Graph Convolutional Networks for Event Causality Identification with Rich Document-level Structures

Minh Tran Phu
VinAI Research
Hanoi, Vietnam
v.minhtp@vinai.io

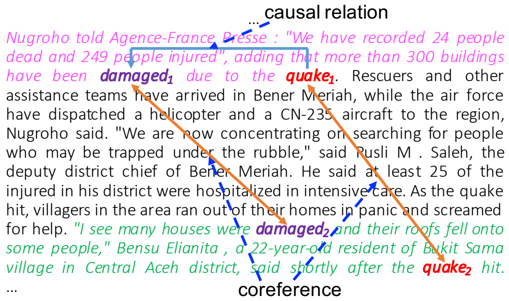
Thien Huu Nguyen
Department of Computer and Information Science
University of Oregon, Eugene, Oregon, USA
thien@cs.uoregon.edu

Abstract

We study the problem of Event Causality Identification (ECI) to detect causal relation between event mention pairs in text. Although deep learning models have recently shown state-of-the-art performance for ECI, they are limited to the intra-sentence setting where event mention pairs are presented in the same sentences. This work addresses this issue by developing a novel deep learning model for document-level ECI (DECI) to accept inter-sentence event mention pairs. As such, we propose a graph-based model that constructs interaction graphs to capture relevant connections between important objects for DECI in input documents. Such interaction graphs are then consumed by graph convolutional networks to learn document context-augmented representations for causality prediction between events. Various information sources are introduced to enrich the interaction graphs for DECI, featuring discourse, syntax, and semantic information. Our extensive experiments show that the proposed model achieves state-of-the-art performance on two benchmark datasets.

1 Introduction

Event Causality Identification (ECI) is an important problem in Information Extraction that seeks to predict causal relation between a pair of events mentioned in text. For instance, in the sentence "*The building was nearly destroyed by a fire early Tuesday morning.*", an ECI system should be able to recognize the causal relation between the two events triggered by "*destroyed*" and "*fire*" (called event mentions), i.e., "*fire*" $\xrightarrow{\text{cause}}$ "*destroyed*". ECI finds its applications for a wide range of problems in natural language processing (NLP), including machine reading comprehension (Berant et al., 2014), future event forecasting (Hashimoto, 2019), and why-question answering (Oh et al., 2016).



... Nugroho told Agence-France Presse : "We have recorded 24 people dead and 249 people injured", adding that more than 300 buildings have been **damaged$_1$** due to the **quake$_1$**. Rescuers and other assistance teams have arrived in Bener Meriah, while the air force have dispatched a helicopter and a CN-235 aircraft to the region, Nugroho said. "We are now concentrating on searching for people who may be trapped under the rubble," said Rusli M . Saleh, the deputy district chief of Bener Meriah. He said at least 25 of the injured in his district were hospitalized in intensive care. As the quake hit, villagers in the area ran out of their homes in panic and screamed for help. "I see many houses were **damaged$_2$** and their roofs fell onto some people," Bensu Elianita , a 22-year-old resident of Bukit Sama village in Central Aceh district, said shortly after the **quake$_2$** hit. ...

Figure 1: An example for document-level ECI.

The early approach for ECI has involved feature-based methods (Do et al., 2011; Hashimoto, 2019; Ning et al., 2018; Gao et al., 2019) while the recent approach has examined deep learning methods to deliver state-of-the-art performance for this task (Kadowaki et al., 2019; Liu et al., 2020). Despite the good performance, the existing deep learning methods for ECI are limited in that they only model the context at the sentence level, assuming the event mention pairs of interest to be in the same sentences (i.e., intra-sentence setting). On the one hand, this assumption fails to cover the inter-sentence scenario where the input pairs of event mentions can appear in different sentences in the documents, e.g., in the recent EventStoryLine dataset for ECI (Caselli and Vossen, 2017). On the other hand, the sole modeling of sentence context cannot benefit from the document-level information that can provide useful evidence to facilitate the causality prediction for events. An example can be seen in Figure 1 where the interested pair of event mentions involves *damaged*$_2$ and *quake*$_2$ in the last (green) sentence. A system that only considers sentence context might find it challenging to predict causal relation in this case due to the long distance and the appearance of many irrelevant words between *damaged*$_2$ and *quake*$_2$ in the sentence. However, if a system relies on document-level information and recognizes the coreference of the event mention pairs (*damaged*$_2$, *damaged*$_1$) and (*quake*$_2$, *quake*$_1$), it can exploit the clear ev-

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3480–3490
June 6–11, 2021. ©2021 Association for Computational Linguistics

idence of "*damaged*$_1$ *due to the quake*$_1$" to infer the causal relation for *damaged*$_2$ and *quake*$_2$.

To fill this gap, this work aims to develop a deep learning model for document-level ECI (DECI) where input event mentions can reside in different sentences of an input document. As such, a major challenge in modeling document-level context with deep learning involves capturing necessary interactions/connections between relevant objects for ECI. For instance, in our example in Figure 1, relevant objects include the event mentions and the important context words (i.e., "*due to*) while necessary connections involve event coreference and interactions of event mentions with context words (i.e., between *damaged*$_1$, *quake*$_1$, and "*due to*"). Motivated by this intuition, we design the graph-based model for DECI where interaction graphs over relevant objects for documents are explicitly generated and consumed by Graph Convolutional Networks (GCN) (Kipf and Welling, 2017; Nguyen and Grishman, 2018) to induce representation vectors for prediction. To our knowledge, this is the first work that employs interaction graphs for documents and GCNs for ECI.

