Combining Heterogeneous Structures for Event Causality Identification

Amir Pouran Ben Veyseh1, Franck Dernoncourt2, Thien Huu Nguyen1,3

1Department of Computer Science, University of Oregon, Eugene, OR, USA
2Adobe Research, Seattle, WA, USA
3VinAI Research, Vietnam

apouranb@cs.uoregon.edu, dernonco@adobe.com, thien@cs.uoregon.edu

Abstract
Event Causality Identification (ECI) is the task of recognizing causal relationship between events mentioned in texts. Due to its applications, ECI has been extensively explored by the Information Extraction community. However, the existing works are limited to either sentence-level ECI or they employ limited word interactions/structures at document level. As such, we propose a novel and effective method to comprehensively model contextual structures for ECI at three different levels, i.e., syntax, semantics, and background knowledge. Specifically, the contextual structures are integrated at different levels of the input encoder. Structure-aware representations are also combined using the graph transformer architecture to induce richer representations for ECI. We extensively evaluate the proposed model on two different benchmark datasets. The experiments show the effectiveness of the proposed method by establishing new SOTA performance.

Index Terms: Information Extraction, Even Causal Identification, Document-level Structures

1. Introduction
One of the important subtasks of an Information Extraction pipeline is Event Understanding. In the IE community [1], an Event is defined as an incident whose occurrence changes the state of real-world entities. For instance, in the sentence “After the earthquake, most of the houses in the village have been destroyed”, the two words “earthquake” and “destroyed” refer to two event mentions. Given two events mentioned in a piece of text, the Event Causality Identification (ECI) task aims to recognize causal relationship between the events. For instance, in the given example, the first event “earthquake” causes the second event “destroyed”. ECI is necessary for constructing a storyline of events mentioned in texts and could be directly employed in downstream applications including Question Answering [2], machine reading comprehension [3], and event forecasting [4].

Given the importance and applications of ECI, in the literature, several methods have been proposed. The majority of the existing works are limited to the sentence level setting [5, 6, 7, 8, 9, 10, 4]. In this setting, both event mentions appear in the same sentence. Although the existing methods for this setting are effective, they cannot effectively encode causal relations between event mentions that are in different sentences. To address this limitation, recent methods have employed contextual information at the document level [11, 12, 13]. Such methods have shown that employing structural information at the document level is effective to identify the causal relations between event mentions [12]. However, these methods fail to encode any interaction between structures. In particular, the syntax and semantics-based structures are often separately employed to encode input document context and the separate structure-aware representations are simply summed in the end. Due to the simplicity of the structural applications in these methods, they cannot encode sophisticated communications between structures to enrich representation learning.

To address the limitations in the current document-level ECI models, we propose a novel and effective method to encode the interactions between different structures in the input document. In particular, the word interactions are encoded in three different levels: (1) Syntax: In this level, the dependency trees of the sentences, connected via their roots, are employed; (2) Semantics: To obtain the semantic level word interactions, the document level Abstract Meaning Representation (AMR) are used; and (3) Background Knowledge: In order to encode the contextual information that is not directly present in the document, we employ ConceptNet [14] representations of the words, encoded in a dense graph. Unlike prior works that simply use structures separately to encode input context, we propose to employ these structures on two different levels: (A) Document Encoding: In the literature, it has been shown that transformer-based language models, e.g., BERT, encodes the syntactic and semantic information at different layers [15, 16]. Given this observation, in this work, we propose to explicitly encourage the representations obtained at specific levels to be aware of selected document structures. In particular, the early layers of the document encoder are encouraged to encode syntactic structures while the upper layers are enforced to encode semantics and background knowledge; (B) Heterogeneous structural paths: Although integrating different structures into the document encoder could improve the interactions between these graphs, it cannot guarantee modeling of heterogeneous paths between words, i.e., communications between words that involve more than two types of structures. In order to encode this type of communication, we propose to employ Graph Transformer (GT) [17]. Specifically, the adjacency matrices of the different structures are employed by GT to encode heterogeneous paths of arbitrary lengths.

