# SEAG: Structure-Aware Event Causality Generation

**Zhengwei Tao**[1] **Zhi Jin**[1*] **Xiaoying Bai**[2*] **Haiyan Zhao**[1]
**Chengfeng Dou**[1] **Yongqiang Zhao**[1] **Fang Wang**[1] **Chongyang Tao**[1]
[1]Peking University, [2]Advanced Institute of Big Data
{tttzw,yongqiangzhao,fangwang}@stu.pku.edu.cn, baixy@aibd.ac.cn
{zhijin,zhhy.sei,chengfengdou,chongyangtao}@pku.edu.cn

## Abstract

Extracting event causality underlies a broad spectrum of natural language processing applications. Cutting-edge methods break this task into Event Detection and Event Causality Identification. Although the pipelined solutions succeed in achieving acceptable results, the inherent nature of separating the task incurs limitations. On the one hand, it suffers from the lack of cross-task dependencies and may cause error propagation. On the other hand, it predicts events and relations separately, undermining the integrity of the event causality graph (ECG). To address such issues, in this paper, we propose an approach for Structure-Aware Event Causality Generation (SEAG). With a graph linearization module, we generate the ECG structure in a way of text2text generation based on a pre-trained language model. To foster the structural representation of the ECG, we introduce the novel Causality Structural Discrimination training paradigm in which we perform structural discriminative training alongside auto-regressive generation enabling the model to distinguish from constructed incorrect ECGs. We conduct experiments on three datasets. The experimental results demonstrate the effectiveness of structural event causality generation and the causality structural discrimination training.

## 1 Introduction

Event Causality plays an essential role in Natural Language Processing (Girju, 2003). Event Causality Extraction aims to recognize events and their inter-causal relations from text. As shown in Figure 1, given the input text, the model should be able to identify three events, i.e., "suffered", "invasion" and "destroyed", and the causal relations in between. Extracting event causality has impactful applications such as question answering (Yang et al., 2022; Ho et al., 2022), event forecasting (Hashimoto et al., 2014) and reading comprehension (Berant et al., 2014).

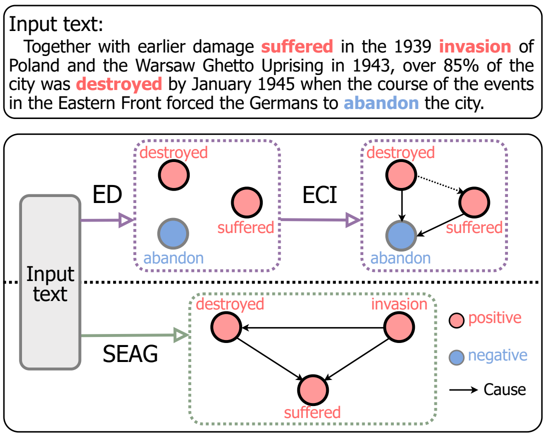


Figure 1: Comparison between SEAG and the pipelined model. SEAG completes Event Causality Extraction by text2text generation while the pipelined methods break this task into two sub-tasks, i.e., Event Detection (ED) and Event Causality Identification (ECI). Dashed arrows stand for failing to extract the correct causalities.

For the purpose of Event Causality Extraction, current methods break down the task into two subtasks, i.e. Event Detection (ED) (Chen et al., 2015; Wang et al., 2019; Lin et al., 2020a) and Event Causality Identification (ECI) (Zuo et al., 2020; Phu and Nguyen, 2021; Chen et al., 2022). Then the solution of Event Causality Extraction is integrating these two sub-tasks. Although such a pipelined method is feasible to a certain extent, two limitations deteriorate the efficacy. First, in the view of task formulation, it over-simplifies the problem as two local extraction processes with ignorance of cross-task dependencies. Such separation hinders feature and knowledge sharing when extracting events and their causal interdependence. It also can result in error propagation. As shown in Figure 1, although the event "abandon" has no causal relations with other events in the context, the

*Corresponding authors.



*Findings of the Association for Computational Linguistics: ACL 2023*, pages 4631–4644
July 9-14, 2023 ©2023 Association for Computational Linguistics

ED model still detects it neglecting the cross-task dependencies with ECI. Second, from the event causality perspective, the events and their causal relations form a global event causality graph (ECG). The pipelined models break the innate structural causality of all events in the context when extracting the events and their inter-relations without capturing the structural interactions within the ECG. For example in Figure 1, events "suffered", "invasion" and "destroyed" form an ECG. The pipelined models fail to leverage this structural information and then miss the causality between "destroyed" and "suffered" by only counting on these two events themselves. Without such structural understanding, models are likely prone to extract events and causal relations by the perception of superficial linguistic features (Wang et al., 2022a) and make biased predictions.

To bypass the absence of the task dependency brought by pipelining separation, we propose Structure-Aware Event Causality Generation (SEAG), a novel paradigm for Event Causality Extraction. Specifically, SEAG linearizes the ECG with a label semantic enhanced template and then generates the extraction results in a text2text generative format based on a generative pre-trained language model (Raffel et al., 2020). SEAG generates all events and relations in one-pass, avoiding local prediction in either ED or ECI. This end-to-end extraction can further mitigate the error propagation. Moreover, regarding the ECG as the output, SEAG enables the reasoning and interactions in the whole structure. Such a structural extraction process maintains the ECG and its semantics intact. To further improve the understanding of the ECG, we adopt Causality Structural Discrimination. We first sample negative events and relations to compose negative ECGs. Then we conduct discriminative learning to foster the model's awareness of the positive ECG. During this discriminative process, the model can learn to comprehend the in-depth event causality semantic of ECG structures, not learn only on superficial features.

We conduct experiments on three Event Causality Extraction datasets to testify to the effectiveness of SEAG. Evaluation results show that the end-to-end extraction of events and their causal relations in a text2text format is effective. Moreover, Causality Structural Discrimination further improves the model's extraction ability with a better understanding of event structure. We summarize our contributions as follows:

- We propose SEAG for Event Causality Extraction which extracts events and causal relations via end-to-end text2text generation.
- We introduce Causality Structural Discrimination to further foster the event structural understanding of our model.
- We conduct extensive experiments to test the effectiveness of our model.

