt template generation for ECI models to further enhance performance. Other knowledge sources like WordNet may provide additional useful knowledge for this task, and we will use WordNet and other knowledge sources as external knowledge for our future research.

Author Contributions Ruijuan Hu: Conceptualization, Methodology, Writing-Original Draft. Jian Li: Software, Data Curation, Methodology, Validation. Haiyan Liu: Supervision. Guilin Qi: Investigation, Project administration. YuXin Zhang: Writing-Review & Editing.

Data availability and access Publicly available datasets were analyzed in this study. EventStoryLine data can be found here: <https://github.com/cltl/EventStoryLine>. Causal-TimeBank data can be found here: <https://dh.fbk.eu/2021/02/causal-timebank/>.

Declarations

Competing interests 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.

Ethical and informed consent for data used Data used in the present study are publicly available, and ethical approval and informed consent were obtained in each original study. The authors confirm that neither the manuscript nor any parts of its content are currently under consideration or published in another journal.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other

third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

References

- Han R, Hsu I-H, Sun J, Baylon J, Ning Q, Roth D, Peng N (2021) Ester: A machine reading comprehension dataset for reasoning about event semantic relations. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 7543–7559
- Zeng C, Li S, Li Q, Hu J, Hu J (2020) A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. *Appl Sci* 10(21):7640
- Rogers A, Gardner M, Augenstein I (2023) Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Comput Surv* 55(10):1–45
- Zadeh A, Chan M, Liang PP, Tong E, Morency L-P (2019) Social-iq: A question answering benchmark for artificial social intelligence. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8807–8817
- Hu W, Yang Y, Cheng Z, Yang C, Ren X (2021) Time-series event prediction with evolutionary state graph. In: Proceedings of the 14th ACM international conference on web search and data mining, pp 580–588
- Lei L, Ren X, Franciscus N, Wang J, Stantic B (2019) Event prediction based on causality reasoning. In: Intelligent information and database systems: 11th Asian Conference, ACIIDS 2019, Yogyakarta, Indonesia, April 8–11, 2019, Proceedings, Part I 11, pp 165–176. Springer
- Do Q, Chan YS, Roth D (2011) Minimally supervised event causality identification. In: Proceedings of the 2011 conference on empirical methods in natural language processing, pp 294–303
- Hashimoto C (2019) Weakly supervised multilingual causality extraction from wikipedia. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 2988–2999
- Gao L, Choubey PK, Huang R (2019) Modeling document-level causal structures for event causal relation identification. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol 1 (Long and Short Papers), pp 1808–1817
- Kadowaki K, Iida R, Torisawa K, Oh J-H, Kloetzer J (2019) Event causality recognition exploiting multiple annotators' judgments and background knowledge. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 5816–5822
- Liu J, Chen Y, Zhao J (2021) Knowledge enhanced event causality identification with mention masking generalizations. In: Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence, pp 3608–3614
- Kenton JDM-WC, Toutanova LK (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of naacL-HLT, vol 1, p 2
- Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized bert pretraining approach. [arXiv:1907.11692](https://arxiv.org/abs/1907.11692)
- Chen X, Zhang N, Xie X, Deng S, Yao Y, Tan C, Huang F, Si L, Chen H (2022) Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In: Proceedings of the ACM Web Conference 2022, pp 2778–2788
- Wei C, Xie SM, Ma T (2021) Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *Adv Neural Inf Process Syst* 34:16158–16170
- Jawahar G, Sagot B, Seddah D (2019) What does bert learn about the structure of language? In: ACL 2019-57th Annual meeting of the association for computational linguistics
- Yenicelek D, Schmidt F, Kilcher Y (2020) How does bert capture semantics? a closer look at polysemous words. In: Proceedings of the Third BlackboxNLP workshop on analyzing and interpreting neural networks for NLP, pp 156–162
- Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. *Adv Neural Inf Process Syst* 33:1877–1901
- Cao P, Zuo X, Chen Y, Liu K, Zhao J, Chen Y, Peng W (2021) Knowledge-enriched event causality identification via latent structure induction networks. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (vol 1: Long Papers), pp 4862–4872
- Liu J, Zhang Z, Guo Z, Jin L, Li X, Wei K, Sun X (2023) Kept: Knowledge enhanced prompt tuning for event causality identification. *Knowl-Based Syst* 259:110064
- Riaz M, Girju R (2013) Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In: Proceedings of the SIGDIAL 2013 conference, pp 21–30
- Riaz M, Girju R (2014) In-depth exploitation of noun and verb semantics to identify causation in verb-noun pairs. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp 161–170
- Riaz M, Girju R (2014) Recognizing causality in verb-noun pairs via noun and verb semantics. In: Proceedings of the EACL 2014 workshop on Computational Approaches to Causality in Language (CAtoCL), pp 48–57
- Riaz M, Girju R (2010) Another look at causality: Discovering scenario-specific contingency relationships with no supervision. In: 2010 IEEE fourth international conference on semantic computing, pp 361–368. IEEE
- Hidey C, McKeown K (2016) Identifying causal relations using parallel wikipedia articles. In: Proceedings of the 54th annual meeting of the association for computational linguistics (vol 1: Long Papers), pp 1424–1433
- Beamer B, Girju R (2009) Using a bigram event model to predict causal potential. In: Computational linguistics and intelligent text processing: 10th International Conference, CICLing 2009, Mexico City, Mexico, March 1-7, 2009. Proceedings 10, pp 430–441. Springer
- Zhao K, Ji D, He F, Liu Y, Ren Y (2021) Document-level event causality identification via graph inference mechanism. *Inf Sci* 561:115–129
- Kruengkrai C, Torisawa K, Hashimoto C, Kloetzer J, Oh J-H, Tanaka M (2017) Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 31
- Kadowaki K, Iida R, Torisawa K, Oh J-H, Kloetzer J (2019) Event causality recognition exploiting multiple annotators' judgments and background knowledge. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 5816–5822