We conduct extensive experiments on two benchmark datasets to evaluate the effectiveness of the proposed method. The results show that compared to the strong baselines, the proposed model achieves state-of-the-art results on both datasets.

2. Model
In this work, we formulate the task of ECI as a binary classification problem. Concretely, the input to the model is the document $D = [w_1, w_2, \ldots, w_n]$ consisting of $n$ words and $k$ sentences and the two event mentions $w_a$ and $w_b$. The gold label for the document $D$ is $y_{ab}$ which is 1 if there is a causal relation between event mentions $w_a$ and $w_b$. The goal is to pre-
dict if there is any causal relation between events \( w_s \) and \( w_t \), i.e., \( \hat{y}_d \).

Our proposed model for this task consists of three major components: (1) Document Encoder: This component consumes the input sequence of words, i.e., \( D \), to generate context-based representations for the words \( w_i \in D \). In this work, we employ transformer-based language models, i.e., BERT [18] or LongFormer [19], to encode the input text. As mentioned in the introduction, to facilitate the encoding of different types of interactions between words in the input document \( D \), we propose to enforce different types of structures at specific layers of the input encoder. In particular, the syntax, semantics, and background knowledge information are imposed on selected layers of the document encoder; (2) Graph Fusion: To encode rich interactions between different types of document structures, we propose to encode heterogeneous paths between words. Concretely, the structure-aware representations of the words \( w_i \in D \) are consumed by a graph fusion component, i.e., Graph Transformer (GT). Note that, for this component, in addition to the syntax, semantics, and background knowledge, we also introduce discourse-based structure, i.e., entity co-references. (3) Prediction: The representations obtained from GT are finally consumed by the event causality prediction component. This component consumes the representations of the events \( w_s \) and \( w_t \) and the document \( D \) to generate the label probability \( P \). The rest of this section provides details on these components.

2.1. Document Encoder

The input document \( D = [w_1, w_2, \ldots, w_n] \) is first encoded by a transformer-based language model \( M \) to obtain the vector representations \( E = [e_1, e_2, \ldots, e_n] \). In this work, we employ BERT or Longformer for this purpose\(^{1}\), both of which have 12 layers of self-attention. In these models, each layer \( l \) in the encoder employs a multi-head self-attention mechanism to compute pair-wise importance scores between every pair of words. Specifically, the attention matrix \( A_l^{h,i,j} \) for the head \( h \) at layer \( l \) is computed by \( A_l^{h,i,j} = \sigma(Q_l^{h,i,j} K_l^{h,i,j} / d) \), where \( Q_l^{h,i,j} \) and \( K_l^{h,i,j} \) are the query and key matrices, \( \sigma \) is the softmax activation function and \( d \) is the normalization factor. On the other hand, in the literature, it has been shown that early layers in transformer-based language models such as BERT encode syntactic structures while the upper layers provide more semantic-based information. Given this observation, we argue that these structures could best be integrated into the representations \( E \) if explicitly provided to the language model \( M \). To this end, we propose to exploit the attention matrices \( A_l \) in different layers of the encoder \( M \) to be consistent with selected structures (i.e., syntax, semantics, and background knowledge). In this section, we first describe how the document structures are obtained, next we explain the details of how the structures are provided to the encoder \( M \).

**Syntax:** To obtain the syntactic structure of the document \( D \), we employ dependency trees of the sentences in \( D \). Note that the root of the dependency trees of the different sentences is connected to each other to create a connected graph. Formally, the syntactic graph is represented by the adjacency matrix \( A_{syn} \) where \( A_{syn}^{i,j} \) is 1 if there is a dependency edge between words \( w_i \) and \( w_j \).