## 2 Preliminaries

**Task Formulation.** Event Causality Extraction aims to extract events and their inter-causal relations from a text sequence. Formally, given an input text sequence $\mathcal{X}$ consisting of $n$ words $\mathcal{X} = [x_1, x_2, ..., x_n]$. Event $\mathcal{E}_i$ is represented as multiple consecutive words $\{x_k^i\}$. A model should extract all causal triplets $(\mathcal{E}_i, \mathcal{E}_j, \mathcal{R}_{ij}) \in \mathbb{S}$. $\mathcal{E}_i$ is the $i^{th}$ event. $\mathcal{R}_{ij}$ is the causal relation. The causal triplet $(\mathcal{E}_i, \mathcal{E}_j, \mathcal{R}_{ij})$ stands for the existence of a causal relation $\mathcal{R}_{ij}$ between $\mathcal{E}_i$ and $\mathcal{E}_j$. Conventionally, there are two analogous task settings:

- **Directed.** The model predicts the directionality of cause and effect events. The type of $R_{ij}$ is binary, i.e. $\mathcal{R}_{ij} \in \{\text{CAUSE}, \text{EFFECT}\}$[1].
- **Undirected.** The model only predicts the existence of causality between events. The type of $R_{ij}$ is unary, i.e. $\mathcal{R}_{ij} = \text{CAUSAL}$.

**Pipelined Extraction.** Pipelined methods break Event Causality Extraction into Event Detection (ED) and Event Causality Identification (ECI). They solve the task as learning the probability $P(\mathcal{R}|\mathcal{E}, \mathcal{X}) \cdot P(\mathcal{E}|\mathcal{X})$ which breaks the joint probability of $P(\mathcal{E}, \mathcal{R}|\mathcal{X})$.

**Event Causality Graph.** An Event Causality Graph (ECG) is a graph $\mathcal{G} = (\mathbb{E}, \mathbb{V})$. $\mathcal{E}_i \in \mathbb{E}$ is an event and an edge $\mathcal{V}_{ij} \in \mathbb{V}$ denotes there exists a causal relation between $\mathcal{E}_i$ and $\mathcal{E}_j$. In the Directed setting, $\mathcal{G}$ is a directed acyclic graph while in the Undirected case, $\mathcal{G}$ is an undirected graph.

**Structural Generation.** We model Event Causality Extraction as structural generation. We first compose the causal triplet set $\mathbb{S}$ as an ECG. We then learn a model $P(\mathcal{G}|\mathcal{X})$ to identify the ECG in a text2text paradigm given the context $\mathcal{X}$.

[1] CAUSE and EFFECT are symmetrical. For any $R_{ij}$, we don't augment its symmetric relation in Directed setting.

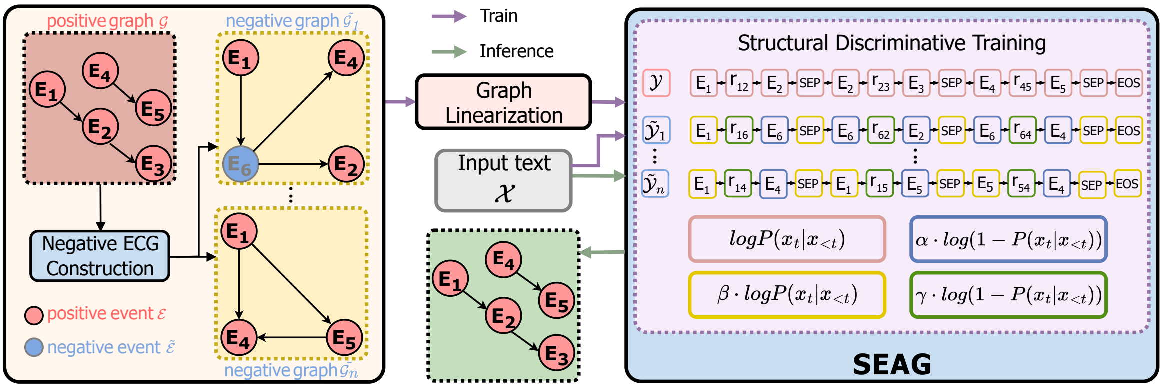


Figure 2: Overview of our SEAG. SEAG completes Event Causality Extraction by Structure-Aware Event Causality Generation. The Graph Linearization module linearizes all positive $\mathcal{G}$ and negative $\tilde{\mathcal{G}}$s into $\mathcal{Y}$ and $\tilde{\mathcal{Y}}$s. Then SEAG conducts Structural Discriminative Training to foster ECG understanding.

## 3 Method

**Model Overview.** Our SEAG first linearizes the ECG $\mathcal{G}$ with a label semantic enhanced template. Then it leverages a generative pre-trained language model to predict the results. At last, it improves the ECG structural comprehension via Causality Structural Discrimination. The overview of SEAG is in Figure 2. In the rest of this section, we first detail the graph linearization in Section 3.1. Then we elaborate the generation process in Section 3.2. Finally, we present the Causality Structural Discrimination in Section 3.3.

### 3.1 Graph Linearization

To extract the ECG $\mathcal{G}$ in a generative paradigm, the ECG should be linearized into a sequence $\mathcal{Y}$. This linearization should keep the structural information intact and be as consistent as possible with natural language characteristics.

Considering the appearance order of events in the context is a crucial feature for our task, we infuse the positional information of events in $\mathcal{Y}$. Given an ECG $\mathcal{G}$ originating from triplet set $\mathbb{S}$, we sort triplets in $\mathbb{S}$ by the head event position in $\mathcal{X}$. If two triplets have the same head event, we arrange them according to the second event.

