



# Event causality extraction through external event knowledge learning and polyhedral word embedding

Xiao Wei<sup>1</sup> · Chenyang Huang<sup>1</sup> · Nengjun Zhu<sup>1</sup>

Received: 18 April 2023 / Revised: 1 August 2023 / Accepted: 17 October 2023 /

Published online: 22 January 2024

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2024

## Abstract

Extracting causal relations between events from text is vital in natural language processing. Existing methods, which explore the text shallowly, usually aim at casual connection words but neglect implicit causal cues. Furthermore, most of them represent words based solely on contextual semantics, without explicitly considering information related to causality. All of these factors contribute to the inaccuracy of causal relation extraction. To address these issues, in this paper, we propose an event causality extraction method based on external event Knowledge Learning and Polyhedral Word Embedding to alleviate these issues. Specifically, the related background knowledge in knowledge bases is embedded into a vector initially. This infusion of information beyond the sentence allows for the discovery of latent causal relationships. Additionally, we enhance the causal semantic features of words by utilizing their part-of-speech and character features, which helps distinguish causal-related words in sentences. The experimental results on an extended SemEval dataset indicate that our method achieves the best results compared to other existing methods.

**Keywords** Causality extraction · External knowledge · Word information enhancement · Sequence labeling

---

Editors: Bingxue Zhang, Feida Zhu, Bin Yang, João Gama.

---

✉ Nengjun Zhu  
zhu\_nj@shu.edu.cn

Xiao Wei  
xwei@shu.edu.cn

Chenyang Huang  
20721645@shu.edu.cn

<sup>1</sup> The School of Computer Engineering and Science, Shanghai University, 333 Nanchen Road, Shanghai 200444, China

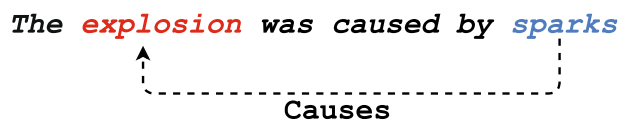
## 1 Introduction

Event causality extraction aims to identify causal relations between events in natural language text. For example, in Fig. 1, the sparks are the cause of the explosion in the sentence “The explosion was caused by sparks”. It plays an important role in question answering (Oh et al., 2013; Dalal et al., 2021), event detection (Radinsky et al., 2012), scene generation (Hashimoto et al., 2014), and other task applications. In texts, the causality of events is expressed in complex forms. The causal relationship between some events is often based on common sense, which means there are cases where the sentence lacks words such as “because” and “therefore” that can clearly indicate the causal connection. This type of causal relationship is called implicit causal relationship, which brings a great challenge to the task of extracting the causal relationship between events.

Recent work (De Silva et al., 2017; Kadowaki et al., 2019) typically divides the causality extraction process into two steps: First, selecting candidate causal event pairs from the text, and then classifying the relationships between these candidate event pairs. But these methods have the problem of error propagation (Yan et al., 2021; Chen et al., 2020). The extraction error of candidate causal event pairs will affect the accuracy of the subsequent relationship classification task. Since joint extraction can mitigate the impact of error propagation (Miwa & Bansal, 2016; Zheng et al., 2017), some researchers use end-to-end models to extract entities and relations simultaneously. The information interaction between entities and relationships is enhanced by enabling the two sub-processes to share the underlying parameters of the network. However, most of the existing work only uses the given text to analyze the causal relationships between events, and it is difficult to discover the implicit causal relationships when explicit causal correlation words are not available. Besides, although existing work considers the influence of contextual information when embedding words, the relevant features of the causality extraction task are still insufficient, so it is difficult to highlight possible causal-related words in a sentence.

To this end, we introduce external knowledge from the knowledge graph into our model. Try to enhance the model’s ability to uncover implicit causal links between events by adding related knowledge of events. Moreover, we integrate the character features of the words as well as the POS properties to address the problem of insufficient causal features. The knowledge graph is a semantic network that contains rich entity relationships. Through the triples in the knowledge graph, we can obtain knowledge related to event entities. For example, background knowledge related to “hurricane” can be described as (hurricane - IsA - natural disaster), (hurricane - Causes - house collapse), etc. The model can make use of the knowledge associated with these events to deduce the hidden causal relationship between events in the absence of explicit causal correlation words in the text, so as to improve the extraction effect of an implicit causal relationship. Besides, most of the event words indicating causality are usually composed of nouns and verbs, and the POS properties of the word have a strong correlation with the causal labels corresponding to the word. And there are similar word morphological features among some of the causal words, as

**Fig. 1** An example of a sentence that contains a causal relationship



*S1:He was **died** in the **disaster***  
*S2:The building was completely **destroyed** by **fire***  
*S3:She was **fired** for **lying** to her boss*

**Fig. 2** Some causal words with similar character structures

shown in Fig. 2<sup>1</sup>. Therefore, we capture the character morphological features of words, and enhance the information of words combined with the POS features to improve the degree of differentiation between causal event words and other words.

Specifically, to model the event-related knowledge representation, we consider the neighbors of the current event in the knowledge graph. Encoding the related knowledge representation of the current event based on the association relationships and weights among those nodes. Meanwhile, we use convolutional neural networks to obtain the character-level feature of input words, and then combine the POS feature to obtain the enhanced word representation of words. After obtaining the related knowledge representation and the word-enhanced information representation, we fuse them with the word representation generated by BERT (Kenton & Toutanova, 2019). Feeding them into Bi-directional Gated Recurrent Unit (Bi-GRU) (Cho et al., 2014) to capture the global features based on the sentence context. Finally, by combining Conditional Random Field (CRF) (Lafferty et al., 2001), we predict the causal role label corresponding to each word in the text.

The contributions of this paper can be summarized as follows:

- To solve the problem of lacking explicit causal correlation words, we introduce external knowledge into our model, so that the model can use the event-related knowledge to establish the implied causal links between events.
- To solve the problem that word representation lacks features related to the causality extraction task, we propose a word information enhancement method. Getting additional information on words from its POS and character features to highlight possible causal-related words in the sentence.
- Experimental results and analysis indicate that our proposed model (KLPWE) has achieved the best results and outperformed other previous state-of-the-art methods.

The structure of the paper is as follows: Sect. 2 introduces related work; In Sect. 3, we present the overall framework and each module of our model; In Sect. 4, we analyze the experimental results and verify the effectiveness of our method; Finally, in Sect. 5, we summarize the work of this paper and discuss the possible future research directions.

## 2 Related work

Our work focuses on using external knowledge to enrich the representation of events, and combining the character morphology and POS of words to enhance the causal semantic features of words, to extract the causal relationship of events in the text. Therefore, it is highly related to Causal Extraction Methods, External Knowledge-Based Methods,

<sup>1</sup> As result words, “died”, “destroyed”, and “fired” all have the same suffix in character morphology “ed”, with the cause event marked in blue and the effect event marked in red in the figure.