**Semantics:** To represent the semantic structure of the document \( D \), we propose to employ Abstract Meaning Representations (AMR) of the document. The AMR graph is obtained using an off-the-shelf AMR parser\(^2\) and it is represented by the sparse matrix \( A_{sem} \) where \( A_{sem}^{i,j} \) is 1 if the words \( w_i \) and \( w_j \) are connected to each other in AMR graph.

**Background Knowledge:** In addition to the syntax and semantics of the document \( D \), commonsense knowledge is also necessary to infer the interactions between words in the input document. As such, to introduce this information to the encoder, we propose to use the background knowledge of the words of the document \( D \). Specifically, we employ embeddings of the words obtained from the ConceptNet knowledge graph. Concretely, the pairwise similarity between words \( w_i \) and \( w_j \) is computed as dot product between the corresponding embeddings \( v_i \) and \( v_j \) obtained from ConceptNet embeddings: \( A_{BK}^{i,j} = v_i \odot v_j \).

2.2. Integrating Structures to Encoder

In order to incorporate the structural information into the computations of the input encoder, we propose to employ the internal architecture of the encoder. In particular, since the transformer-based encoders employ a self-attention mechanism to model the interactions between words, our hypothesis is that the structural information should be directly imposed on the self-attention mechanism. Concretely, to encourage the attention scores to be consistent with the adjacency matrices of different structures, we regularize the attention scores with the adjacency matrices of syntactic, semantic, and background knowledge structures\(^1\):

\[
L_{syn} = \sum_{h \in H} \alpha^{2}_{syn}(A_{syn} - A_{h,i,j}^{syn})^2 \quad (1)
\]

\[
L_{sem} = \sum_{h \in H} \alpha^{2}_{sem}(A_{sem} - A_{h,i,j}^{sem})^2 \quad (2)
\]

\[
L_{BK} = \sum_{h \in H} \alpha^{2}_{BK}(A_{BK} - A_{h,i,j}^{BK})^2 \quad (3)
\]

where \( l \) is the selected layer to be regularized with the corresponding structures and \( H \) is the total number of attention heads at layer \( l \). In order to select specific layers for each structure, following prior works [15, 16], we propose to employ the early layers for the syntactic structure, the middle layers for semantic structure, and finally the upper layers for the background knowledge structure. Concretely, the 12 layers of BERT or Longformer are divided into 3 groups of 4 layers. The layers \( l = 1 \ldots 4 \) are regularized with \( L_{syn} \), the layers \( l = 5 \ldots 8 \) are regularized with \( L_{sem} \) and finally the layers \( l = 9 \ldots 12 \) are regularized with \( L_{BK} \). Note that the decision on the number of layers and the order in which the structures are imposed to the encoder \( M \) is empirically supported in our experiments.

2.3. Graph Fusion

As discussed in the introduction, the rich interaction between words in the input document may involve heterogeneous paths with different lengths. As such, it is necessary to ensure that this type of interaction is encoded in the model. Although imposing the structural information into the internal computations of the input encoder could facilitate communication between different structures, it achieves this goal indirectly by passing the structural information from lower layers to upper layers. Unfortunately, it cannot guarantee the heterogeneous paths are encoded. To address this limitation, we propose to explicitly combine the

\(^{1}\)For the Longformer encoder, the global attention is computed for all tokens in the input.