After obtaining sorted $\mathbb{S}$[2], one possible linearization process is to iteratively compose $(\mathcal{E}_i, \mathcal{E}_j, \mathcal{R}_{ij})$ into $\mathcal{Y}$, which ends up with $\mathcal{Y} = [\mathcal{E}_1, \mathcal{E}_2, \mathcal{R}_{12}, \mathcal{E}_1, \mathcal{E}_3, \mathcal{R}_{13}, ...]$. This template is widely adopted in entity relation extraction (Giorgi et al., 2022; Guo et al., 2022). However, we find that this is not suitable for Event Causality Extraction since the relations, often verbs, between events entail dynamic information. The above template violates event causal label semantics and severely deteriorates linguistic structure. In order to inject event causal label semantics and enable making more use of the pre-trained language model, we linearize $\mathcal{G}$ as follows:

$$\mathcal{Y} = [\mathcal{E}_1, \mathcal{R}_{12}, \mathcal{E}_2, \text{SEP}, \mathcal{E}_1, \mathcal{R}_{13}, \mathcal{E}_3, \text{SEP}, ...] \,. \quad (1)$$

SEP is a separator indicator. In this template, relation words act as verbs making the sequence more fluent and close to real natural language sentences.

**Modifier Pruning.** In the real scenario, an event may consist of a core word and several modifiers (e.g. "magnitude-6.1 earthquake"). These modifiers incur noise when generating the results. Therefore, we prune all events in $\mathcal{Y}$ by only keeping the last word to represent the event.

**Natural Language Quantifier.** Another issue arises when there are events that share the same words, which can be problematic for a generation model as well. We tackle this problem by adding a natural language quantifier to distinguish duplicated event words. Supposing $\mathcal{E}_i, i = [1, 2, 3]$ have the same words, we adapt them into "first $\mathcal{E}_1$", "second $\mathcal{E}_2$", "third $\mathcal{E}_3$". In case of more duplicated numbers, the language quantifier goes on.

In sum, we linearize $\mathcal{G}$ into $\mathcal{Y}$ and denote this linearization process as $\mathcal{Y} = \textbf{\textit{Linear}}\ (\mathcal{G})$.

### 3.2 Structure-Aware Event Causality Generation

After obtaining $\mathcal{Y}$, we have adapted this task into a text2text generation $P(\mathcal{Y}|\mathcal{X})$ format. In this sec-

[2] We only consider annotated relations rather than any relations derived by transitivity.

tion, we elaborate on this generation process.

Given input $\mathcal{X}$, SEAG outputs $\mathcal{Y}$ via the generation process. This generation is modeled by pre-trained generative language model $\mathcal{M}$ such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020), which are pre-trained on large-scale corpus. SEAG first encodes the input $\mathcal{X}$ via the encoder ***Enc*** of $\mathcal{M}$. The encoding output is $\mathcal{H} = \boldsymbol{Enc}(\mathcal{X}; \theta_e)$, which is the encoding hidden states representation. SEAG then generates $\mathcal{Y}$ with decoder of $\mathcal{M}$ in an auto-regressive process.

$$P(\mathcal{Y}|\mathcal{X}) = \prod_i \boldsymbol{Dec}(\mathcal{Y}_{<i}, \boldsymbol{H}; \theta_d). \tag{2}$$

We denote the encoder and decoder parameters of $\mathcal{M}$ as $\theta_{\mathcal{M}} = (\theta_e, \theta_d)$.

**Training and Inference.** SEAG is trained on all $(\mathcal{X}, \mathcal{Y})$ pairs by log-likelihood maximization loss:

$$\mathcal{L}^G = -\sum_{(\mathcal{X},\mathcal{Y})} \log P(\mathcal{Y}|\mathcal{X}; \theta_{\mathcal{M}})\ . \tag{3}$$

To infer an answer providing input $\mathcal{X}$, SEAG first encodes it with its encoder and then generates $\mathcal{Y}$ through the beam search mechanism.

**Constraint Decoding.** During inference generation, in order to constrain not generating irrelevant words, methods solving entity extraction introduce pointer mechanism (Zeng et al., 2018, 2020), and indices-based generation (Nayak and Ng, 2020). However, to keep the decoding process neat, we only constrain the generated words to be words in $\mathcal{X}$, relation tokens $\mathcal{R}$, and separator SEP. We find this constraint is enough for SEAG to complete Event Causality Extraction.

### 3.3 Causality Structural Discrimination

Although the above generation process keeps the ECG structural information intact, only trained with generation loss $\mathcal{L}^G$ (3), the model tends to extract unfaithful ECGs (Zhu et al., 2020). The model is prone to extract events and causal relations on superficial linguistic features. The culprit is the inadequacy of structural event causality comprehension of the model.

One possible solution is to perform contrastive learning which considers a graph as a whole and aims to differentiate positive and negative graphs in a hidden space. However, not all nodes and edges in a negative graph are incorrect.

Therefore, to bypass this dilemma, we introduce Causality Structural Discrimination to improve the model's understanding of ECG. We first construct several negative the ECGs. Then we conduct a discrimination process to train the model to be aware of the positive ECG structure.

**Algorithm 1:** Negative ECG Construction.

**Input :** Positive ECG $\mathcal{G} = (\mathbb{E}, \mathbb{V})$. Input text $\mathcal{X}$. Hyper-parameters $n$ and $L$.
**Output :** Constructed negative ECG list $\tilde{\mathbb{G}}$.

$\tilde{\mathbb{G}} = [\,]$
$\tilde{\mathbb{E}} = \texttt{FindNegEvent}(\mathcal{X})$
$\mathbb{N} = \mathbb{E} \cup \tilde{\mathbb{E}}$
**for** $i \leftarrow 1$ **to** $n$ **do**
  $l = \texttt{RandomInt}(L)$
  $\{(\mathcal{E}_i, \mathcal{E}_j)\} = \texttt{SampleNegPair}(\mathbb{N},\ l)$
  $\{(\mathcal{E}_i, \mathcal{E}_j, \mathcal{R}_{ij})\} = \texttt{AssignRel}(\mathbb{N},\ l)$
  **foreach** $(\mathcal{E}_i, \mathcal{E}_j, \mathcal{R}_{ij})$ **do**
    Assert $(\mathcal{E}_i, \mathcal{E}_j, \mathcal{R}_{ij}) \notin \mathbb{V}$
  **end foreach**
  $\tilde{\mathcal{G}} = \texttt{Compose}(\{(\mathcal{E}_i, \mathcal{E}_j, \mathcal{R}_{ij})\})$
  $\tilde{\mathbb{G}}.\texttt{Append}(\tilde{\mathcal{G}})$
**end for**
**return** $\tilde{\mathbb{G}}$