How can interaction graphs for documents (i.e., nodes and edges) be formed to learn effective representation vectors for ECI? First, the intuitive approach to design nodes for interaction graphs is to leverage relevant objects for ECI in documents. Accordingly, we employ all the words, event mentions and entity mentions in a document to establish nodes for its interaction graph. Here, we note that entity mentions (e.g., names, pronouns, nominals) might also be helpful for ECI as entity mentions can serve as arguments (participants) of events and events with the same arguments might have better chance to involve in the causal relation.

Second, for edges of interaction graphs, we propose to exploit different knowledge sources or information types to create different types of connections for the graph nodes. Such connection types are then combined to produce a single rich interaction graph for an input document for representation learning in ECI. In particular, we focus on three major types of information for node connections for ECI in this work, i.e., discourse-based, syntax-based, and semantic-based information. As such, the discourse-based information explores the sentence boundary and coreference of entity/event mentions in documents to link the nodes in interaction graphs (motivated by our example in Figure 1). The syntax-based information connects words based on their syntactic relations in dependency trees of sentences, suggested by the use of shortest dependency paths between event mentions as features for ECI in prior work (Gao et al., 2019). In contrast, the intuition for semantic-based information is that semantically related words/entity/event mentions in documents can also provide useful evidences to infer the causal relation for events. For instance, consider the following sentence:

"*The violence in and near the Yida refugee camp, located 10 miles south of the border, came one day after* **bombings** *were reported in another region of South Sudan, an* **attack** *that provoked strong* **condemnation** *from the U.S. State Department.*"

Here, the causal relation between "*attack*" and "*condemnation*" can be easily predicted due to the direct evidence in the context (i.e., via "*provoked*"). However, the more complicated and implicit context between "*bombings*" and "*condemnation*" would make it more difficult for ECI systems to realize the causality in this case. Fortunately, the systems can combine the causal relation between "*attack*" and "*condemnation*" and the close semantic similarity between the two events "*bombings*" and "*attack*" to facilitate the causality prediction between "*bombings*" and "*condemnation*".

Finally, we propose a novel mechanism to regularize interaction graphs and representation vectors to further improve the representation learning for DECI. As such, we aim to constrain the model so edges with small weights in the generated graphs have minimized contribution to representation vectors. In this way, we expect the model to be more robust against irrelevant/noisy edges in the graphs and still promote useful edges for representation learning. We conduct extensive experiments on two datasets for DECI. The results demonstrate the effectiveness of the proposed model and lead to state-of-the-art performance for DECI.

2 Model

We formulate DECI as a binary classification problem. The input to the models include a document $D = w_1, w_2, \ldots, w_N$ (of N words/tokens) that can have multiple sentences, and two event mentions of interest e_s and e_t in D. The goal of DECI is to predict whether there exists a causal relation between e_s and e_t in D. Our model for DECI involves three major components: (i) Document Encoder to transform the words into representation vectors,

(ii) Structure Generation to generate an interaction graph for D, and (iii) Representation Regularization to regularize the representation vectors. We provide details for these components below.

2.1 Document Encoder

In the first step, we transform each word $w_i \in D$ into a representation vector x_i using the contextualized embeddings BERT (Devlin et al., 2019) (i.e., the BERT$_{base}$ version). In particular, as BERT might split w_i into several word-pieces, we employ the average of the hidden vectors for the word-pieces of w_i in the last layer of BERT as the representation vector x_i for w_i. To handle long documents with BERT, we divide D into segments of 512 word-pieces to be encoded separately. The resulting sequence $X = x_1, x_2, \ldots, x_n$ for D is then sent to the next steps for further computation.

2.2 Structure Generation

The goal of this section is to generate an interaction graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ for D to facilitate representation learning for DECI. As such, the nodes and edges in $\mathcal{G}$ for our DECI problem are constructed as follows:

Nodes: The node set $\mathcal{N}$ for our interaction graph $\mathcal{G}$ should capture relevant objects for the causal prediction between the two event mentions of interest e_s and e_t in D. As motivated earlier, we consider all the context words w_i, event mentions, and entity mentions in D as relevant objects for our DECI problem. Formally, let $E = \{e_1, e_2, \ldots, e_{|E|}\}$ and $M = \{m_1, m_2, \ldots, m_{|M|}\}$ be the sets of event mentions and entity mentions in D respectively (i.e., $e_s, e_t \in E$). The node set $\mathcal{N}$ for $\mathcal{G}$ is thus formed by the union of D, E, and M: $\mathcal{N} = D \cup E \cup M = \{n_1, n_2, \ldots, n_{|\mathcal{N}|}\}$. In this work, we use the provided event mentions in the datasets for E, following prior work on DECI (Gao et al., 2019) while the Stanford CoreNLP toolkit is employed to obtain the entity mentions M.

Edges: To formally represent the edges between the nodes in $\mathcal{N}$ for $\mathcal{G}$, we use the adjacency matrix $A = \{a_{ij}\}_{i,j=|\mathcal{N}|}$ ($a_{ij} \in \mathbb{R}$). Here, as we aim to use A as the input for Graph Convolutional Networks (GCN) to learn representation vectors for DECI, the value/score a_{ij} between two nodes n_i and n_j in $\mathcal{N}$ is expected to estimate the importance (or the level of interaction) of n_j for the representation computation of n_i. In this way, n_i and n_j can directly interact and influence the representation computation for each other even if they are sequentially far away from each other in D. As presented in the introduction, three types of information are exploited to design the edges $\mathcal{E}$ (or compute the interaction scores a_{ij}) for $\mathcal{G}$ in our model, including the discourse-based, syntax-based and semantic-based information.

Discourse-based Edges: As the input document D can involve multiple sentences and event/entity mentions, understanding where they span and how they relate to each other is crucial to effectively encode the document for DECI. As such, we propose to exploit three types of discourse information to obtain the interaction graph $\mathcal{G}$ for D, i.e., the sentence boundary, the coreference structure, and the mention span for event/entity mentions in D.