\(^{2}\)https://github.com/bjascob/amrlib

\(^{3}\)Note that for the Longformer encoder, we use the global attention scores.
structural information using the representations obtained from the input encoder $M$. In particular, in this work, we employ Graph Transformer (GT) [17], to directly model the interactions between different structures. In particular, we first assemble the adjacency matrices $A = [A^{syn}, A^{sem}, A^{BK}]$. In addition to these general structures, we propose to use discourse-based structures too. Specifically, in order to better encode the context for the two given event mentions, we employ the entity co-reference links. Concretely, we first employ an entity co-reference resolution toolkit [20] to identify the entity mentions that refer to each other. Next, we construct the adjacency matrix $A^{ref}$ for entity co-references where $A_{i,j}^{ref}$ is 1 if words $w_i$ and $w_j$ refer to the same entity. In addition, following [17], we add the identity matrix $I$ (of size $n \times n$) to the set of structures to enable GT to encode multi-hop paths of different lengths, i.e., $A = [A^{syn}, A^{sem}, A^{BK}, A^{ref}, I] = [A_1, A_2, A_3, A_4, A_5]$. Inspired by the multi-head attention in Transformer architecture [21], GT consumes the input structures $A$ in C different channels. Each channel $i$ involves $m$ intermediate structures $Q_1, Q_2, \ldots, Q_m$ of size $n \times n$ ($1 \leq i \leq C$). Note that each channel corresponds to a layer in GT. To obtain the intermediate structure $Q_i$, the weighted sum of the input structures $A$ is computed: $Q_i = \sum_{j=1,5}^{m} \alpha_{i,j}^a A_{i,j}$, where $\alpha_{i,j}^a$ is a learnable weight. Using the learnable weights is a mechanism to facilitate the model to reason which input structure in $A$ is helpful to encode the interactions between the words for the ECI task. Afterward, in order to encode multi-hop paths in the input structure, the intermediate structures in the $i$-th channel are multiplied to create a final combined structure $Q_i$ for the $i$-th channel: $Q_i = Q_1 \times Q_2 \times \ldots \times Q_m$. We name $Q_i$ the final structure for the corresponding channel. Note that since the intermediate structures $Q_i$ involve all types of input structures they can encode heterogeneous paths. On the other hand, due to the inclusion of the identity matrix in $A$ and the multiplication of the intermediate structures, the final structure $Q_i$ encodes heterogeneous paths of different lengths. Note that similar to the multi-head attention mechanism, each channel in GT encodes different heterogeneous multi-hop paths, thus a richer document structure is encoded for the task of ECI.

In the next step, the final structures $Q_1^t, Q_2^t, \ldots, Q_C^t$ are exploited by different Graph Convolution Networks (GCN) [22, 23] to update the representations of the words using the multi-hop structures encoded by GT. Note that structures $Q_i$ are treated as weighted fully connected graphs by the GCN. Specifically, each GCN consists of $G$ layers. In each layer, an abstract representation for the words in $D$ is computed using the corresponding adjacency matrix $Q_i$. Formally, for the $k$-th final structure $Q_i^t$, the representation vector for the word $w_i$ in $t$-th layer of GCN is computed by:

$$\hat{h}_i^{t,k} = ReLU(U^t \Sigma_{j=1...n} Q_{i,j}^t \hat{h}_j^{k,t-1} \Sigma_{u=1...N} Q_{u,i}^t)$$

where $U^t$ is the weight matrix for the $t$-th GCN layer and the input vector $\hat{h}_i^{0,t}$ for the GCN model are taken from the final layer of the input encoder $M$ (i.e., $\hat{h}_i^{0,t} = e_i$ for all $1 \leq k \leq C$ and $1 \leq i \leq n$).

Given the final representation for the $i$-th word obtained from the $G$-th layer of GCN $k$, the final representation for each word is constructed by concatenating the vector representations $\hat{h}_i^{k, G}, \hat{h}_i^{k, G}, \ldots, \hat{h}_i^{k, G}$, where "·" is the concatenation operation. Note that the vectors $\hat{h}_i$ provide a structure-rich representation of the word $w_i$, in which different structural views and multi-hop heterogeneous paths are encoded.