**Negative ECG Construction.** Given the positive event node set $\mathbb{E}$ in a true $\mathcal{G}$, we first pre-extract the negative event node set $\tilde{\mathbb{E}}$ via a linguistic toolkit. We denote the total event set as $\mathbb{N} = \mathbb{E} \cup \tilde{\mathbb{E}}$. After that, we sample negative event pairs from $\mathbb{N}$ and assign them random relations but guarantee they are not the positive edges in $\mathbb{V}$, i.e. $\{(\mathcal{E}_i, \mathcal{E}_j) | \ (\mathcal{E}_i, \mathcal{E}_j) \notin \mathbb{V}\}$. Then we compose these negative edges as a negative ECG $\tilde{\mathcal{G}}$.

We repeat the above negative construction process $n$ times, to obtain $n$ negative ECGs. Each time, the number of sampled event pairs is different. We randomize this number between 1 and a maximum threshold of $L$. Formally, The Negative ECG Construction is shown as Algorithm 1.

**Structural Discriminative Training.** After acquiring all $\tilde{\mathcal{G}}$s, we apply the same linearization process in Section 3.1 to linearize the $\tilde{\mathcal{G}}$s. Then we get negative sequences $\tilde{\mathcal{Y}} = \boldsymbol{Linear}(\tilde{\mathcal{G}})$. We next propose to train the model to be able to distinguish negative ECG $\tilde{\mathcal{G}}$ which equals to minimize the probability of $\tilde{\mathcal{Y}}$.

However, adopting structural discriminative training upon a generative model is not simple since not all parts of $\tilde{\mathcal{G}}$ are negative. Considering the $\tilde{\mathcal{G}}_1$ shown in Figure 2, $(\mathcal{E}_1, \mathcal{E}_6)$ is a negative edge while $\mathcal{E}_1$ is a positive event. One simple solution is to minimize the probability of this edge. Notice SEAG is trained in an auto-regressive way. This solution may confuse the model when adding probability reduction to $\mathcal{E}_1$. The sub-sequence till the step of $\mathcal{E}_1$ is the same as that of positive $\mathcal{Y}$ since $\mathcal{E}_6$ has never shown up yet.

We solve this dilemma by designing the structural discriminative training. For a negative ECG $\tilde{\mathcal{G}}$, we assign different optimization objectives to each token after linearization. Considering an edge $(\mathcal{E}_i, \mathcal{E}_j)$ from $\tilde{\mathcal{G}}$, according to the linearization in Section 3.1, it results in a sub-sequence of $\tilde{\mathcal{Y}}_{[u:u+4]} = [\mathcal{E}_i, \mathcal{R}_{ij}, \mathcal{E}_j, \text{SEP}]$. If $\mathcal{E}_i$ is a negative event, we reduce the probability of both events:

$$
\begin{aligned}
&if \;\; \mathcal{E}_i \notin \mathbb{E}: \\
&\quad D(\mathcal{E}_i) = -\,\alpha \cdot log(1 - P(\mathcal{E}_i | \tilde{\mathcal{Y}}_{<u})) \\
&\quad D(\mathcal{E}_j) = -\,\alpha \cdot log(1 - P(\mathcal{E}_j | \tilde{\mathcal{Y}}_{<u+2}))\;.
\end{aligned} \tag{4}
$$

If $\mathcal{E}_i$ is a positive event while $\mathcal{E}_j$ is a negative event or if $\mathcal{E}_i$ and $\mathcal{E}_j$ are both positive events but there's no path between $\mathcal{E}_i$ and $\mathcal{E}_j$ in $\mathcal{G}$:

$$
\begin{aligned}
&if \;\; \mathcal{E}_i \in \mathbb{E} \wedge (\mathcal{E}_j \notin \mathbb{E} \;\vee not\; \textbf{\textit{HasPath}}(\mathcal{E}_i, \mathcal{E}_j, \mathcal{G})): \\
&\quad D(\mathcal{E}_i) = -\,\beta \cdot log(P(\mathcal{E}_i | \tilde{\mathcal{Y}}_{<u})) \\
&\quad D(\mathcal{E}_j) = -\,\alpha \cdot log(1 - P(\mathcal{E}_j | \tilde{\mathcal{Y}}_{<u+2}))\;,
\end{aligned} \tag{5}
$$

where $\textbf{\textit{HasPath}}(\cdot)$ is a function to find whether there exists a path between two nodes in a graph[3]. The motivation here is to train the model to be aware of ECG structural semantics. Firstly, non-connected events or negative events entail no causality. Secondly, since causality has the property of transitivity, there should exist a causal relation between events linked by a path even if they are not directedly connected. This discriminative learning injects the causality structural knowledge into our model. We treat $\mathcal{R}_{ij}$ and SEP tokens as:

$$
\begin{aligned}
&D(\mathcal{R}_{ij}) = -\,\gamma \cdot log(1 - P(\mathcal{R}_{ij} | \tilde{\mathcal{Y}}_{<u+1})) \\
&D(\text{SEP}) = -\,\beta \cdot log(P(\text{SEP} | \tilde{\mathcal{Y}}_{<u+3}))\;.
\end{aligned} \tag{6}
$$

where $\alpha$, $\beta$ and $\gamma$ are hyper-parameters to control the structural discriminative learning.

We conduct the same optimization for the rest of $\tilde{\mathcal{Y}}$. Therefore, the structural discriminative learning for $\tilde{\mathcal{G}}$ is $\textbf{\textit{Dis}}(\tilde{\mathcal{G}}) = \sum_t D(\tilde{\mathcal{Y}}_t)$. The final structural discriminative training loss is computed over all constructed negative ECGs. Then we conduct multi-task training to train SEAG with $\mathcal{L}^G$ and $\mathcal{L}^D$

$$
\mathcal{L} = \mathcal{L}^G + \mathcal{L}^D, \;\; \mathcal{L}^D = \sum_i \textbf{\textit{Dis}}(\tilde{\mathcal{G}}_i). \tag{7}
$$

[3]In Undirected setting, deciding whether there's a path between two nodes equals to determine if these two nodes are in the same component.