Sentence Boundary: The intuition for this type of information is that two event/entity mentions appearing in the same sentences tend to be more contextually related to each other than those in different sentences. This suggests the better usefulness of event/entity mentions in the same sentences for the representation computation of each other. To capture this intuition, we propose to compute the sentence boundary-based interaction score a_{ij}^{sent} for the nodes n_i and n_j in $\mathcal{N}$ where $a_{ij}^{sent} = 1$ if n_i and n_j are the event/entity mentions of the same sentences in D (i.e., $n_i, n_j \in E \cup M$); and 0 otherwise. a_{ij}^{sent} will be used as an input to compute the overall interaction score a_{ij} for $\mathcal{G}$ later.

Coreference Structure: Instead of considering within-sentence information as in a_{ij}^{sent}, coreference structure concerns the connection of event and entity mentions across sentences to enrich their representations with the contextual information of the coreferring ones (illustrated in Figure 1). As such, to enable the interaction of representations for coreferring event/enity mentions, we compute the conference-based score a_{ij}^{coref} for each pair of nodes n_i and n_j to contribute to the overall score a_{ij} for representation learning. Here, a_{ij}^{coref} is set to 1 if n_i and n_j are coreferring event/entity mentions in D, and 0 otherwise. Note that we use the Stanford CoreNLP toolkit to determine the coreference of entity mentions while similar to (Gao et al., 2019), golden event coreference information in the DECI datasets is utilized in this work.

Mention Span: The sentence boundary and coreference structure scores only model interactions of event and entity mentions in D based on discourse information. To further connect event and entity mentions with context words w_i for representation learning, we employ the mention span-based interaction score a_{ij}^{span} as another input for a_{ij}, where a_{ij}^{span} is only set to 1 (i.e., 0 otherwise) if n_i is a word ($n_i \in D$) in the span of the entity/event mention n_j ($n_j \in E \cup M$) or vice verse. Note that a_{ij}^{span} is important as it allows representation vectors for event/entity mentions to be grounded on the contextual information in D.

Syntax-based Edges: Prior work has leveraged dependency parsing trees of sentences in documents as an useful source of information to generate features for DECI systems, e.g., using the shortest dependency paths between the two event mentions of interest (Gao et al., 2019). As such, we expect the dependency trees of the sentences in D can also provide beneficial information to connect the nodes in $\mathcal{N}$ to learn effective representation vectors for DECI. To this end, we propose to employ the dependency relations/connections between the words in D to obtain a syntax-based interaction score a_{ij}^{dep} for each pair of nodes n_i and n_j in $\mathcal{N}$, serving as an additional input for a_{ij}. In particular, directly inheriting the graph structures of the dependency trees of the sentences in D, we set a_{ij}^{dep} to 1 if n_i and n_j are two words in the same sentence (i.e., $n_i, n_j \in D$) and they are connected to each other in the corresponding dependency tree, and 0 otherwise. Thus, two words are considered important to each other for representation learning in DECI if they are neighbors in the dependency trees[1].

Semantic-based Edges: This information exploits the semantic similarity of the nodes in $\mathcal{N}$ to enrich the overall interaction scores a_{ij} for $\mathcal{G}$. The motivation is that a node n_i would contribute more to the representation vector of another node n_j for DECI if n_i is more semantically related to n_j (illustrated in the introduction). To this end, we propose two complementary methods to compute the semantic similarity between the nodes for a_{ij} based on context-based and knowledge-based information.

Context-based Semantic: In this method, we seek to first obtain a representation vector v_i for the semantic of each node n_i in $\mathcal{N}$ based on its context in D. The context-based semantic similarity $a_{ij}^{context}$ for the nodes is then be computed via such representation vectors and fed into the estimation of the overall interaction score a_{ij}. In particular, the context-based representation vector v_i for a word node $n_i \in D$ is directly inherited from the contextualized embedding vector $x_c \in X$ (i.e., $v_i = x_c$) of the corresponding word w_c for n_i. In contrast, for event and entity mentions, their representation vectors are computed by max-pooling the contextualized embedding vectors in X that correspond to the words in the event/entity mentions' spans. Eventually, the context-based similarity score $a_{ij}^{context}$ for two nodes n_i and n_j in $\mathcal{N}$ is obtained via the normalized score:

$$k_i = U_k v_i, q_i = U_q v_i$$
$$a_{ij}^{context} = \exp(k_i q_j) / \sum_{u=1..|\mathcal{N}|} \exp(k_i q_u) \quad (1)$$

where U_k and U_q are trainable weight matrices, and the biases are omitted for brevity in this work.

Knowledge-based Semantic: Instead of using contextual information, this method leverages the external knowledge of the nodes from knowledge bases to capture their semantic for node similarity computation. We expect the external knowledge for the nodes to provide complementary information for the contextual information in D, thus further enriching the semantic similarity scores (and overall interaction scores a_{ij}) for the nodes in $\mathcal{N}$. To this end, we propose to utilize WordNet (Miller, 1995), a rich knowledge base for word meanings, to obtain external knowledge for the words in D. As such, WordNet involves a network of word meanings (i.e., synsets) that are connected to each other via various semantic relations (e.g., synonyms, hyponyms). Our first step to generate knowledge-based similarity scores involves mapping each word node $n_i \in D \cap \mathcal{N}$ to a synset node M_i in WordNet using a Word Sense Disambiguation (WSD) tool. In particular, we employ WordNet 3.0 and the state-of-the-art BERT-based WSD model in (Blevins and Zettlemoyer, 2020) to perform the word-synset mapping in this work. Afterward, we compute a knowledge-based similarity score a_{ij}^{struct} for each pair of word nodes n_i and n_j in $D \cap \mathcal{N}$ using the structure-based similarity of their linked synsets M_i and M_j in WordNet (i.e., $a_{ij}^{struct} = 0$ if either n_i or n_j is not a word node in $D \cap \mathcal{N}$). Accordingly, the Lin similarity measure (Lin et al., 1998) for synset nodes in WordNet is utilized for this purpose: $a_{ij}^{struct} = \frac{2*\text{IC}(\text{LCS}(M_i,M_j))}{\text{IC}(M_i)+\text{IC}(M_j)}$, where IC and LCS amount to the information content of synset nodes and the least common subsumer of two synsets in the WordNet hierarchy (the most specific ancestor node) respectively[2].