Character-Level Feature-Based Methods, and POS-Based Methods. Thus, in this section, we will briefly summarize some of the above-mentioned works.

## 2.1 Causal extraction methods

The early task of extracting event causality mainly adopted the methods based on pattern matching (Ittoo & Bouma, 2011; Kim et al., 2018; Hashimoto et al., 2015). For instance, Khoo et al. (2000) propose an extraction method by combining syntactic trees for causal relationships in the medical domain, Mirza et al. (2014) propose a method for causal labeling between event pairs based on the properties of events. Some studies have used a combination of syntactic patterns and statistical features to extract causal relationships (Luo et al., 2016; Gao et al., 2019). Girju (2003) proposes an inductive learning approach, learning syntactic and semantic constraints of causality by automatic induction of syntactic patterns; For the extraction of causality in medical diseases, Lee and Shin (2017) present a method based on causality frequency and the strength of association between causal event pairs. In recent years, many researchers have started trying to apply deep learning to event causality extraction tasks. Some works (Feng et al., 2018; Khetan et al., 2022; Kadowaki et al., 2019) convert the causal extraction problem into determining whether there is a causal relationship between two events. However, these methods not only rely on the correctness of the event extraction task, but also need to pair all the extracted events. In addition, since a pipeline-based approach is employed in the work, it is difficult to avoid the problem of error propagation and entity redundancy. To address the impact of the aforementioned problems, joint extraction methods (Fu et al., 2011; Martínez-Cámara et al., 2017) based on sequence annotation schemes have emerged. Li et al. (2021) proposed SCITE, and transfer the Flair embedding (Akbik et al., 2018) into their model; Xu et al. Jinghang et al. (2020) extend syntactic dependency trees to syntactic dependency graphs, and propose a graph attention network based on syntactic dependency graphs for identifying event causalities. However, these studies usually focus on the analysis of causality from a given text, and it is often difficult to find more causal clues when the text lacks sufficient causal information.

## 2.2 External knowledge-based methods

With the development of the knowledge graph, many researchers begin to apply external knowledge to natural language processing. Yang and Mitchell propose KBLSTM (Yang & Mitchell, 2017), using external knowledge bases to improve recurrent neural networks for machine reading. BP Majumder et al. Majumder et al. (2022) inject external knowledge into the reply of dialogue models. In terms of event causality extraction, Kruengkrai et al. (2017) retrieve descriptions related to a given causality candidate pair from a large number of knowledge sources, and input them into the multi-column convolutional neural network. Cao et al. propose Latent Structure Induction Network (LSIN) (Cao et al., 2021), learning descriptive knowledge and relational knowledge of events respectively through two different modules, and inferences the causal relationship of events according to the inductive structure. Although previous work has shown that introducing external knowledge can help models better identify causal relationships between events, not all external knowledge is useful in this task. Besides, there are also differences in the importance of knowledge.

### 2.3 Character-level feature-based methods and POS-based methods

In terms of character morphological features, the character-level CNN model was first used to deal with text classification (Zhang et al., 2015). Chung (Chung et al., 2016) proposes a character-level decoder without explicit segmentation; Lee et al. (2017) propose a fully character-level Neural Machine Translation (NMT) model, which proved that character-level CNN could effectively alleviate the problem of Out-Of-Vocabulary (OOV); Chiu and Nichols (2016), Santos and Guimarães (2015) use CNN to learn character-level features of words; The study of Cherry et al. (2018) show that the character-level model can outperform the word-level model with sufficient time and model capacity; R Van Noord et al. (2020) combine the character-level model with the context language model, and find that adding character-level information can still improve the performance of the model even when large pre-trained language models have become very popular. Different from our approach, these studies focus on using character-level information to improve the performance of language models, ignoring the fact that morphological similarity in words can also be used as a feature.

In terms of POS, Fabio (Celli, 2010) adds part-of-speech counting in the process of relation extraction, and finds that POS information was useful for predicting the position of entities in relation. Cai et al. (2019) improve the accuracy of entity boundary detection with the help of POS. For Japanese named entity recognition, M Suzuki et al. Suzuki et al. (2018) use POS tagging to fine-tune name entity recognition (NER), to learn a NER model with high performance. Although POS information has been used in many NLP tasks, few researchers have noticed the association between causal words and POS.

### 2.4 Similarities and Differences Between KLPWE and Other Methods

In general, existing causality extraction methods focus on how to mine as much causal information as possible from a given text, and it is difficult to discover deep implicit causality when the information contained in the text is limited. Therefore, our approach introduces external knowledge into this task, thus providing additional information to the model. Moreover, unlike other external knowledge-based methods, KLPWE also considers the importance of different knowledge. Besides, existing character-level and POS feature-based approaches have demonstrated that the above features are useful in some NLP tasks. Therefore, we introduce them into the event causality extraction task and use these features to improve the differentiation of causal event words.

## 3 Our KLPWE method

In this section, we will introduce the details of our proposed model named KLPWE. We first split a text into a sequence of words, where some words are related to event entities in a knowledge graph. Then, as shown in Fig. 3, our model is divided into four main modules: (1) Entity embedding module, which forms entity representations through their neighbors in a knowledge graph; (2) Static word embedding module, which generates static word representations according to the POS and character-level morphological characteristics of words. This representation is shared across different sentences. (3) Dynamic word embedding module, which outputs dynamic word representation by a Bert model. The semantic

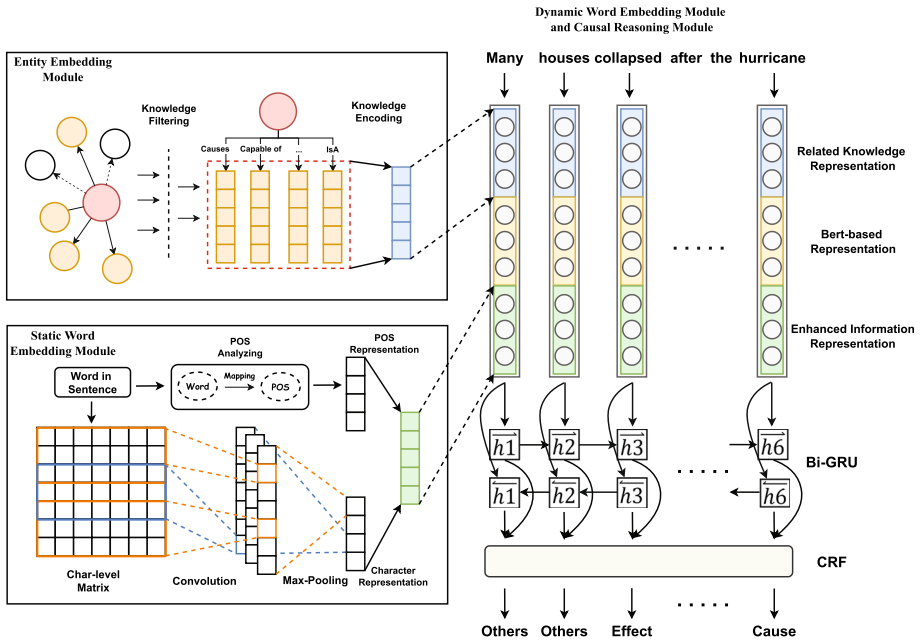


Fig. 3 The overall framework of our model

meaning of each word would be different in different sentences. (4) Causal reasoning module, which utilizes the combination of entity representations, static word representations, and dynamic word representation to construct a more informative token representation. The token representation is then fed to a Bi-GRU to evaluate the causal role of each token in a sentence. Finally, CRF is adopted to jointly decode the label sequence, to assign corresponding causal labels to each word.