## 4 Experiments

This section first gives the datasets we want to use in Section 4.1. We elaborate on the evaluation metrics for Event Causality Extraction in Section 4.2. The baselines and the implementation details are in Section 4.3 and 4.4 respectively. We finally report the experimental results in Section 4.5.

### 4.1 Datasets

**EventStoryLine.** It's a wildly ECI dataset. It contains 258 documents, 22 topics, 5,334 events, and 1,770, and 3,885 intra- and inter-causal relation pairs (Caselli and Vossen, 2017). Following Gao et al. (2019), we take event pairs annotated with 'FALLING_ACTION' as CAUSE relation and 'PRECONDITION' as EFFECT. We conduct both Undirected and Directed settings on this dataset. For both settings, we use documents from topics 37 and 41 as the validation set and leave the rest to perform 5-fold cross-validation.

**MAVEN-ERE.** This is the newest Event Relation Extraction dataset, including causal, temporal, and sub-event relation types (Wang et al., 2022b). It contains 4,480 documents, 103,193 events, and 57,992 causal relation pairs. The causal event pairs are annotated by 'CAUSE' or 'PRECONDITION', which are both for the CAUSE relation. Therefore, we only conduct the Undirected setting in this dataset and take triplets annotated with 'CAUSE' and 'PRECONDITION' as gold data. Since this dataset has not published its test set, we conduct in-house validation. We sample 10% of the data from the original training set as the validation set and leave the rest as the training set. We use the original validation set as the test set.

**SCITE.** This is a CAUSE-EFFECT span detection dataset by extending the annotations of more causal triplets in the SemEval 2010 task 8 dataset (Li et al., 2021). We conduct both Undirected and Directed settings on this dataset.

To handle documents that are longer than the maximum allowed length for T5, we split the documents in both EventStoryLine and MAVEN-ERE via the following method: we identify the two sentences that contain the starting and ending events and gather the sentences in between them. These sentences are all used as the context for the event triplet. All the event triplets of the same sentences are grouped together, and each group is treated as a single data point. So each data point in our dataset

| | Undirected | | | Directed | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| PIPELINED MODEL | | | | | | |
| DB+BERT (Wang et al., 2019) | 23.03 ± 4.36 | 26.73 ± 4.76 | 24.51 ± 3.80 | 27.90 ± 3.59 | 16.78 ± 3.84 | 20.69 ± 2.86 |
| DB+LONG (Beltagy et al., 2020) | 28.26 ± 2.95 | 34.15 ± 5.93 | 30.86 ± 4.13 | 28.33 ± 2.79 | 17.53 ± 4.88 | 21.37 ± 3.99 |
| DB+ERGO (Chen et al., 2022) | 25.76 ± 3.27 | **34.76 ± 7.88** | 29.46 ± 4.86 | 25.01 ± 2.95 | 20.90 ± 3.97 | 22.53 ± 2.49 |
| GENERATIVE MODEL | | | | | | |
| Seq2Rel (Giorgi et al., 2022) | 32.26 ± 6.09 | 24.24 ± 3.34 | 27.63 ± 4.42 | 25.17 ± 5.58 | 18.78 ± 3.43 | 21.47 ± 4.29 |
| SEAG (Ours) | **37.98 ± 8.48** | 32.33 ± 6.14 | **34.85 ± 7.10** | **31.69 ± 6.93** | **23.30 ± 4.44** | **26.77 ± 5.31** |

Table 1: Results on **EventStoryLine** dataset on both settings. We report the average and standard deviation scores on conducting 5-folds cross-validation. Bold numbers represent the highest scores.

| | Undirected | | | Directed | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| PIPELINED MODEL | | | | | | |
| DB+BERT (Wang et al., 2019) | 51.03 | 87.83 | 64.55 | 73.14 | 66.72 | 69.78 |
| DB+LONG (Beltagy et al., 2020) | 51.90 | 84.96 | 64.45 | 70.87 | 67.39 | 69.09 |
| DB+ERGO (Chen et al., 2022) | 85.15 | **92.06** | 88.47 | 85.16 | 66.89 | 74.92 |
| JOINT MODEL | | | | | | |
| SCI (Li et al., 2021) | - | - | - | 83.33 | **85.81** | 84.55 |
| GENERATIVE MODEL | | | | | | |
| Seq2Rel (Giorgi et al., 2022) | 86.37 | 81.41 | 83.82 | 88.34 | 79.39 | 83.62 |
| SEAG (Ours) | **91.78** | 90.54 | **91.60** | **90.68** | 85.47 | **88.00** |

Table 2: Results on **SCITE** datasets on both settings. Bold numbers represent the highest scores.

is successive sentences, not the whole document. We testify our method and all baselines under this setting.

## 4.2 Evaluation Metrics

We use precision (P), recall (R) and F1-score ($F_1$) as the evaluation metrics:

$$\mathsf{P} = \frac{\mathsf{TP}}{\mathsf{TP}+\mathsf{FP}}, \mathsf{R} = \frac{\mathsf{TP}}{\mathsf{TP}+\mathsf{FN}}, \mathsf{F}_1 = 2 \cdot \frac{\mathsf{P}\cdot\mathsf{R}}{\mathsf{P}+\mathsf{R}},$$

TP, FP and FN are all computed on event triplets. In the Undirected setting, we count an extracted event causal triplet as a *True Positive* triplet if it has exactly matched two events (regardless of distinguishing cause and effect). In Directed setting, we additionally require the extracted triplet to have the same cause and effect events as the gold triplet.

## 4.3 Baselines

**DB+BERT** (Wang et al., 2019; Devlin et al., 2019). This is a typical pipelined event causality extraction baseline. The system first detects events via DMBERT and then identifies the inter causal relations by BERT-base. For Event Causality Identification, we concatenate two event trigger representations from the context encoded by BERT, then classify the relation type by a ***MLP*** layer.