[1] We use Stanford CoreNLP to parse the sentences.

[2] We use the `nltk` tool to obtain the Lin similarity: https://www.nltk.org/howto/wordnet.html.

Structure Combination: Up to now, six scores have been generated to capture the level of interactions in representation learning for each pair of nodes n_i and n_j in $\mathcal{N}$ according to different information sources (i.e., a_{ij}^{sent}, a_{ij}^{coref}, a_{ij}^{span}, a_{ij}^{dep}, $a_{ij}^{context}$ and a_{ij}^{struct}). For convenience, we group the six scores for each node pair n_i and n_j into a vector $d_{ij} = [a_{ij}^{sent}, a_{ij}^{coref}, a_{ij}^{span}, a_{ij}^{dep}, a_{ij}^{context}, a_{ij}^{struct}]$ of size 6. To unify the scores in d_{ij} to form an overall rich interaction score a_{ij} for n_i and n_j in $\mathcal{G}$, we use the following normalization:

$$a_{ij} = \exp(d_{ij}q^T) / \sum_{u=1..|\mathcal{N}|} \exp(d_{iu}q^T) \qquad (2)$$

where q is a learnable vector of size 6.

As mentioned above, given the combined interaction graph $\mathcal{G}$ with the adjacency matrix $A = \{a_{ij}\}_{i,j=|\mathcal{N}|}$, we use GCNs to induce representation vectors for the nodes in $\mathcal{N}$ for DECI. In particular, the GCN model in our work takes the context-based representation vectors v_i of the nodes $n_i \in \mathcal{N}$ as the input. For convenience, we organize v_i into rows of the input matrix $H_0 = [v_1, \dots, v_{|\mathcal{N}|}]$. The GCN model then involves G layers that generate the matrix H_l at the l-th layer for the nodes in $\mathcal{N}$ ($1 \leq l \leq G$) via: $H_l = ReLU(AH_{l-1}W_l)$ (W_l is the weight matrix for the l-th layer). The output of the GCN model after G layers is H_G whose rows are denoted by $H_G = [h_1, \dots, h_{|\mathcal{N}|}]$, serving as more abstract representation vectors for the nodes n_i for causality prediction. This GCN-based computation of H_L is written as $H_G = [h_1, \dots, h_{|\mathcal{N}|}] = GCN(H_0, A, \mathcal{G})$ for convenience.

2.3 Representation Regularization

Our model so far renders $\mathcal{G}$ as a fully connected graph for representation learning whose edge weights are induced and recorded in the adjacency matrix $A = \{a_{ij}\}_{i,j=1..|\mathcal{N}|}$ ($0 < a_{ij} < 1$). However, it is intuitive that not all the edges in $\mathcal{G}$ are relevant/necessary for the representation vectors in DECI. Some edges might even introduce noisy information if they are preserved in the graph. As such, we hypothesize that edges with small weights/scores assigned by the learning process in A are mostly noisy edges and should have minimal contribution to the induced representation vectors. To this end, we propose to obtain a sparser version $\mathcal{G}'$ of $\mathcal{G}$ where edges with small weights are completely eliminated. In particular, we employ a threshold α ($0 < \alpha < 1$) and compute the adjacency matrix $A' = \{a'_{ij}\}_{i,j=1..|\mathcal{N}|}$ for $\mathcal{G}'$ via: $a'_{ij} = a_{ij}$ if $a_{ij} > \alpha$; and 0 otherwise.

To explicitly encourage the minimal contribution of small-weight edges, our goal is to enforce that the representation vectors learned by the sparse graph $\mathcal{G}'$ are still close to those learned by the full graph $\mathcal{G}$ (i.e., the removal of small-weight edges in $\mathcal{G}'$ does not have much effect on representation learning). To implement this idea, we first apply our GCN model over the sparse graph $\mathcal{G}'$ to learn another version of GCN-based representation vectors for the nodes in $\mathcal{N}$: $H'_G = [h'_1, \dots, h'_{|\mathcal{N}|}] = GCN(H_0, A', \mathcal{G}')$. Afterward, we seek to minimize the difference L_{reg} between representation vectors of corresponding nodes in H_G and H'_G in the overall loss function: $L_{reg} = 1/|\mathcal{N}| \sum_{i=1..|\mathcal{N}|} ||h_i - h'_i||_2^2$.

Finally, let $n_{s'}$ and $n_{t'}$ be the two nodes in $\mathcal{N}$ that correspond to the two event mentions of interest e_s and e_t for DECI. An overall representation vector $V = [h_{s'}, h_{t'}, h'_{s'}, h'_{t'}]$ is formed (from both H_L and H'_L) and fed into a two-layer feed-forward network with softmax in the end to produce the distribution $P(.|D, e_s, e_t)$ over the two possible types for our DECI problem (whether there is a causal relation between e_s and e_t or not). The negative log-likelihood function L_{pred} is then computed by: $L_{pred} = -\log P(y^*|D, e_s, e_t)$ (y^* is the golden type for DECI). The overall loss function to train our model is thus: $L = L_{pred} + \gamma L_{reg}$ where γ is a trade-off parameter.