The design of KLPWE can incorporate external knowledge graph information, static word morphological characteristics, and dynamic word semantic information, simultaneously. Thus, it can promote the performance of causal reasoning. Next, we will introduce each module of KLPWE in detail.

### 3.1 Entity embedding module

Just as humans can infer implicit causal connections between two event entities with their prior knowledge, as a large-scale semantic network constructed by connections among many entities, the knowledge graph provides a rich source of knowledge for computers to make causal inferences through connections among entities. We use ConceptNet as the source of external knowledge. As one of the most commonly used knowledge graphs, ConceptNet (Speer et al., 2017) contains more than 8 million nodes and 21 million edges, and it assigns weights to each edge according to the strength of association between nodes. Besides, in this module, we recode the knowledge representation of events by using Numberbatch<sup>2</sup>. Numberbatch is a set of static word vectors based on ConceptNet.

<sup>2</sup> <https://github.com/commonsense/conceptnet-numberbatch>

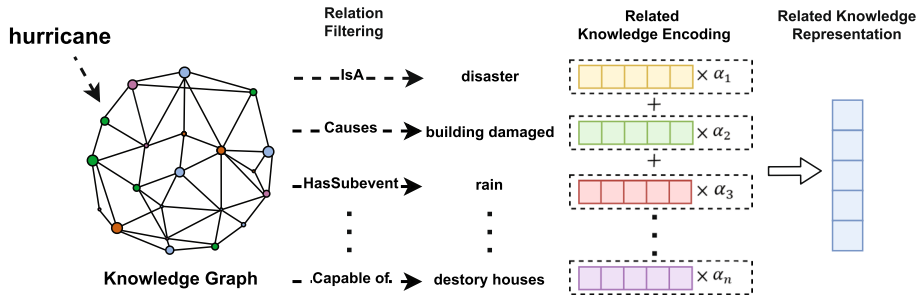


Fig. 4 Entity embedding module

In constructing Numberbatch, the ConceptNet graph is represented as a sparse matrix, and Speer et al. (2017) computes the word embeddings of Numberbatch from this sparse matrix following the same method as Levy et al. (2015). Since it utilizes both semi-structured knowledge and textual information in ConceptNet, it has some semantic features that may not be learned from the trained text corpus alone. The structure of this module is shown in Fig. 4.

Specifically, the entity embedding module consists of two parts: relation filtering and knowledge encoding.

### 3.1.1 Knowledge filtering

In the Knowledge Graph, not all neighbor nodes associated with event entities can be used as the source of event-related knowledge representation in causal extraction tasks. Considering that in ConceptNet, the weight value of a node is calculated based on the credibility of the message, the node with higher credibility has a higher weight. Therefore, we believe that the neighbor nodes with higher weights are more able to represent the knowledge associated with the event. So, when facing a huge amount of related knowledge, selecting the knowledge with a higher weight can better highlight the associated knowledge features of the event.

Thus, for a given event node  $E$ , we search for neighboring nodes associated with it in ConceptNet and filter these nodes according to the type of relationship between these neighboring nodes and  $E$ . Neighbor nodes with relationships such as “Antonym” and “ExternalURL” with  $E$  will be excluded. We only choose “Causes”, “HasSubevent”, “Capable of” and other types of relations that can clearly indicate causality or can be used for causal reasoning. And we retain the top  $n$  neighbor nodes  $\{N_1, N_2, N_3, \dots, N_n\}$  with the highest relevance and their corresponding association weights  $\{W_1, W_2, W_3, \dots, W_n\}$ .

### 3.1.2 Knowledge encoding

After obtaining the neighbors’ information related to the given event, we encode the related knowledge representation of this event based on these neighbor nodes and their weights. At this stage, we normalize the weight of each neighbor node. The final event-related knowledge representation is formed by combining multiple knowledge representation vectors according to their normalized weights. For each neighbor  $N_j$ , We define its normalized weights according to the following equation:

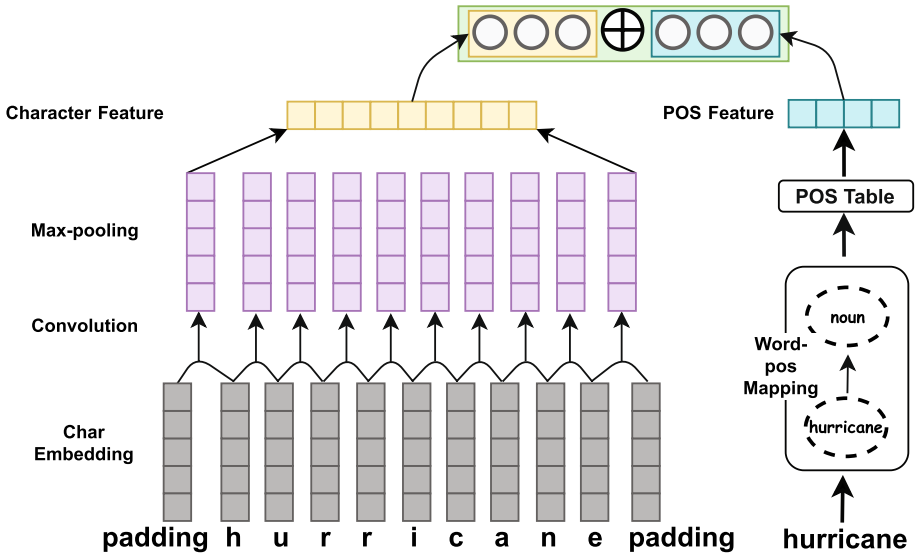


Fig. 5 Static word embedding module

$$\alpha_j = \frac{W_j}{\sum_{i=1}^n W_i} \tag{1}$$

Where  $W_j$  denotes the weight value of the associated edge between  $\alpha_j$  and  $E$ . After that, we calculate the related knowledge representation  $F^{knowl}$  of event node  $E$  according to equation (2):

$$F^{knowl} = \sum_{j=1}^n \alpha_j v_j \tag{2}$$

where  $v_j$  is the feature vector of  $N_j$  in Numberbatch.