**DB+LONG** (Wang et al., 2019; Beltagy et al., 2020). This pipelined method is the same as the previous one except we replace the backbone with Longformer-base (Beltagy et al., 2020) for Event Causality Identification.

**DB+ERGO** (Wang et al., 2019; Chen et al., 2022). The system detects events via DMBERT and then identifies the inter causal relations by the SOTA Event Causality Identification model ERGO. We implement ERGO based on BERT-base.

**Seq2Rel** (Giorgi et al., 2022) This is a generative entity triplet extraction model. We directly adapt it to Event Causality Extraction and implement it on T5-base for fair comparison.

**SCI** (Li et al., 2021) This is a joint Cause-Effect extraction model. SCI models Cause-Effect extraction as a BIO tagging task and proposes a multi-head self-attention mechanism. Then it aligns the extracted results via a tag2triplet algorithm.

| | Undirected | | | Directed | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| SEAG (Ours) | 37.98 ± 8.48 | 32.33 ± 6.14 | 34.85 ± 7.10 | 31.69 ± 6.93 | 23.30 ± 4.44 | 26.77 ± 5.31 |
| SEAG w.o. CSD | 36.26 ± 7.92 | 29.68 ± 4.96 | 32.56 ± 6.17 | 28.91 ± 7.05 | 21.45 ± 4.02 | 24.59 ± 5.22 |
| SEAG w.o. Event Ordering | 38.25 ± 6.23 | 25.39 ± 3.70 | 30.47 ± 4.52 | 32.06 ± 8.09 | 21.01 ± 5.37 | 25.36 ± 6.43 |
| SEAG w.o. Modifier Pruning | 35.26 ± 7.37 | 31.63 ± 5.51 | 33.27 ± 6.33 | 28.33 ± 5.48 | 24.02 ± 4.42 | 25.90 ± 4.70 |

Table 3: Ablation study on EventStoryLine of both settings. We report the average and standard deviation scores on conducting 5-fold cross-validation.

| MAVEN-ERE | | | |
|---|---|---|---|
| | P | R | $F_1$ |
| PIPELINED MODEL | | | |
| DB+BERT (Wang et al., 2019) | 43.91 | 41.31 | 42.57 |
| DB+LONG (Beltagy et al., 2020) | 42.49 | 41.69 | 42.09 |
| DB+ERGO (Chen et al., 2022) | 45.48 | 40.79 | 43.01 |
| GENERATIVE MODEL | | | |
| Seq2Rel (Giorgi et al., 2022) | 47.13 | 49.25 | 48.17 |
| SEAG (Ours) | **49.28** | **50.57** | **49.92** |

Table 4: Results of **MAVEN-ERE** on the Undirected setting. Bold numbers represent the highest scores.

### 4.4 Implementation Details

We use T5-base (Raffel et al., 2020) as the backbone model. We use AdamW optimizer with 5e-5 learning rate. We apply linear weight decay. The batch size is 8. We train all models until epoch 20 and select the epoch that performs best on the validation set for the test. We don't use warm-up and label smoothing tricks. We implement all the experiments on Tesla V100 GPU.

For EventStoryLine, we conduct a grid search for threshold hyper-parameters and find $n = 10$ and $L = 2$ work the best. In the same way, we find $n = 5$ and $L = 1$ in SCITE and $n = 3$ and $L = 1$ in MAVEN-ERE are appropriate. We find $\alpha = 1$, $\beta = \gamma = 0$ suits all three datasets.

We leverage Spacy[4] to extract all verbs and nouns which are not the positive events as the negative event set $\tilde{\mathbb{E}}$. We conduct pilot experiments and find using the word "next" as SEP works well. As well, using relation tokens as $\mathcal{R}_{ij} \in \{\text{CAUSE}, \text{EFFECT}\}$ in directed setting and $\mathcal{R}_{ij} = \text{CAUSAL}$ in undirected setting effects better.

### 4.5 Evaluation Results

The results of SEAG on EventStoryLine, MAVEN-ERE, and SCITE are shown in Table 1, Table 2, and Table 4 respectively. SEAG outperforms all pipelined models in $F_1$ score on all three datasets of two settings. The superior performance demonstrates extracting event causality by our structure-aware event causality generation effects. SEAG can handle the cross-task dependencies and maintain the ECG structures intact.

Based on the results, we find that SEAG performs better than Seq2Rel. The results first testify the strength of our graph linearization process. This linearization process accounts for event causality semantics and the event dynamic property. Second, the results confirm the benefits of using the suggested Causality Structural Discrimination training. SEAG comprehends the ECG better and can distinguish the positive ECG from negative ones. This ECG semantic understanding of SEAG hinges generates better predictions.

We notice that the gains of SEAG come more from precision scores. That is because SEAG models the cross-task dependencies and filters the events of false causal relations. SEAG maintains the semantic of ECGs and can extract more correct answers which aligns with our intuition.

### 4.6 Discussions

**Ablations.** We conduct ablation studies on the EventStoryLine dataset of both Undirected and Directed settings. We list the results in Table 3. SEAG w.o. CSD stands for SEAG without Causality Structural Discrimination. In SEAG w.o. Event Ordering, we shuffle event orders and then compose them into the generative template. SEAG w.o. Modifier Pruning is the model in which we don't prune event modifiers. The results indicate the effectiveness of Causality Structural Discrimination in both Undirected and Directed settings. SEAG is enhanced by event causality structural knowledge via this discrimination process. Event Ordering is crucial for SEAG especially in the Undirected setting which shows that the generative extraction takes advantage of the ability of sequential ranking

[4] https://spacy.io/

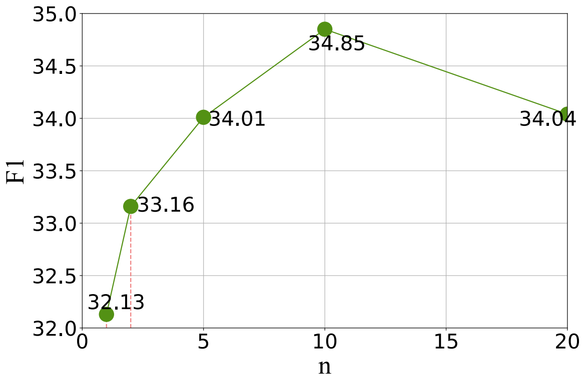


Figure 3: Analysis of number of negative ECGs ($n$) on EventStoryLine of Undirected setting.