3 Experiments

3.1 Datasets and Hyperparameters

Following prior work (Gao et al., 2019; Liu et al., 2020), we evaluate our models on two benchmark datasets for ECI, i.e., EventStoryLine and Causal-TimeBank. In particular, EventStoryLine (i.e., version 0.9) is introduced in (Caselli and Vossen, 2017), involving 258 documents, 22 topics, 4316 sentences, 5334 event mentions, 7805 intra-sentence and 46521 inter-sentence event mention pairs (1770 and 3855 of them are annotated with a causal relation respectively). Following (Gao et al., 2018), we group documents according to their topics and put the topics in the order based on their topic IDs. The documents in the last two topics are used for the development data while the remaining 20 documents are employed for a 5-fold

	Intra-sentence			Inter-sentence			Intra + Inter		
Model	P	R	F1	P	R	F1	P	R	F1
OP (Caselli and Vossen, 2017)	22.5	98.6	36.6	8.4	99.5	15.6	10.5	99.2	19.0
LSTM (Cheng and Miyao, 2017)	34.0	41.5	37.4	13.5	30.3	18.7	17.6	33.9	23.2
Seq (Choubey and Huang, 2017)	32.7	44.9	37.8	11.3	29.5	16.4	15.5	34.3	21.4
KnowDis* (Zuo et al., 2020)	39.7	66.5	49.7	-	-	-	-	-	-
LR+ (Gao et al., 2019)	37.0	45.2	40.7	25.2	48.1	33.1	27.9	47.2	35.1
LIP (Gao et al., 2019)	38.8	52.4	44.6	35.1	48.2	40.6	36.2	49.5	41.9
BERT* (our implementation)	39.2	49.3	43.7	22.3	29.2	25.3	27.3	35.3	30.8
Know* (Liu et al., 2020)	41.9	62.5	50.1	-	-	-	-	-	-
RichGCN* (proposed)	49.2	63.0	**55.2**	39.2	45.7	**42.2**	42.6	51.3	**46.6**

Table 1: Model's performance on EventStoryLine. The performance improvement of RichGCN over the baselines is significant with $p < 0.01$. * designates models that use BERT embeddings.

cross-validation evaluation, using the same data split in (Gao et al., 2019; Liu et al., 2020). For Causal-TimeBank (Mirza, 2014a), this dataset consists of 184 documents, 6813 events, and 318 of 7608 event mention pairs annotated with a causal relations. Following (Liu et al., 2020), we do a 10-fold cross-validation evaluation using the same data split for this dataset. Note that as in (Liu et al., 2020), we only evaluate the ECI performance for intra-sentence events in Causal-TimeBank as the number of inter-sentence event mention pairs with the causal relation is very small (i.e., only 18 pairs).

We tune the hyperparameters for our model on the development data of EventStoryLine and use the chosen parameters to train the models for both EventStoryLine and Causal-TimeBank. The selected values from the tuning process include: $1e$-5 for the learning rate of the Adam optimizer; 8 for the mini-batch size; 128 hidden units for all the feed-forward network and GCN layers; 2 layers for the GCN model ($G = 2$), $\alpha = 0.5$ for the weight threshold, and $\gamma = 0.2$ for the trade-off parameter in the loss function L. Finally, as mentioned earlier, we use the BERT$_{base}$ model (of 768 dimensions) for the pre-trained word embeddings (updated during the training) in this work.

3.2 Main Results

We compare our model (called **RichGCN**) with the state-of-the-art models for ECI in each benchmark dataset as follows.

EventStoryLine: For this dataset, the following baselines are chosen for comparison: (i) **OP**: a dummy model used in (Caselli and Vossen, 2017) that assigns a causal relation to every pair of event mentions; (ii) **LSTM** (Gao et al., 2019): a dependency path based sequential model that is adopted from (Cheng and Miyao, 2017); (iii) **Seq** (Gao et al., 2019): another dependency path based sequential model that is originally developed for temporal relation prediction from (Choubey and Huang, 2017) and applied to ECI; (iv) **BERT**: a baseline method that takes the embedding vectors from BERT and performs ECI in (Liu et al., 2020). Note that (Liu et al., 2020) only reports the performance on intra-sentence events of EventStoryLine for this model. We reimplement and fine-tune the model to obtain its performance for inter-sentence events. Our reimplemented model for **BERT** achieves higher performance on intra-sentence ECI than those in (Liu et al., 2020); (v) **KnowDis** (Zuo et al., 2020): a BERT-based model that leverages additional data from distant supervision; (vi) **LR+** and **LIP** (Gao et al., 2019): document structure-based models that have the current state-of-the-art performance for inter-sentence ECI; and (vii) **Know** (Liu et al., 2020): a BERT-based model that exploits ConceptNet and achieves the state-of-the-art performance for intra-sentence ECI. Table 1 shows the performance of the models.

Model	P	R	F1
RB (Mirza, 2014b)	36.8	12.3	18.4
ML (Mirza, 2014a)	67.3	22.6	33.9
BERT* (Liu et al., 2020)	30.3	41.1	34.9
Know* (Liu et al., 2020)	36.6	55.6	44.1
RichGCN* (proposed)	39.7	56.5	**46.7**

Table 2: Model's performance on Causal-TimeBank (for intra-sentence events). RichGCN is significantly better than the baselines with $p < 0.01$. * indicates BERT-based models.