### 2.4. Prediction

Given the final representations of the words $w_i$ in the document $D$, the feature vector to predict the event causality relation between the event mentions $w_i$ and $w_j$ is constructed as follows: $h_i = [\hat{h}_i^{1:G} : \hat{h}_i^{2:G} : \ldots : \hat{h}_i^{G:G}]$, where $h_i$ is structure-rich representations for the words $w_i$, obtained from the GCNs, $e_i$ is the input embedding for the word $w_i$ produced by the input encoder $M$ (i.e., BERT or Longformer). $h_D$ is the structure-rich representation for the document $D$ obtained via max-pooling of word representations $w_i$, i.e., $h_D = MAX_POOL_{i=1...n}(\hat{h}_i)$, $e_D$ is the input encoding of the document $D$ generated by max-pooling of the word embeddings $e_i$ for $1 \leq i \leq n$, and finally "·" represents concatenation operation. The feature vector $h_i$ is next fed into a two-layer feed-forward network with a sigmoid activation function at the end to produce the label probability $P(y_{id} = 1|D, s, t) = \sigma(W_2 \times (W_1 \times h + b_1) + b_2)$, where $W_1$ and $W_2$ are weight matrices and $h$ and $b_1$ and $b_2$ are biases. $\sigma$ is the sigmoid activation function. To train the model, we employ a binary cross-entropy loss function:

$$\mathcal{L}_{task} = - (y_{id} \log(P(y_{id} = 1|D, s, t)) + (1 - y_{id}) \log(1 - P(y_{id} = 1|D, s, t)))$$

(5)

where $y_{id}$ is the gold label for the input document $D$ and $y_{id} = 1$ indicates the existence of a causal relation between event mentions $w_i$ and $w_j$. Finally, to regularize the input encoder $M$ using the structural information, we employ the following combined loss function to train the entire model:

$$\mathcal{L} = \mathcal{L}_{task} + \alpha \mathcal{L}_{syn} + \beta \mathcal{L}_{sem} + \gamma \mathcal{L}_{BK}$$

(6)

where $\alpha$, $\beta$, and $\gamma$ are hyper-parameters to be adjusted based on the performance of the evaluation set. We name the proposed model, Event Casualty Identification with Heterogenous Structures (ECHI).

### 3. Experiments

We employ two datasets to evaluate the effectiveness of the proposed model. In particular, following prior works [12], we employ EventStoryLine and Causal-TimeBank datasets. Specifically, we use version 0.9 of EventStoryLine introduced by [24]. This dataset contains 258 documents, 22 topics, 4,316 sentences, 5,334 event mentions, 7,805 intra-sentence and, 4,652 inter-sentence event mention pairs, of which 1,770 and 3,855 pairs are annotated with causal relation, respectively. Documents in EventStoryLine are categorized based on their topics. Following [25], we order the topics based on their topic IDs and we use the last two topics for the evaluation set while the remaining topics are employed for 5-fold cross-validation evaluation. In our experiments, we follow [11, 26, 12] data split.

To evaluate the performance of the models, we report Precision, Recall, and F1 scores for correctly identifying the causal relations between candidate event pairs. We compare the proposed model with the prior state-of-the-art models reported in [12] and [13]. These models employ large language models combined with structural information obtained for the entire document.
Table 1: Performance of models on EventStoryLine dataset in different settings. [+] and [∗] represent models that employ BERT or Longformer encoders, respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>Intra-sentence</th>
<th>Inter-sentence</th>
<th>Intra+Inter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>F1</td>
<td>F1</td>
</tr>
<tr>
<td>OP</td>
<td>36.6</td>
<td>15.6</td>
<td>19.0</td>
</tr>
<tr>
<td>LSTM</td>
<td>37.4</td>
<td>18.7</td>
<td>23.2</td>
</tr>
<tr>
<td>Seq</td>
<td>37.8</td>
<td>16.4</td>
<td>21.4</td>
</tr>
<tr>
<td>KnowDis [0]</td>
<td>49.7</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Lr+</td>
<td>40.7</td>
<td>33.1</td>
<td>35.1</td>
</tr>
<tr>
<td>Lip</td>
<td>44.6</td>
<td>40.6</td>
<td>41.9</td>
</tr>
<tr>
<td>BERT [0]</td>
<td>43.7</td>
<td>25.3</td>
<td>30.8</td>
</tr>
<tr>
<td>Know [0]</td>
<td>50.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RichGCN [0]</td>
<td>55.2</td>
<td>42.2</td>
<td>46.6</td>
</tr>
<tr>
<td>GenECI [0]</td>
<td>58.8</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ERGO [0]</td>
<td>59.0</td>
<td>45.8</td>
<td>48.1</td>
</tr>
<tr>
<td>ERGO [+]</td>
<td>63.9</td>
<td>47.1</td>
<td>50.9</td>
</tr>
<tr>
<td>ECIRS (Ours) [0]</td>
<td>62.2</td>
<td>46.7</td>
<td>49.3</td>
</tr>
<tr>
<td>ECIRS (Ours) [+]</td>
<td>65.2</td>
<td>48.9</td>
<td>52.4</td>
</tr>
</tbody>
</table>