### 3.2 Static word embedding module

In the event causality extraction task, our goal is to identify the event words in sentences with causal semantic role labels, and distinguish these words from others that are not causally related. Previous methods usually obtain the contextual features of words directly based on the initial semantic vector, but this is not enough to highlight the causal features of words. Especially when the context lacks connecting words such as “because” and “cause”, which can explicitly indicate causality. Since causal words often have similar character-level morphological features among themselves, so we extract the character morphological features of the word as an enhanced Information representation of it. In addition, since causal events in sentences are usually composed of verbs and nouns, there is a certain correlation between word POS and causal semantic role labels. So, we add the POS feature as part of the enhanced information as well. Combining character morphological features and POS features together to construct



static word embeddings, to highlight the causal features of words. The structure of this module is shown in Fig. 5.

The module consists of two parts: character feature capture and POS feature capture, which will be shown in detail in the following subsections.

### 3.2.1 Character feature capture

Often these words have the same place in the syntactic structure of the sentence. Learning the character morphological representation of these words can highlight the local vocabulary in the sentence, and then better help the model learn the common position of causal words in the sentence structure.

Previous studies (Santos & Guimarães, 2015; Labeau et al., 2015) have demonstrated the effectiveness of CNN in extracting word character-level features. To capture the character features of causal words, we use the same convolutional neural network as Chiu et al Chiu and Nichols (2016), splitting the words into multiple characters for convolution. In the convolution process, to avoid the problem of information loss, we first fill the boundaries of the word. For a given word  $W$  of length  $t$ , we split it by character to obtain the set of characters  $\{c_1, c_2, \dots, c_t\}$ . Subsequently, we look up the character feature vector  $v_i$  corresponding to each character  $c_i$  from the character-to-character feature mapping table and construct the character feature vector matrix  $R^{m \times d}$  corresponding to  $W$  by combining the features of the filled character. Let the set of convolution kernels  $K = \{k_1, k_2, \dots, k_n\}$ , then for a local feature  $f_i^c$ , it can be calculated by the following equation:

$$f_i^c = f(w \cdot v_{i:i+l-1} + b) \quad (3)$$

where  $w \in R^{m \times d}$ ,  $l$  is the window length of the convolution kernel  $k_i$ ,  $b$  is the bias value,  $f_i^c$  denotes the feature obtained by the  $i$ -th filter, and  $f$  is the activation function Relu. We compute the convolution of features for each window that  $k_i$  slides through, get  $F_i^c = \{f_1^c, f_2^c, \dots, f_{m-l+1}^c\}$ . And then perform maximum pooling to obtain the feature  $\tilde{F}_i^c = \max(F_i^c)$  corresponding to this convolution kernel. Eventually, for a given word  $W$ , its character features under the action of  $n$  convolutional kernels in the set  $K$  of convolutional kernels are represented as:

$$F^c = \{\tilde{F}_1^c, \tilde{F}_2^c, \dots, \tilde{F}_n^c\} \quad (4)$$

### 3.2.2 POS feature capture

In general, the words with causal role labels are usually the core ones in causal sentences. Considering that in the event causality extraction task, words such as determiners, gerunds, complements, and other modifying and restricting words are relatively less important in the sentence and not usually in the key structure of the sentence. So, we can distinguish the POS of words to further highlight the influence that each word has on the sentence. Based on the above, we build a POS table and initialize the feature vector for each POS in the table. For the input sentence, we perform part-of-speech tagging on each word in the sentence to obtain the POS of word  $W$ . After this, we look up the corresponding POS feature embedding  $F^p$  of  $W$  in the POS table according to its POS.

$$F^{word} = F^p \oplus F^c \quad (5)$$

### 3.3 Dynamic word embedding module

In this module, we combine the related knowledge feature and word information enhancement feature with the dynamic word vector from the pre-trained language model, and get the final word representation as the input of the neural network layer.

To perform the feature fusion, we need to convert the input text into the corresponding word vector representation. BERT is a pre-trained language model built on the bidirectional transformer. Since BERT is pre-trained with the help of MLM (Masked Language Model) and NSP (Next Sentence Prediction) tasks, it has a powerful semantic acquisition capability and can effectively solve the problem of multiple meanings of words.

In our model, we use BERT-base to model the text. For each word  $w_i$  in input sentence  $S = \{w_1, w_2, \dots, w_i\}$ , after BERT encoding, we use the output  $F_i^{bert}$  as the word embedding, and fuse it with the previously obtained background knowledge representation and word information enhancement representation to obtain the final representation of words:

$$F_i = F_i^{bert} \oplus F_i^{knowl} \oplus F_i^{word} \tag{6}$$

### 3.4 Causal reasoning module

In this module, we predict the causal label with the highest probability for each word in the sentence based on the output of the dynamic word embedding module.

Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) is a special kind of recurrent neural network structure. It regulates the sequence of information by designing a structure called “gates”, which can selectively preserve the contextual information in sentences. We use the Gated Recurrent Unit to model the global semantic feature of a word. GRU is a variant of the LSTM network, which has a simpler structure and faster training speed compared with LSTM. For the input semantic feature vector, GRU is calculated using the following formula:

$$\begin{aligned} z_i &= \sigma(W_z F_i + U_z h_{i-1} + b_z) \\ r_i &= \sigma(W_r F_i + U_r h_{i-1} + b_r) \\ \tilde{h}_i &= \tanh(W_h F_i + U_h (r_i \times h_{i-1}) + b_h) \\ h_i &= z_i \times h_{i-1} + (1 - z_i) \times \tilde{h}_i \end{aligned} \tag{7}$$

where  $\sigma$  is the sigmoid activation function,  $F_i$  represents the fused feature vector corresponding to the  $i$ -th word in the input sentence,  $W_z, W_r, W_h, U_z, U_r, U_h$  is the weight matrix in GRU,  $b_z$  and  $b_r$  are the bias variables.

Considering that the cause event and effect event in the sentence is context-dependent, the forward GRU can only consider the text before the current word, so we add the backward GRU and use the bidirectional GRU to obtain global semantic features. Finally, the output  $h_t$  of GRU layer is determined by both the forward and backward GRU:

$$h_t = \text{concat}(\overrightarrow{h}_t, \overleftarrow{h}_t) \tag{8}$$

where  $\overrightarrow{h}_t, \overleftarrow{h}_t$  respectively denotes the output vector after  $F_i$  goes through forward GRU and backward GRU, and concat denotes the splicing function between vectors.