Causal-TimeBank: We use the following baselines for this dataset: (i) **RB**: a rule-based system in (Mirza, 2014b); (ii) **ML**: a feature-based model for ECI in (Mirza, 2014a); and (iii) **BERT** and **Know** (Liu et al., 2020): These are the same models **BERT** and **Know** (respectively) for EventStoryLine (both are based on BERT). We use the reported performance for the two models in (Liu et al., 2020) for a fair comparison. **Know** has the current state-of-the-art performance for this dataset in our 10-fold cross-validation setting. Note that the **BERT** model essentially corresponds to our **RichGCN** model when the interaction graphs $\mathcal{G}$ and $\mathcal{G}'$ (thus the GCN model) are completely excluded. Table 2 presents the performance of these models on Causal-TimeBank.

The most important observation from the tables is that the proposed model **RichGCN** significantly outperforms all the baselines for both intra- and inter-sentence events on both EventStoryLine and Causal-TimeBank ($p < 0.01$), thus clearly demonstrating the effectiveness of the proposed model for DECI. In addition, we also see that **BERT** performs much worse than the document structure-based models **LR+**, **LIP** and **RichGCN**. The sequential modeling of the context in BERT is thus not effective for document-level ECI, necessitating better mechanisms to encode document context (e.g., via the interaction graph of relevant objects as we do). Finally, the significant better performance of **RichGCN** over **Know** for intra-sentence ECI in different datasets confirms our intuition in the introduction that capturing context beyond sentences (i.e., document context as in **RichGCN**) is helpful for causal prediction of intra-sentence event pairs.

3.3 Ablation Study

This section analyzes the contribution of each component in the proposed model with an ablation study. In particular, we examine the following ablated models: (i) “**RichGCN - x**” where **x** is one of the six interaction scores generated to compute the unified score a_{ij} (i.e., a_{ij}^{sent}, a_{ij}^{coref}, a_{ij}^{span}, a_{ij}^{dep}, $a_{ij}^{context}$ and a_{ij}^{struct}). For instance, “**RichGCN -** a_{ij}^{coref}” refers to the **RichGCN** model where the coreference-based interaction score a_{ij}^{coref} is excluded in the computation of the overall score a_{ij} in Equation 2; (ii) “**RichGCN - Entity Nodes**”: the entity mention nodes in M are not included in the construction of interaction graph $\mathcal{G}$ in this model (i.e., $\mathcal{N} = D \cup E$ only); (iii) “**RichGCN - Event Nodes**”: the event mention nodes in E do not appear in the node set $\mathcal{N}$ of the interaction graph $\mathcal{G}$ in **RichGCN** (i.e., $\mathcal{N} = D \cup M$). We directly use the representation vectors v_i for the event mentions in the overall representation vector V for prediction in this model. Note that the interaction matrix A is also adapted accordingly in the ablated models “**RichGCN - Entity Nodes**” and “**RichGCN - Event Nodes**”; (iv) “**RichGCN - GraphCombination**”: this model does not combine the six generated interaction scores to compute an overall score a_{ij} for A in Equation 2. Instead, it considers each of the six generated interaction scores as forming a separate interaction graph, thus generating six different graphs. The GCN model is then applied over these six graphs (using the same input representation vectors v_i for the nodes n_i in $\mathcal{N}$). The outputs of the GCN model for the same node n_i (with different graphs) are then concatenated to produce the final representation vector for n_i (i.e., serving as h_i in the model). Note that we still employ the sparse graph idea (with $\mathcal{G}'$ and the loss L_{reg}) in this model; (v) “**RichGCN -** $\mathcal{G}$” and “**RichGCN -** $\mathcal{G}$’”: these models exclude the full graphs $\mathcal{G}$ or $\mathcal{G}'$ from **RichGCN** (respectively). The regularization loss L_{reg} is thus not used and the vectors generated by the excluded graphs are not employed in the final vector V (i.e., $h_{s'}, h_{t'}, h'_{s'}, h'_{t'}$) for prediction in these cases; and (vi) “**RichGCN -** L_{reg}”: this model removes the regularization term L_{reg} from the overall loss function L.

Model	Intra	Inter	Intra +Inter
RichGCN (full)	**59.5**	**43.3**	**48.3**
RichGCN - a_{ij}^{sent}	55.8	40.3	44.9
RichGCN - a_{ij}^{coref}	57.2	37.6	43.0
RichGCN - a_{ij}^{span}	49.2	35.8	39.5
RichGCN - a_{ij}^{dep}	54.1	39.5	43.5
RichGCN - $a_{ij}^{context}$	57.7	42.2	46.9
RichGCN - a_{ij}^{struct}	57.5	41.5	46.3
RichGCN - GraphCombination	54.7	41.1	44.7
RichGCN - Entity Nodes	51.6	38.7	42.8
RichGCN - Event Nodes	52.0	38.4	42.5
RichGCN - $\mathcal{G}$	53.7	39.9	44.3
RichGCN - $\mathcal{G}'$	54.6	40.5	44.8
RichGCN - L_{reg}	56.8	41.3	46.3

Table 3: Performance of models (F1) on the development data of EventStoryLine.

Table 3 shows the performance of the models on the development data of EventStoryLine. As can be seen from the table, all the components are helpful for the proposed model **RichGCN** as eliminating any of them degrades the performance significantly for both intra- and inter-sentence ECI. Notably, the worse performance of “**RichGCN -** $\mathcal{G}$” suggests that only using the sparse graph $\mathcal{G}$’ for GCN to completely cancel small-weight edges in $\mathcal{G}$ is suboptimal as it might unexpectedly remove some useful (though small-weight) edges. Instead, the sparse graph should be exploited in conjunction with the full graph to minimize the overall contribution of small-weight edges as we do in **RichGCN**.