The evaluation results on the EventStoryLine dataset are presented in Table 1. This table clearly demonstrates the superiority of the proposed model in different settings, i.e., intra-sentence and inter-sentence ECI. Specifically, compared to the baselines that employ BERT embedding, the proposed ECIRS improves the state-of-the-art performance by 1.7% (on average). The better performance of ECIRS is evident when Longformer is exploited as an input encoder too. In particular, compared to ERGO with Longformer, ECIRS improves the F1 score by 1.5% (on average). Since the prior models employ structural information about the input document, this superior performance of the proposed model could be attributed to the richer structure-aware representations that are encoded by our model. Moreover, compared to other baselines that lack any structural information, the introduced model clearly achieves better performance. It indicates the necessity of integrating the structural information into the document encoding for ECI.

To provide insight into the generalization ability of the proposed method, we also evaluate it on the Causal-TimeBank dataset. The results of the experiment are presented in Table 2. Compared to the prior baselines, the proposed ECIRS improves the F1 score by 1.9% and 1.3% using BERT and Longformer embeddings, respectively. It is noteworthy that the reported improvements are achieved in the intra-sentence setting. This is important as it shows that structural information is required even for the shorter context of a sentence. This insight could provide the opportunity for further investigation of the importance of structural information for the task of ECI in all settings.

4. Related Work

The task of Event Causal Identification (ECI) is one of the important problems in the Information Exactions field. Prior works can be categorized under sentence level or document level. For sentence level, both event mentions are in the same sentence. These methods employ feature-based [27, 5, 28] or deep learning [4, 11, 29] architectures to encode the interaction between event mentions. For the document level, the event mentions might be in different sentences. For this setting, the structure of the text is exploited to improve the performance [12, 13]. These methods employ syntactic or semantic structures for this purpose. We also note some recent work on using generative models for ECI [30] or exploring ECI for non-English languages [31]. However, none of the prior works consider the efficient combination of heterogeneous structures that is proposed in this paper. Finally, our work on ECI is related to studies on event-event relation extraction where temporal relations and subevents [32, 33] are also considered.

5. Conclusion

In this work, we present a novel method for the task of event causal identification (ECI) at a document level. To improve the representations of the words produced by the document encoder, we propose to enrich the language model with different types of structural information for the given document. Specifically, the self-attention mechanism in the input transformer is regularized with the adjacency matrix of syntactic, semantic, and background knowledge structures. Moreover, to facilitate the interactions between different types of structures and encode multip-hop heterogeneous structures in the input text, we propose to further abstract the representations of the words using Graph Transformer (GT). We evaluate the performance of the proposed structure-based model on two benchmark datasets and the state-of-the-art performance is achieved on both datasets. Our experiment reveals the efficiency of the proposed ECIRS model in this setting too.

Acknowledgement: This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112, the NSF grant CNS-1747798 to the IUCRC Center for Big Learning and, the NSF grant # 2239570. This research is also supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract 2022-22072200003. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

6. References