30. Li Z, Li Q, Zou X, Ren J (2021) Causality extraction based on self-attentive bilstm-crf with transferred embeddings. *Neurocomputing* 423:207–219
31. Caselli T, Vossen P (2017) The event storyline corpus: A new benchmark for causal and temporal relation extraction. In: *Proceedings of the events and stories in the news workshop*, pp 77–86
32. Zuo X, Cao P, Chen Y, Liu K, Zhao J, Peng W, Chen Y (2021) Improving event causality identification via self-supervised representation learning on external causal statement. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp 2162–2172
33. Zuo X, Cao P, Chen Y, Liu K, Zhao J, Peng W, Chen Y (2021) Learnra: Learnable knowledge-guided data augmentation for event causality identification. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th international joint conference on natural language processing*, vol 1: Long Papers, pp 3558–3571
34. Zuo X, Chen Y, Liu K, Zhao J (2020) Knowdis: Knowledge enhanced data augmentation for event causality detection via distant supervision. In: *Proceedings of the 28th international conference on computational linguistics*, pp 1544–1550
35. Mukherjee S, Awadallah A (2020) Uncertainty-aware self-training for few-shot text classification. *Adv Neural Inf Process Syst* 33:21199–21212
36. Han X, Zhao W, Ding N, Liu Z, Sun M (2022) Ptr: Prompt tuning with rules for text classification. *AI Open* 3:182–192
37. Si J, Peng X, Li C, Xu H, Li J (2022) Generating disentangled arguments with prompts: A simple event extraction framework that works. In: *ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp 6342–6346. IEEE
38. Ye H, Zhang N, Bi Z, Deng S, Tan C, Chen H, Huang F, Chen H (2022) Learning to ask for data-efficient event argument extraction (student abstract). In: *Proceedings of the AAAI conference on artificial intelligence*, vol 36, pp 13099–13100
39. Huang Y, He K, Wang Y, Zhang X, Gong T, Mao R, Li C (2022) Copner: Contrastive learning with prompt guiding for few-shot named entity recognition. In: *Proceedings of the 29th international conference on computational linguistics*, pp 2515–2527
40. Chen Y, Yang G, Wang D, Li D (2024) Eliciting knowledge from language models with automatically generated continuous prompts. *Expert Syst Appl* 239:122327
41. Liu X, Ji K, Fu Y, Tam W, Du Z, Yang Z, Tang J (2022) P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In: *Proceedings of the 60th annual meeting of the association for computational linguistics (vol 2: Short Papers)*, pp 61–68
42. Zhang H, Ke W, Zhang J, Luo Z, Ma H, Luan Z, Wang P (2023) Prompt-based event relation identification with constrained prefix attention mechanism. *Knowl-Based Syst* 281:111072
43. Shen S, Zhou H, Wu T, Qi G (2022) Event causality identification via derivative prompt joint learning. In: *Proceedings of the 29th international conference on computational linguistics*, pp 2288–2299
44. Ni S, Kao H-Y (2023) Kpt++: Refined knowledgeable prompt tuning for few-shot text classification. *Knowl-Based Syst* 274:110647
45. Liu W, Zhou P, Zhao Z, Wang Z, Ju Q, Deng H, Wang P (2020) K-bert: Enabling language representation with knowledge graph. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 34, pp 2901–2908
46. Zhang N, Deng S, Cheng X, Chen X, Zhang Y, Zhang W, Chen H, Center HI (2021) Drop redundant, shrink irrelevant: Selective knowledge injection for language pretraining. In: *IJCAI*, pp 4007–4014
47. Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 33, pp 7370–7377
48. Phu MT, Nguyen TH (2021) Graph convolutional networks for event causality identification with rich document-level structures. In: *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp 3480–3490
49. Chen M, Cao Y, Deng K, Li M, Wang K, Shao J, Zhang Y (2022) Ergo: Event relational graph transformer for document-level event causality identification. [arXiv:2204.07434](https://arxiv.org/abs/2204.07434)
50. Speer R, Chin J, Havasi C (2017) Conceptnet 5.5: An open multilingual graph of general knowledge. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 31
51. Mirza P, Tonelli S (2014) An analysis of causality between events and its relation to temporal information. In: *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical Papers*, pp 2097–2106
52. Cheng F, Miyao Y (2017) Classifying temporal relations by bidirectional lstm over dependency paths. In: *Proceedings of the 55th Annual meeting of the association for computational linguistics (vol 2: Short Papers)*, pp 1–6
53. Choubey PK, Huang R (2017) A sequential model for classifying temporal relations between intra-sentence events. [arXiv:1707.07343](https://arxiv.org/abs/1707.07343)
54. Mirza P (2014) Extracting temporal and causal relations between events. In: *Proceedings of the ACL 2014 Student research workshop*, pp 10–17

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.