3.4 Cross-Topic Evaluation

To further demonstrate the benefits of document context modeling with GCN for intra-sentence ECI, we perform a cross-topic evaluation on EventStoryLine as in (Liu et al., 2020). In particular, as documents in different topics tend to mention different events in EventStoryLine, this section aims to train the models on a source topic, but evaluate them on other topics (i.e., the target topics) to reveal the topic generalization. Following (Liu et al., 2020), we choose topics T8, T13, and T18 in EventStoryLine as the source topics. For each of these source topics, the other topics are ranked according to their similarity with the source topic. As such, the similarity score between two topics t_1 and t_2 is based on $\delta = \frac{E_{t_1} \cap E_{t_2}}{E_{t_1} \cup E_{t_2}}$, where E_t is the set of lemmas of event trigger words in topic t. Afterward, topics with the lowest, medium and highest similarity scores with the source topic are chosen as the target topics for evaluation. Table 4 present the intra-sentence ECI performance (i.e., F1 scores) of **LIP**, **Know** (Liu et al., 2020) and the proposed model **RichGCN** for this cross-topic experiment.

Setting	Source (Train)	Target (Test)	δ	LIP	Know	**RichGCN (Proposed)**
Low	T8	T35	0%	2.8	44.7	**47.0**
	T13	T12	0%	-	25.1	**42.7**
	T18	T30	0%	-	19.5	**28.2**
Medium	T8	T3	1.7%	6.7	30.9	**38.0**
	T13	T41	0.1%	4.5	28.6	**41.6**
	T18	T35	2.8%	17.1	44.5	**50.0**
High	T8	T19	12.4%	19.4	45.1	**54.0**
	T13	T14	17.1%	27.4	46.0	**50.5**
	T18	T33	27.2%	32.2	49.0	**53.1**

Table 4: Cross-topic performance (F1) for inter-sentence ECI. $\delta = \frac{E_{t_1} \cap E_{t_2}}{E_{t_1} \cup E_{t_2}}$ is the topic similarity score.

It is clear from the table that **RichGCN** is significantly better than the baselines **LIP** and **Know** over different cross-topic settings, thereby further testifying to the generalization advantages of capturing document-level context via GCN for intra-sentence ECI in the proposed model.

3.5 Error Analysis

To suggest potential directions for future research, we analyze the errors made by the proposed model. In particular, we sample 100 event mention pairs in the development data of EventStoryLine whose causal relation cannot be predicted correctly by **RichGCN**. Afterward, we manually categorize these examples into different types that are described below:

(i) **Implicit causal relation**: 33% of the errors in our model is due to the implicit indication of the causal relation between two event mentions in the context, necessitating common-sense knowledge to make correct causality prediction. For instance, consider the following document:

"*South Sudan warns of war after Sudan bombs refugee camp. Military aircraft from Sudan crossed the new international border with South Sudan and dropped bombs Thursday in and around a camp filled with refugees, officials said. A government official initially reported deaths, but an American activist who spoke to aid workers at the camp later said there were no casualties.*"

RichGCN cannot recognize the causal relation between two events "*bombs*" and "*deaths*" in this document. The reason is that there is no explicit context in the document to hint such a relation. The models need to rely on the common-sense causal order of "*bombs*" and "*deaths*" to correctly predict the label in this case.

(ii) **Preprocessing toolkit**: Our model leverages several toolkit to obtain information to construct the interaction graph $\mathcal{G}$, including the dependency parser, the entity mention detection and coreference (i.e., from Stanford CoreNLP), and the word sense disambiguation model. 18% errors in our model originate from the errors in such toolkit that introduce noise into our model. For instance, Stanford CoreNLP incorrectly considers "*South Sudan*" and "*Sudan*" as the same entity in some of the examples.

(iii) **Lack of factuality modeling**: Our model fails in this error type as it does not consider the factuality of the causal relation, treating hypothetical relations as the actual ones. This accounts for 5% of the errors. For instance, in the document above, the proposed model predicts the causal relation between "*war*" and "*bombs*"; however, this is incorrect (not factual) due to the appearance of the word "*warns*".

(iv) **Lack of fine-grained distinction**: The errors in this type (accounting for 23%) are due to the failure to capture the fine-grained distinction of event mentions in the context, causing the confusion and incorrect predictions for the model. For instance, in the sentence "*Updated : July 02 , 2013 15:50 IST A 6. 1-magnitude earthquake which hit the Indonesian province of Aceh on Tues-*

day killed a child, injured dozens and destroyed buildings, sparking panic in a region devastated by the quake-triggered tsunami of 2004.", our model incorrectly predict "*killed*" and "*injured*" as having a causal relation with "*quake*" (underlined). This stems from the strong connection between the underlined "*quake*" and the "*1-magnitude earthquake*" in the same sentence (i.e., due to the sentence boundary- and semantic-based interaction scores). Such strong connection leads the model to believe that "*killed*" and "*injured*" are also caused by the underlined "*quake*" as the "*1-magnitude earthquake*". The model would need to better encode the fine-grained distinction between the underlined "*quake*" and the "*1-magnitude earthquake*" (i.e., of the year 2004 and 2013 respectively) to address this issue. Finally, our analysis shows that the other errors have to do with annotation errors (6%) and more complicated issues that cannot be categorized clearly.