In the causal extraction task, there is usually a strong dependency between the causal semantic role label of words. For an “Effect” label, there must be a corresponding “Cause” label. To use the constraint relationship between causal labels, we take Conditional Random Field (CRF) to assign final causal semantic role labels for words in the sentence, and obtain a globally optimal label chain for the given input sequence. CRF is a special case of Markov random field, which is able to predict the conditional probability distribution of the output sequence corresponding to a set of given input sequences. We take the global semantic feature vector of sentence  $S$  after passing through the Bi-GRU network layer as the input of the CRF layer. For the given sentence  $S = \{w_1, w_2, \dots, w_n\}$  and label sequence  $y = \{y_1, y_2, \dots, y_n\}$ , CRF uses the following formula for scoring:

$$\text{score}(S, y) = \sum_{i=0}^n A_{y_i, y_{i+1}} + \sum_{i=1}^n P_{i, y_i} \quad (9)$$

where  $A$  is the transfer matrix,  $A_{y_i, y_{i+1}}$  denotes the transfer score from the label  $y_i$  to  $y_{i+1}$ , and  $P_{i, y_i}$  denotes the probability that the  $i$ -th word is labeled as  $y_i$ . For input sentence  $S$ , we calculated the probability of tag sequence  $y$  based on the above scoring formula:

$$p(y | S) = \frac{e^{\text{score}(S, y)}}{\sum_{\tilde{y} \in Y_S} \text{score}(S, \tilde{y})} \quad (10)$$

where  $Y_S$  denotes all possible label combinations of  $S$ , and  $\tilde{y}$  denotes the real label. The model is trained by the maximum likelihood function to maximize  $p(y | S)$ :

$$\log(p(y | S)) = \text{score}(S, y) - \log \left( \sum_{\tilde{y} \in Y_S} e^{\text{score}(S, \tilde{y})} \right) \quad (11)$$

Finally, the highest scoring predicting label sequence will be output by the following formula:

$$y^* = \arg \max_{\tilde{y} \in Y_S} \text{score}(S, \tilde{y}) \quad (12)$$

## 4 Experiments

### 4.1 Dataset

We extend the annotation of causal sentences based on SemEval 2010 Task8 (Hendrickx et al., 2010). There are some ambiguous annotations in the original annotations of the dataset. For example, in the sentence “These <e1>germs</e1> cause illnesses ranging from common ailments, like the cold and <e2>flu</e2>, to disabling.”, “cold” and “flu” are specific cases under the concept of “illnesses”. The original dataset, however, only labels “flu” as the effect. To address the impact of ambiguous annotations on reliability and accuracy in the original dataset, we relabeled the original dataset and extended it. For these ambiguous annotations, we use the word with the highest conceptual level in the sentence as the final annotation. In the above example, “illnesses” will be labeled as “Effect”. In addition, for the annotation of phrase types, we uniformly select the most core word in the phrase as the annotation result. Finally, our corpus consists of 3000 sentences with 1331

causal instances. We divide our dataset into train set, validation set, and test set by the ratio of 4.5:1:1.

## 4.2 Evaluation metrics

Same as the previous method, we use Precision, Recall, and F1-score as evaluation metrics, which can be calculated by the following formulas:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (13)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (14)$$

$$\text{F1 - Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (15)$$

Where TP is True Positive, denotes the predicted value is true and the actual value is true. FP is False Positive, which denotes the predicted value is true and the actual value is false. FN is False Negative, which denotes the predicted value is false and the actual value is true.

## 4.3 Experimental settings

We use the “bert-base-uncased” model under BERT to get the embedding representation of input text. Set the batch size to 8, the learning rate to  $1 \times 10^{-5}$ , the epoch of training to 50, and the hidden size layer of GRU to 256. And based on the average length of sentences in the dataset, the maximum length of sentences is set to 64. In the entity embedding module, we keep the top 10 neighbor nodes with the highest relevance and set the dimension of background knowledge embedding to 300. In the static word embedding module, we use CNN with 128 convolution kernels, set the window size of convolution to 3, and obtain 37 different word properties based on NLTK’s word annotation library.

## 4.4 Results and analysis

We compared our model with baselines and conducted ablation experiments as a way to demonstrate the effectiveness of our work. Each experiment has been performed five times, and then evaluation metrics were calculated based on multiple experiments. We selected IDCNN, CLSTM, and other mainstream methods for comparison:

**IDCNN-CRF** (Strubell et al., 2017): The model uses Iterated Dilated Convolutions to replace Bi-LSTM, which allows convolutions of fixed depth to run in parallel throughout the document. The Iterated Dilated Convolutions significantly improves the speed of training while maintaining the same accuracy as Bi-LSTM.

**CLSTM-BiLSTM-CRF** (Lample et al., 2016): The model uses a bidirectional LSTM as a character encoder (Char LSTM) to generate word embeddings deriving from characters, which are connected to pre-trained word vectors in the word table as input to the lower-level model. The bidirectional LSTM encoder enables the model to benefit from both word and character-level representations.

**Table 1** The performance of different models on the test set

Model	Precision	Recall	F1
IDCNN-CRF	0.7560	0.5960	0.6665
CNN-BiLSTM-CRF	0.7457	0.7030	0.7237
CLSTM-BiLSTM-CRF	0.7757	0.7592	0.7674
BERT-BiLSTM-CRF	0.7893	0.8263	0.8073
KLPWE w/o word	<b>0.8123</b>	0.8175	0.8149
KLPWE w/o knowl	0.8040	0.8210	0.8124
KLPWE	0.8073	<b>0.8280</b>	<b>0.8175</b>

Best results of each column are highlighted in bold

KLPWE w/o word and KLPWE w/o knowl respectively denote models without the static word embedding module and the entity embedding module

**CCNN-BiLSTM-CRF** (Ma et al., 2016): Similar to the previous model, but the difference is that this model uses CNN as a character encoder (Char CNN) to learn word features instead of CLSTM.

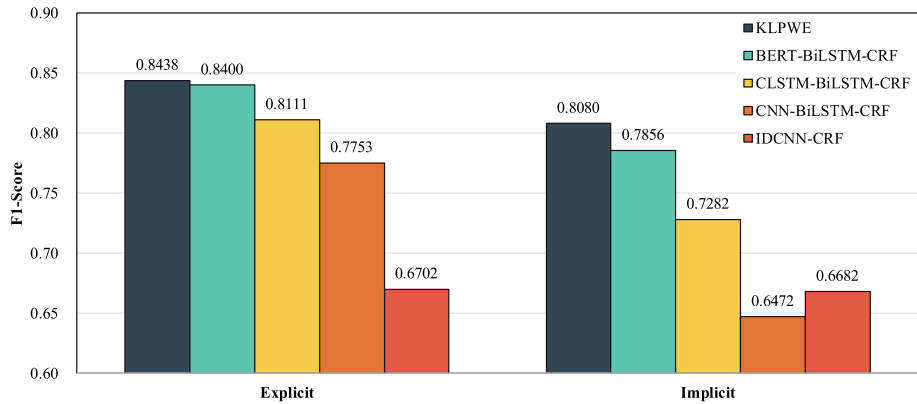
**BERT-BiLSTM-CRF**: This is a widely used model in sequence annotation tasks and extended on the basis of Huang et al. (2015). The model uses BERT as a pre-trained model to obtain dynamic word vectors based on contextual contexts as input to the lower-level model, which can handle the presence of multiple meanings of a word.

