OntoED: Low-resource Event Detection with Ontology Embedding

Shumin Deng[1,2]*, **Ningyu Zhang**[1,2]*, **Luoqiu Li**[1,2], **Hui Chen**[3], **Huaixiao Tou**[3],
Mosha Chen[3], **Fei Huang**[3], **Huajun Chen**[1,2]†

[1]Zhejiang University & AZFT Joint Lab for Knowledge Engine, China
[2]Hangzhou Innovation Center, Zhejiang University, China
[3]Alibaba Group, China

{231sm, zhangningyu, 3160102409}@zju.edu.cn, {weidu.ch, huaixiao.thx}@alibaba-inc.com,
{chenmosha.cms, f.huang}@alibaba-inc.com, huajunsir@zju.edu.cn

Abstract

Event Detection (ED) aims to identify event trigger words from a given text and classify it into an event type. Most of current methods to ED rely heavily on training instances, and almost ignore the correlation of event types. Hence, they tend to suffer from data scarcity and fail to handle new unseen event types. To address these problems, we formulate ED as a process of event ontology population: linking event instances to pre-defined event types in event ontology, and propose a novel ED framework entitled OntoED with ontology embedding. We enrich event ontology with linkages among event types, and further induce more event-event correlations. Based on the event ontology, OntoED can leverage and propagate correlation knowledge, particularly from data-rich to data-poor event types. Furthermore, OntoED can be applied to new unseen event types, by establishing linkages to existing ones. Experiments indicate that OntoED is more predominant and robust than previous approaches to ED, especially in data-scarce scenarios.

1 Introduction

Event Detection (ED) (Chen et al., 2015) is the task to extract structure information of events from unstructured texts. For example, in the event mention "*Jack is married to the Iraqi microbiologist known as Dr. Germ.*", an ED model should identify the event type as '*Marry*' where the word '*married*' triggers the event. The extracted events with canonical structure facilitate various social applications, such as biomedical science (Li et al., 2019; Wang et al., 2020c), financial analysis (Deng et al., 2019; Liang et al., 2020), fake news detection (Wang et al., 2018; Nikiforos et al., 2020) and so on.

As a non-trivial task, ED suffers from the low-resource issues. On the one hand, the maldistribu-
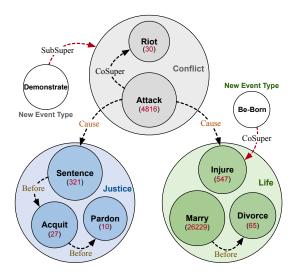

Figure 1: Low-resource Event Detection *w.r.t.* Event Correlation in *FewEvent* (Deng et al., 2020) Dataset.

tion of samples is quite serious in ED benchmark datasets, *e.g.*, *FewEvent* (Deng et al., 2020) and *MAVEN* (Wang et al., 2020b), where a large portion of event types contain relatively few training instances. As shown in Figure 1, the sample size of two event types *Attack* and *Riot* differs greatly (4816 & 30). In low-resource scenarios, supervised ED models (Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018) are prone to overfitting since they require sufficient training instances for all event types. On the other hand, real-world applications tend to be open and evolve promptly, and accordingly there can be numerous new unseen event types. Handling new event types may even entail starting over, without being able to re-use annotations from previous ones (Huang et al., 2018).

Regarding low-resource ED, Huang et al. (2018) take a fresh look at ED, by mapping each event mention to a specific type in a target event ontology, which can train from few seen event types and then transfer knowledge to new unseen ones. However, the event ontology here merely considers the intra-structure for each event mention and event type.

---

* Equal Contribution.
† Corresponding Author.

In this paper, we enrich the event ontology with more inter-structures of event types, such as temporal, causal and hierarchical event-event relations (Ning et al., 2018; Wang et al., 2020a). For example, as seen in Figure 1, *Attack* $\xrightarrow{\text{CAUSE}}$ *Sentence*, *Sentence* $\xrightarrow{\text{BEFORE}}$ *Acquit*, *Attack* $\xrightarrow{\text{COSUPER}}$ *Riot*. Our key intention is to fully utilize the event ontology and leverage correlation knowledge from data-rich event types (*i.e.*, *Attack*) to data-poor ones (*i.e.*, *Sentence*, *Acquit* and *Riot*). Besides, new event types (*i.e.*, *Be-Born*) can be learned with correlation (*i.e.*, COSUPER) of existing ones (*i.e.*, *Injure*).