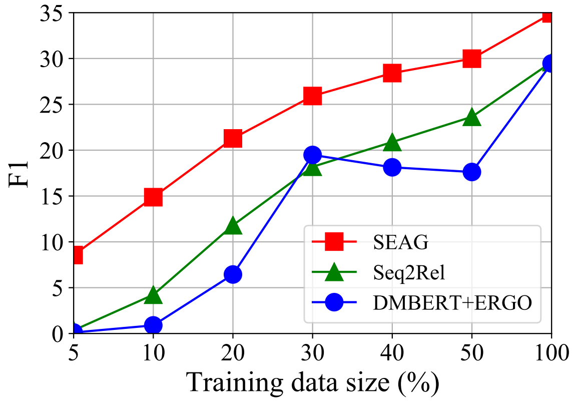


Figure 4: Low-resource Analysis on EventStoryLine of Undirected setting.

to some extent. Finally, modifier pruning benefits extraction, as training becomes more challenging when events have a large number of words.

**Number of Negative ECGs.** We conduct experiments on EventStoryLine to inspect the influence of the number of negative ECGs $n$. The results are given in Figure 3. The performance improves greatly when $n$ increases from 1 to 3 and remains relatively stable until $n$ reaches 10. After that, the performance decreases when $n$ becomes larger. The results provide evidence of the effectiveness of our causality structural discrimination training method. Additionally, the results show that, for most of the datasets, a small number of ECG are sufficient for the Causality Structural Discrimination training. However, there are still some cases where more ECGs are needed.

**Low-resource Analysis.** To test SEAG's ability to work with limited resources, we conduct experiments on training models using a subset of the EventStoryLine data. As shown in Figure 4, SEAG performs better in low-resource scenery than other baselines. When there's only 5% data, Seq2Rel and DB+ERGO fail to extract events and inter-causal relations while SEAG can still identify causality triplets. With the increase of the data sizes, all models get better performances and SEAG outperforms other models in all percentages of the training set. The results demonstrate our intuitions that structure-aware event causality generation can better take advantage of generative pretrained language models. For the same reason, we notice that the Seq2Rel is better than DB+ERGO when there are only very limited data. Besides, our causality structural discrimination training enables SEAG to distinguish from negative structures even with only a small amount of data.

## 5 Related Works

**Event Causality Extraction** Current methods for Event Causality Extraction mainly break this task into Event Causality Identification (ECI) and Event Detection (ED). For ECI, Kadowaki et al. (2019) leverages the pre-trained language model to grasp the annotator's policy. Liu et al. (2021); Cao et al. (2021) incorporate external event knowledge. Tan et al. (2021) augments dataset via generation of counterfactual causal sentences. Zuo et al. (2021a) enhances extracting performance by introducing causal statements. Zuo et al. (2021b) generates synthetic data via a dual-learning framework. Phu and Nguyen (2021) builds document-level graphs and encodes them via a graph neural network. Chen et al. (2022) formulates ECI as a node classification task. Liu et al. (2023) constructs a prompt to inject event knowledge. For ED, Wang et al. (2019) performs weak supervision by applying an adversarial training mechanism. Liu et al. (2016) builds a semi-supervised corpus on FrameNet text. Yang et al. (2019) performs distant supervision via Freebase, Wikipedia, and FrameNet. Huang and Peng (2021) proposes a method to model document-level structures. Xu et al. (2021) proposes a Graph-based method to capture the global interaction between entities in a document. Luan et al. (2019) employs an interactive graph-based propagation between events and entities. Lin et al. (2020b) enforces global constraints to the final extraction results. Nguyen et al. (2022) induces a cross-task dependency graph to boost representation learning. Among these methods, we are the first to extract an ECG by end-to-end generation.

**Cause-Effect Span Detection** Cause-Effect Span Detection is to detect the spans of two units where the "cause" is the producer of the "effect". Existing methods model this task as sequence labeling. Dasgupta et al. (2018) uses word-level embeddings and some linguistic features to detect causes and effects. Li et al. (2021) proposes to use Bi-LSTM-CRF with Flair Embeddings. Tan et al. (2022) introduces a news corpus with an annotation schema breaking out the restrictions that only explicit relations apply. The main difference between our work and these methods is that SEAG extracts an ECG consisting of a set of events with their structural inter causal relations rather than identifying two boundaries of "cause" and "effect".

**Generative Triplet Extraction** While previous methods extract information jointly (Li et al., 2022) in processes of discrimination, current research also explores using a generative paradigm to solve triplet extraction tasks. Zeng et al. (2018, 2020) introduce copy mechanism into entity relation extraction. Cabot and Navigli (2021); Lu et al. (2022) utilize structural knowledge and label semantics with generative formats. Nayak and Ng (2020) designs indices-based generation for entity relation extraction. Chia et al. (2022) proposes to train with synthetic data in a generative way. Compared to existing generative triplet extraction approaches, we propose to extract the whole ECG structure of the context, which requires a global semantic understanding of all events and their causal relations.

## 6 Conclusion

We propose a novel Structure-Aware Event Causality Generation (SEAG) for Event Causality Extraction. We model this task as structural generation and design the novel ECG linearization. We also adopt the Causality Structural Discrimination training to foster the model's understanding of the ECG. We conduct experiments on two settings of three datasets. Results demonstrate that SEAG outperforms the pipelined models for Event Causality Extraction on all datasets.

## 7 Acknowledgement

Our work is supported by the National Key Research and Development Program of China (Project Number: 2020AAA0109400). we kindly appreciate all the researchers who provide valuable insights, discussions, and comments on this work.