Table 1 shows the experimental results of different models for the causal extraction task. We can find that our model has achieved an F1 score of 0.8175 in test sets, outperforming the other models, thus confirming the validity of our work. Meanwhile, to verify the role of the entity embedding module and the static word embedding module, we conduct ablation experiments on our model. We test the performance of our model in the absence of entity embedding module and static word embedding module respectively. The final results show that adding both modules has improved the performance to different degrees, and both achieved better results than the baseline model. Moreover, using both modules together can further improve the model's performance, thus verifying the effectiveness of our proposed module for the event causality extraction task.

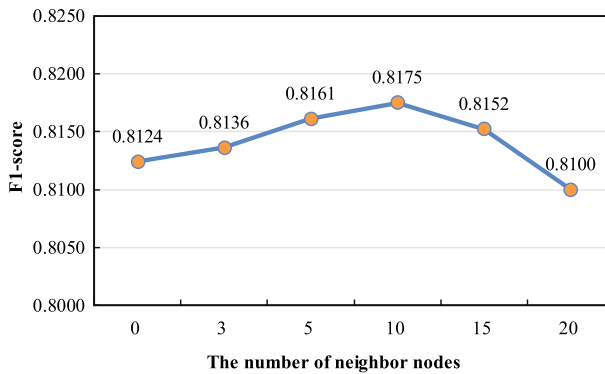
#### 4.5 The effect of causal connection words

To explore the impact of causal connection words in sentences for model extraction performance, we select the sentences with causal instances in the test set and manually classify these sentences into explicit causal sentences with causal connection words and implicit causal sentences without causal connection words. Finally, we obtain 176 explicit causal sentences and 52 implicit causal sentences. We only use the selected explicit and implicit causal sentences as the test set for the experiments in this section, and test the performance of different models in extracting explicit and implicit causal relationships between events, respectively, and the results are shown in FIG.6.

We observe the following: (1) Compared with extracting explicit causality, the performance of each model decreases to different degrees when extracting implicit causality, which indicates that the lack of causal correlation words brings difficulties in mining the deep implicit causality in sentences. (2) Compared with the baseline model, our



**Fig. 6** The performance of different models in extracting explicit and implicit causality tasks



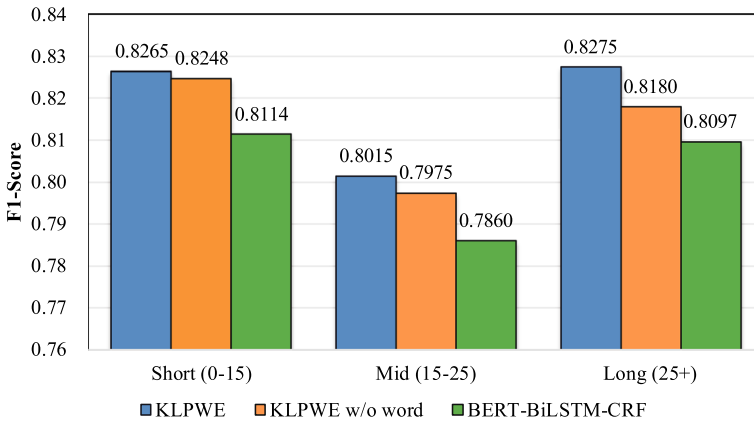
**Fig. 7** F1 scores with the different numbers of neighbor nodes

model achieves the best results on both tasks, especially getting a larger improvement in the implicit causality extraction task. It achieves an improvement of 0.38% on the explicit causality extraction task and 2.24% on the implicit causality extraction task. It is confirmed that our method can effectively alleviate the problem of missing causal correlation words in the sentence, and provide more information on causal clues for the causality extraction task.

#### 4.6 The effect of neighbor node counts

To investigate the effect of the number of relevant neighbor nodes on model extraction results, we select the top 3, 5, 10, 15, and 20 neighbor nodes with the highest relevance ranking to event nodes as the background knowledge sources for knowledge representation encoding, and conduct comparative experiments. Fig. 7 shows the experimental results.

In Fig. 7, we can observe that the F1 score of our model increases with the number of selected neighbors, and reaches the highest score when the number of selected neighbor nodes increases to 10. As the number of selected neighbor nodes continues to increase, the F1 score starts to show a decreasing trend. When the number of selected neighbor nodes



**Fig. 8** Results on test sets of different lengths

reaches 20, the score decreases instead by 0.24% compared with the benchmark model with no related knowledge representation. We have analyzed this, the reason is probably that the small number of relevant neighbor nodes limits the scope that event-related knowledge can cover, leading to a less comprehensive knowledge representation generated. So, a proper number of neighbor nodes can contribute to providing a more adequate representation of event-related knowledge features. When selecting too many neighbor nodes, those related knowledge features with a high association will be diluted, which results in lower quality of the generated event knowledge representation. These excessively-diluted related knowledge features not only make it difficult to represent the relevant knowledge of the event, but even bring negative effects to the model.

#### 4.7 Analysis of static word embedding module

To further analyze the effect of the static word embedding module on the representation of causal semantic features, we divide the test set into Short ( $0 < l < 15$ ), Mid ( $15 \leq l < 25$ ), and Long ( $25 \leq l$ ) according to the length  $l$  of the sentence, with a ratio of roughly 2:2:1. And we conduct experiments on the segmented test set respectively based on baseline model (BERT-BiLSTM-CRF), model without the static word embedding module (KLPWE w/o word), and the final completed model (KLPWE). As shown in Fig. 8, on each test set with different sentence lengths, models using the static word embedding module obtained a certain degree of improvement compared with the baseline model. It is worth noting that compared with the model without the static word embedding module, the F1 score of the completed model only improved by 0.17% on the short-sentence test set. However, the improvement of the completed model reached 0.4% and 0.95% on the mid and long-sentence test sets respectively. This demonstrates that in the case of longer sentences, the model with static word embedding module can effectively highlight important words with causal relevance among numerous words, which verifies the effectiveness of this module in enhancing the representation of word causal semantic features.