4 Related Work

The early feature-based methods for ECI has explored different features and resources to improve the performance, including lexical and syntactic patterns (Hashimoto, 2019; Gao et al., 2019), causality cues/markers (e.g., "*because*") (Riaz and Girju, 2014a; Hidey and McKeown, 2016), statistical co-occurrence of events (Beamer and Girju, 2009; Do et al., 2011; Hu et al., 2017), temporal patterns (Mirza, 2014a; Ning et al., 2018), lexical semantics of events (Riaz and Girju, 2013, 2014b), and weakly supervised data (Hashimoto, 2019). Although we also apply related features and resource for ECI (e.g., syntax, WordNet), our model employs such resources to build interaction graphs for documents to induce more abstract representations with GCNs. Recently, deep learning has been applied to solve ECI, leveraging advanced language models (e.g., BERT) (Kadowaki et al., 2019; Zuo et al., 2020) and common-sense knowledge resources (i.e., ConceptNet) (Liu et al., 2020) to produce state-of-the-art performance. However, none of these deep learning models has explored document-context modeling with rich information for graph construction and GCNs as we do.

Recently, there have been much interest in designing task-specific graphs to learn representation vectors for different NLP tasks, including sentence-level graphs for event factuality identification (Pouran Ben Veyseh et al., 2019) and event argument extraction (Pouran Ben Veyseh et al., 2020; Nguyen and Nguyen, 2021), and document-level graphs for relation extraction (Christopoulou et al., 2019; Nan et al., 2020; Tran et al., 2020) and event argument extraction (Veyseh et al., 2021). Our model is different from such related work in that we design document-level interaction graphs that are tailored to our ECI task. In addition, our model is also the first model that employs the inherent structure of external knowledge graphs (i.e., WordNet) to augment interaction graphs for documents in representation learning.

5 Conclusion

We present a novel deep learning model for document-level ECI to address the limitation of prior deep learning models that only focus on causal prediction for inter-sentence event mention pairs. Our model designs interaction graphs to capture important objects and connections for input documents, leveraging GCNs to induce representation vectors for causal prediction. We introduce several information sources to enrich the interaction graphs based on discourse, syntax, and semantic information. The experiments confirm the effectiveness of the proposed information sources and models for DECI. In the future, we plan to extend our model to other related tasks, e.g., event coreference resolution (Nguyen et al., 2016).

Acknowledgments

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. 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 ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Brandon Beamer and Roxana Girju. 2009. Using a bigram event model to predict causal potential. In *CICLing*.

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, Doha, Qatar. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

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.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6, Vancouver, Canada. Association for Computational Linguistics.

Prafulla Kumar Choubey and Ruihong Huang. 2017. A sequential model for classifying temporal relations between intra-sentence events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1796–1802, Copenhagen, Denmark. Association for Computational Linguistics.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. 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 4925–4936, Hong Kong, China. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Quang Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 294–303, Edinburgh, Scotland, UK. Association for Computational Linguistics.

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.

Qiaozi Gao, Shaohua Yang, Joyce Chai, and Lucy Vanderwende. 2018. What action causes this? towards naive physical action-effect prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 934–945, Melbourne, Australia. Association for Computational Linguistics.

Chikara Hashimoto. 2019. Weakly supervised multilingual causality extraction from Wikipedia. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2988–2999, Hong Kong, China. Association for Computational Linguistics.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433, Berlin, Germany. Association for Computational Linguistics.

Zhichao Hu, Elahe Rahimtoroghi, and Marilyn Walker. 2017. Inference of fine-grained event causality from blogs and films. In *Proceedings of the Events and Stories in the News Workshop*, pages 52–58, Vancouver, Canada. 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.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Dekang Lin et al. 1998. An information-theoretic definition of similarity. In *ICML*.

Jian Liu, Yubo Chen, and Jun Zhao. 2020. Knowledge enhanced event causality identification with mention masking generalizations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3608–3614. International Joint Conferences on Artificial Intelligence Organization.

George A Miller. 1995. Wordnet: a lexical database for english. In *Communications of the ACM*.

Paramita Mirza. 2014a. Extracting temporal and causal relations between events. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 10–17, Baltimore, Maryland, USA. Association for Computational Linguistics.

Paramita Mirza. 2014b. Fbk-hlt-time: a complete italian temporal processing system for eventi-evalita 2014. In *EVALITA*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

Minh Van Nguyen and Thien Huu Nguyen. 2021. Improving cross-lingual transfer for event argument extraction with language-universal sentence structures. In *Proceedings of the Arabic Natural Language Processing Workshop at EACL 2021*, Online. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana. Association for the Advancement of Artificial Intelligence.

Thien Huu Nguyen, Adam Meyers, and Ralph Grishman. 2016. New york university 2016 system for kbp event nugget: A deep learning approach. In *Proceedings of Text Analysis Conference (TAC)*.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.

Jong-Hoon Oh, K. Torisawa, C. Hashimoto, R. Iida, M. Tanaka, and Julien Kloetzer. 2016. A semi-supervised learning approach to why-question answering. In *AAAI*.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019. Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4393–4399, Florence, Italy. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661, Online. Association for Computational Linguistics.

Mehwish Riaz and Roxana Girju. 2013. Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In *SIGDIAL*.

Mehwish Riaz and Roxana Girju. 2014a. In-depth exploitation of noun and verb semantics to identify causation in verb-noun pairs. In *SIGDIAL*.

Mehwish Riaz and Roxana Girju. 2014b. Recognizing causality in verb-noun pairs via noun and verb semantics. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 48–57, Gothenburg, Sweden. Association for Computational Linguistics.

Hieu Minh Tran, Minh Trung Nguyen, and Thien Huu Nguyen. 2020. The dots have their values: Exploiting the node-edge connections in graph-based neural models for document-level relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4561–4567, Online. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Hung Tran, Varun Manjunatha, Lidan Wang, Rajiv Jain, Doo Soon Kim, Walter Chang, and Thien Huu Nguyen. 2021. Inducing rich interaction structures between words for document-level event argument extraction. In *Proceedings of the 25th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.