## Limitations

As shown in Table 1, 2 and 4, although SEAG outperforms the pipelined models, there is still a gap in performances between EventStoryLine, MAVEN-ERE and SCITE. The performance on SCITE is relatively high than EventStoryLine and MAVEN-ERE. This shows that our model suffers in extracting implicit event causality compared to explicit ones. One potential way to deal with this issue could be introducing an in-context prompt for such relation extraction. We leave the modules for implicit event causality extraction for future work.

## References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1499–1510.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

Pengfei Cao, Xinyu Zuo, Yubo Chen, Kang Liu, Jun Zhao, Yuguang Chen, and Weihua Peng. 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 (Volume 1: Long Papers)*, pages 4862–4872.

Tommaso Caselli and Piek Vossen. 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*, pages 77–86, Vancouver, Canada. Association for Computational Linguistics.

Meiqi Chen, Yixin Cao, Kunquan Deng, Mukai Li, Kun Wang, Jing Shao, and Yan Zhang. 2022. ERGO: Event relational graph transformer for document-level event causality identification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2118–2128, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association* *for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. Relationprompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57.

Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, and Abir Naskar. 2018. Automatic extraction of causal relations from text using linguistically informed deep neural networks. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 306–316.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Lei Gao, Prafulla Kumar Choubey, and Ruihong Huang. 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, Volume 1 (Long and Short Papers)*, pages 1808–1817, Minneapolis, Minnesota. Association for Computational Linguistics.

John Giorgi, Gary Bader, and Bo Wang. 2022. A sequence-to-sequence approach for document-level relation extraction. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 10–25.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 76–83.

Qipeng Guo, Yuqing Yang, Hang Yan, Xipeng Qiu, and Zheng Zhang. 2022. Dore: Document ordered relation extraction based on generative framework. *arXiv preprint arXiv:2210.16064*.

Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, István Varga, Jong-Hoon Oh, and Yutaka Kidawara. 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)*, pages 987–997, Baltimore, Maryland. Association for Computational Linguistics.

Matthew Ho, Aditya Sharma, Justin Chang, Michael Saxon, Sharon Levy, Yujie Lu, and William Yang Wang. 2022. Wikiwhy: Answering and explaining cause-and-effect questions. *arXiv preprint arXiv:2210.12152*.

Kung-Hsiang Huang and Nanyun Peng. 2021. Document-level event extraction with efficient end-to-end learning of cross-event dependencies. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 36–47, Virtual. Association for Computational Linguistics.

Kazuma Kadowaki, Ryu Iida, Kentaro Torisawa, Jong-Hoon Oh, and Julien Kloetzer. 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)*, pages 5816–5822, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Jia Li, Yuyuan Zhao, Zhi Jin, Ge Li, Tao Shen, Zhengwei Tao, and Chongyang Tao. 2022. Sk2: Integrating implicit sentiment knowledge and explicit syntax knowledge for aspect-based sentiment analysis. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1114–1123.

Zhaoning Li, Qi Li, Xiaotian Zou, and Jiangtao Ren. 2021. Causality extraction based on self-attentive bilstm-crf with transferred embeddings. *Neurocomputing*, 423:207–219.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020a. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020b. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, and Jun Zhao. 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*, pages 3608–3614.

Jintao Liu, Zequn Zhang, Zhi Guo, Li Jin, Xiaoyu Li, Kaiwen Wei, and Xian Sun. 2023. Kept: Knowledge enhanced prompt tuning for event causality identification. *Knowledge-Based Systems*, 259:110064.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging FrameNet to improve automatic event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2134–2143, Berlin, Germany. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8528–8535.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4363–4374, Seattle, United States. Association for Computational Linguistics.

Minh Tran Phu and Thien Huu Nguyen. 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*, pages 3480–3490.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Fiona Anting Tan, Devamanyu Hazarika, See Kiong Ng, Soujanya Poria, and Roger Zimmermann. 2021. Causal augmentation for causal sentence classification. In *Proceedings of the First Workshop on Causal Inference and NLP*, pages 1–20.

Fiona Anting Tan, Ali Hürriyetoğlu, Tommaso Caselli, Nelleke Oostdijk, Tadashi Nomoto, Hansi Hettiarachchi, Iqra Ameer, Onur Uca, Farhana Ferdousi Liza, and Tiancheng Hu. 2022. The causal news corpus: Annotating causal relations in event sentences from news. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2298–2310, Marseille, France. European Language Resources Association.

Haoyu Wang, Hongming Zhang, Yuqian Deng, Jacob R Gardner, Muhao Chen, and Dan Roth. 2022a. Extracting or guessing? improving faithfulness of event temporal relation extraction. *arXiv preprint arXiv:2210.04992*.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, et al. 2022b. Maven-ere: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. *arXiv preprint arXiv:2211.07342*.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 998–1008.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3533–3546, Online. Association for Computational Linguistics.

Linyi Yang, Zhen Wang, Yuxiang Wu, Jie Yang, and Yue Zhang. 2022. Towards fine-grained causal reasoning and qa. *arXiv preprint arXiv:2204.07408*.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 9507–9514.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2020. Boosting factual correctness of abstractive summarization with knowledge graph. *arXiv preprint arXiv:2003.08612*.

Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng, and Yuguang Chen. 2021a. Improving event causality identification via self-supervised representation learning on external causal statement. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2162–2172.

Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng, and Yuguang Chen. 2021b. Learnda: Learnable knowledge-guided data augmentation for event causality identification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3558–3571.

Xinyu Zuo, Yubo Chen, Kang Liu, and Jun Zhao. 2020. Knowdis: Knowledge enhanced data augmentation for event causality detection via distant supervision. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1544–1550.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Section Limitations*

☑ A2. Did you discuss any potential risks of your work?
*This work could be used for some business behaviors.*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*Section Abstract and Introduction(Section 1).*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☒ Did you use or create scientific artifacts?

*Left blank.*

☐ B1. Did you cite the creators of artifacts you used?
*No response.*

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*No response.*

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*No response.*

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*No response.*

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*No response.*

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*No response.*

### C ☑ Did you run computational experiments?

*Section 4.*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Section 4.4.*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*Section 4.4.*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*Section 4.5, 4.6.*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Section 4.4.*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**
*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*No response.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*No response.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*No response.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*No response.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*No response.*