**Table 2** Result of causality extraction, where the correct part is shown in underlined italic, and the wrong part is shown in bold italic. We use bold to highlight cause-and-effect words in sentences

Sentence 1	Merlin Lindeman (animal sciences) then pooled their expertise to show that the <b>caterpillars</b> caused the <b>disease</b> .
True Causal Labels	{ Cause : <i>caterpillars</i> , Effect : <i>disease</i> }
IDCNN	{ Cause : <b>None</b> , Effect : <i>disease</i> }
CNN-BiLSTM-CRF	{ Cause : <i>caterpillars</i> , Effect : <i>disease</i> }
CLSTM-BiLSTM-CRF	{ Cause : <i>caterpillars</i> , Effect : <i>disease</i> }
BERT-BiLSTM-CRF	{ Cause : <i>caterpillars</i> , Effect : <i>disease</i> }
KLPWE	{ Cause : <i>caterpillars</i> , Effect : <i>disease</i> }
Sentence 2	<b>Landslides</b> caused the majority of the <b>deaths</b> .
True Causal Labels	{ Cause : <i>Landslides</i> , Effect : <i>deaths</i> }
IDCNN	{ Cause : <i>Landslides</i> , <b>deaths</b> , Effect : <b>None</b> }
CNN-BiLSTM-CRF	{ Cause : <b>None</b> , Effect : <i>deaths</i> }
CLSTM-BiLSTM-CRF	{ Cause : <i>Landslides</i> , Effect : <b>majority</b> , <i>deaths</i> }
BERT-BiLSTM-CRF	{ Cause : <i>Landslides</i> , Effect : <i>deaths</i> }
KLPWE	{ Cause : <i>Landslides</i> , Effect : <i>deaths</i> }
Sentence 3	<b>Methadone</b> gives a <b>buzz</b> too, but it's the most effective way to reel in heroin addicts into treatment and real life.
True Causal Labels	{ Cause : <i>Methadone</i> , Effect : <i>buzz</i> }
IDCNN	{ Cause : <b>None</b> , Effect : <b>None</b> }
CNN-BiLSTM-CRF	{ Cause : <b>addicts</b> , Effect : <b>None</b> }
CLSTM-BiLSTM-CRF	{ Cause : <b>None</b> , Effect : <b>None</b> }
BERT-BiLSTM-CRF	{ Cause : <b>reel</b> , Effect : <i>buzz</i> , <b>treatment</b> }
KLPWE	{ Cause : <i>Methadone</i> , Effect : <i>buzz</i> }

## 4.8 Case study

In Table 2, we present some representative examples to illustrate the differences between our proposed approach and other approaches. For each example, we show the input sentence and the causal event words contained in the sentence in the first line. The remaining lines show the causal extraction results of our model and other models.

Sentences 1 and 2 are examples of explicit causality. From this, we observe that explicit causal correlation words can help to extract causal relationships to a certain extent. Most methods can identify explicit causal relationships when the distance between events is close. However, when explicit causal relationships are far apart, even if “caused” can serve as an indicator of causality, methods that do not use pre-trained language models cannot correctly extract the causal relationships. We analyzed this situation, and the reason may be that pre-trained language models can dynamically generate word vectors based on the sentence context, resulting in more accurate semantic representations of words. Therefore,



compared to methods that do not use pre-trained language models, they can achieve better results.

Sentence 3 is an example of implicit causality. From this, we observe that the lack of explicit causal correlation words in sentences presents a significant challenge for learning implicit causal relationships. In this example, only KLPWE can correctly extract the underlying causal relationships between events when compared with other models.

## 5 Conclusion

In this paper, we propose a method for event causality extraction based on external event knowledge learning and polyhedral word embedding. To alleviate the problem that the model has difficulty in discovering implicit causal associations between events in the absence of causal clues in the text, we generated related knowledge representation for events through external knowledge. In addition, to address the lack of causal extraction task-related features in semantic representations of words, we performed an information enhancement representation of the word to highlight the causal-related features. The experimental results verified the effectiveness of our proposed method.

In future work, we will try to extract multiple causal relationships simultaneously from sentences, and extend the extraction of causality from one causal pair to multiple causal pairs. Furthermore, the event-related knowledge can be further extended based on nodes on the multi-hop paths in the knowledge graph. Therefore, investigating how to utilize the relevant knowledge on multi-hop paths for causality extraction is also a potential direction for future research.

**Author Contributions** XW and NZ directed the study. XW contributes to the idea, experiment, theoretical analysis, and writing. CH has contributed to the idea, experiment, writing and theoretical analysis. NZ has contributed to experiments, writing and theoretical analysis. All authors read and approved the final manuscript.

**Funding** This work is partially supported by National Natural Science Foundation of China under Grant No. 62202282 and Shanghai Youth Science and Technology Talents Sailing Program under Grant No. 22YF1413700.

**Availability of data and materials** The SemEval datasets are available from (<https://semeval2.fbk.eu/semeval2.php?location=data>). Our expanded data set will be made publicly available with our code.

**Code availability** Code to conduct experiments will be made publicly available upon publication.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethics approval** This work does not involve any human subjects or animals, so has no ethical concerns.

**Consent to participate** Not Applicable.

**Consent for publication** All authors consent to submission and publication.

## References

- Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20–26, 2018*, pp. 1638–1649.
- Cai, X., Dong, S., & Hu, J. (2019). A deep learning model incorporating part of speech and self-matching attention for named entity recognition of chinese electronic medical records. *BMC Medical Informatics Decision Making*, 19S(2), 101–109.
- 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, ACL/IJCNLP 2021, (Vol. 1: Long Papers), Virtual Event, August 1–D-6, 2021*, pp. 4862–4872 (2021)
- Celli, F. (2010). UNITN: part-of-speech counting in relation extraction. In: *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15–16, 2010*, pp. 198–201.
- Chen, D., Li, Y., Lei, K., & Shen, Y. (2020). Relabel the noise: Joint extraction of entities and relations via cooperative multiagents. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5940–5950.
- Cherry, C., Foster, G.F., Bapna, A., Firat, O., & Macherey, W. (2018). Revisiting character-based neural machine translation with capacity and compression. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31–November 4, 2018*, pp. 4295–4305.
- Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4, 357–370.
- Cho, K., Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, October 25–29, 2014, Doha, Qatar, A Meeting of SIGDAT, a Special Interest Group of The ACL*, pp. 1724–1734.
- Chung, J., Cho, K., & Bengio, Y. (2016). A character-level decoder without explicit segmentation for neural machine translation. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers*.
- Dalal, D., Arcan, M., & Buitelaar, P. (2021). Enhancing multiple-choice question answering with causal knowledge. In: *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 70–80.
- De Silva, T. N., Zhibo, X., Rui, Z., & Kezhi, M. (2017). Causal relation identification using convolutional neural networks and knowledge based features. *International Journal of Computer and Systems Engineering*, 11(6), 696–701.
- Feng, C., Kang, L.Q., Shi, G., & Huang, H.Y. (2018) Causality extraction with gan. *Zidonghua Xuebao/Acta Automatica Sinica*, 44, 811–818.
- Fu, J., Liu, Z., Liu, W., & Zhou, W. (2011). Event causal relation extraction based on cascaded conditional random fields. *Pattern Recognition and Artificial Intelligence*, 24(4), 567–573.
- Gao, L., Choubey, P.K., & 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers)*, pp. 1808–1817.
- Girju, R. (2003). Automatic detection of causal relations for question answering. In: *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pp. 76–83.
- Hashimoto, C., Torisawa, K., Kloetzer, J., & Oh, J. (2015). Generating event causality hypotheses through semantic relations. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, January 25–30, 2015, Austin, Texas, USA, pp. 2396–2403.
- Hashimoto, C., Torisawa, K., Kloetzer, J., Sano, M., Varga, I., Oh, J.-H., & Kidawara, Y. (2014). Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 987–997 (2014)
- Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Séaghdha, D.Ó., Padó, S., Pennacchiotti, M., Romano, L., & Szpakowicz, S. (2010). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15–16, 2010*, pp. 33–38.

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. CoRR arXiv: 1508.01991.
- Ittoo, A., & Bouma, G. (2011). Extracting explicit and implicit causal relations from sparse, domain-specific texts. In: *2014, Natural Language Processing and Information Systems—16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011, Alicante, Spain*.
- Jinghang, X., Wanli, Z., Shining, L., & Ying, W. (2020). Causal relation extraction based on graph attention networks. *Journal of Computer Research and Development*, 57(1), 159.
- Kadowaki, K., Iida, R., Torisawa, K., Oh, J., & 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 2019, Hong Kong, China, November 3–7, 2019*, pp. 5815–5821.
- Kenton, J.D.M.-W.C., & Toutanova, L.K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL-HLT*, pp. 4171–4186.
- Khetan, V., Ramnani, R., Anand, M., Sengupta, S., & Fano, A.E. (2022) Causal bert: Language models for causality detection between events expressed in text. In: *Computing Conference, 2021*, pp. 965–980.
- Khoo, C.S.G., Chan, S., & Niu, Y. (2000). Extracting causal knowledge from a medical database using graphical patterns. In: *38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1–8, 2000*, pp. 336–343.
- Kim, H., Joung, J., & Kim, K. (2018). Semi-automatic extraction of technological causality from patents. *Computers and Industrial Engineering*, 115, 532–542.
- Kruengkrai, C., Torisawa, K., Hashimoto, C., Kloetzer, J., Oh, J., & Tanaka, M. (2017). Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4–9, 2017, San Francisco, California, USA*, pp. 3466–3473.
- Labeau, M., Löser, K., & Allauzen, A. (2015). Non-lexical neural architecture for fine-grained POS tagging. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17–21, 2015*, pp. 232–237.
- Lafferty, J.D., McCallum, A., & Pereira, F.C.N. (2001). Conditional random fields: Probabilistic models for named entity and labeling sequence data. In: *2014, Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA*.
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. In: *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*, pp. 260–270 (2016)
- Lee, D., & Shin, H. (2017). Disease causality extraction based on lexical semantics and document-clause frequency from biomedical literature. *BMC Medical Informatics Decision Making*, 17(S-1), 53–1539.
- Lee, J., Cho, K., & Hofmann, T. (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5, 365–378.
- Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3, 211–225. [https://doi.org/10.1162/tacl\\_a\\_00134](https://doi.org/10.1162/tacl_a_00134).
- Li, Z., Li, Q., Zou, X., & Ren, J. (2021). Causality extraction based on self-attentive bilstm-crf with transferred embeddings. *Neurocomputing*, 423, 207–219.
- Luo, Z., Sha, Y., Zhu, K.Q., Hwang, S., & Wang, Z. (2016) Commonsense causal reasoning between short texts. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25–29, 2016*, pp. 421–431.
- Ma, X., Hovy, & E.H. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Vol. 1: Long Papers*.
- Majumder, B.P., Jhamtani, H., Berg-Kirkpatrick, T., & McAuley, J.J. (2022) Achieving conversational goals with unsupervised post-hoc knowledge injection. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022*, pp. 3140–3153.
- Martínez-Cámara, E., Shwartz, V., Gurevych, I., & Dagan, I. (2017) Neural disambiguation of causal lexical markers based on context. In: *IWCS 2017—12th International Conference on Computational Semantics - Short Papers, Montpellier, France, September 19–22, 2017*.

- Mirza, P., Sprugnoli, R., Tonelli, S., & Speranza, M. (2014). Annotating causality in the tempeval-3 corpus. In: *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pp. 10–19.
- Miwa, M., & Bansal, M. (2016). End-to-end relation extraction using lstms on sequences and tree structures. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics*. <https://doi.org/10.18653/v1/p16-1105>.
- Noord, R., Toral, A., & Bos, J. (2020). Character-level representations improve drs-based semantic parsing even in the age of BERT. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, Online, November 16–20, 2020, pp. 4587–4603.
- Oh, J.-H., Torisawa, K., Hashimoto, C., Sano, M., De Saeger, S., & Ohtake, K. (2013). Why-question answering using intra-and inter-sentential causal relations. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers)*, pp. 1733–1743
- Radinsky, K., Davidovich, S., & Markovitch, S. (2012). Learning causality for news events prediction. In: *Proceedings of the 21st International Conference on World Wide Web*, pp. 909–918.
- Santos, C.N., & Guimarães, V. (2015). Boosting named entity recognition with neural character embeddings. In: *Proceedings of the Fifth Named Entity Workshop, NEWS@ACL 2015, Beijing, China, July 31, 2015*, pp. 25–33.
- Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, February 4–9, 2017, San Francisco, California, USA, pp. 4444–4451.
- Strubell, E., Verga, P., Belanger, D., & McCallum, A. (2017). Fast and accurate entity recognition with iterated dilated convolutions. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017*, pp. 2670–2680.
- Suzuki, M., Komiya, K., Sasaki, M., & Shinnou, H. (2018). Fine-tuning for named entity recognition using part-of-speech tagging. In: *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, PACLIC 2018, Hong Kong, December 1–3, 2018*.
- Yan, Z., Zhang, C., Fu, J., Zhang, Q., & Wei, Z. (2021). A partition filter network for joint entity and relation extraction. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 185–197.
- Yang, B., & Mitchell, T.M. (2017). Leveraging knowledge bases in lstms for improving machine reading. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30–August 4, Vol. 1: Long Papers*, pp. 1436–1446.
- Zhang, X., Zhao, J.J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada*, pp. 649–657.
- Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., & Xu, B. (2017). Joint extraction of entities and relations based on a novel tagging scheme. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers)*, pp. 1227–1236.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.