As the first attempt to construct such event ontology, we propose a novel ED framework with ontology embedding called OntoED. First, we establish the initial event ontology with event instances and types. We capture semantic features and relations of event instances with BERT (Devlin et al., 2019) and utilize prototypes (Snell et al., 2017) to represent event types, where a prototype is the average of its instance embeddings. Second, we extend the event ontology with event-event relations based on extracted relations among event instances, and then learn ontology embedding by aggregating neighbor prototypes for each prototype *w.r.t.* correlations among event types. In this way, semantically similar event types in vector space will be closer, thus, improving the discrimination of dissimilar event types. Third, we design an event correlation inference mechanism to induce new event correlations based on symbolic rules, *e.g.*, (*Sentence*, BEFORE, *Acquit*) ∧ (*Acquit*, BEFORE, *Pardon*) → (*Sentence*, BEFORE, *Pardon*). Thus, we can induce new event-event relations to further enrich the event ontology. To the best of our knowledge, it is the first work to explicitly model correlations among event types with event ontology in low-resource ED.

Our contributions can be summarized as follows:

- We study the low-resource event detection problem and propose a novel ontology-based model, OntoED, that encodes intra and inter structures of events.

- We provide a novel ED framework based on ontology embedding with event correlations, which interoperates symbolic rules with popular deep neural networks.

- We build a new dataset `OntoEvent` for ED. Extensive experimental results demonstrate that our model can achieve better performance on the overall, few-shot, and zero-shot setting.

2 Related Work

Traditional approaches to ED are mostly based on neural networks (Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018; Wang et al., 2019; Yan et al., 2019; Cui et al., 2020; Shen et al., 2020; Lou et al., 2021), and ignore correlation knowledge of event types, especially in low-resource scenarios. Most previous low-resource ED methods (Peng et al., 2016) have been based on *supervised learning*. However, supervised-based methods are too dependent on data, and fail to be applied to new types without additional annotation efforts. Another popular methods for low-resource ED are based on *meta learning*. Deng et al. (2020); Lai et al. (2020); Shen et al. (2021) reformulate ED as a few-shot learning problem to extend ED with limited labeled samples to new event types, and propose to resolve few-shot ED with meta learning. Besides, *knowledge enhancement* and *transfer learning* are applied to tackle low-resource ED problems. Tong et al. (2020) leverage open-domain trigger knowledge to address long-tail issues in ED. Liu et al. (2020); Du and Cardie (2020) propose to handle few-shot and zero-shot ED tasks by casting it as a machine reading comprehension problem. Huang et al. (2018) propose to tackle zero-shot ED problem by mapping each event mention to a specific type in a target event ontology. Note that Huang et al. (2018) establish the event ontology merely with intra-structure of events, while we extend it with inter-structure of event correlations. Though these methods are suitable for low-resource scenarios, they mostly ignore implicit correlation among event types and lack reasoning ability.

In order to utilize correlation knowledge among event types, Li et al. (2020) propose a new event graph schema, where two event types are connected through multiple paths involving entities. However, it requires various annotations of entities and entity-entity relations, which is complicated and demanding. Different from Li et al. (2020), we propose to revisit the ED task as an ontology learning process, inspired by relation extraction (RE) tasks based on ontology and logic-based learning. Lima et al. (2018, 2019) present a logic-based relational learning approach to RE that uses inductive logic programming for generating information extraction (IE) models in the form of symbolic rules, demonstrating that ontology-based IE approaches are advantageous in capturing correlation among classes, and succeed in symbolic reasoning.

3 Methodology

3.1 Problem Formulation

We revisit the event detection task as an iterative process of event ontology population. Given an event ontology $\mathcal{O}$ with an event type set $\mathcal{E} = \{e_i | i \in [1, N_e]\}$, and corpus $\mathcal{T} = \{X_i | i \in [1, K]\}$ that contains K instances, the goal of event ontology population is to establish proper linkages between event types and instances. Specifically, each instance X_i in $\mathcal{T}$ is denoted as a token sequence $X_i = \{x_i^j | j \in [1, L]\}$ with maximum L tokens, where the event trigger x_i^t are annotated. We expect to predict the index t ($1 \leq t \leq L$) and the event label e_i for each instance respectively.

Besides, we utilize a multi-faceted event-event relation set $\mathcal{R} = \mathcal{R}_H \sqcup \mathcal{R}_T \sqcup \mathcal{R}_C$ for event ontology population and learning. Thereinto, $\mathcal{R}_H = \{\text{SUBSUPER}, \text{SUPERSUB}, \text{COSUPER}^1\}$ denotes a set of relation labels defined in the subevent relation extraction task (Wang et al., 2020a; Yao et al., 2020). $\mathcal{R}_T = \{\text{BEFORE}, \text{AFTER}, \text{EQUAL}^2\}$ denotes a set of temporal relations (Han et al., 2020). $\mathcal{R}_C = \{\text{CAUSE}, \text{CAUSEDBY}\}$ denotes a set of causal relations (Ning et al., 2018).

3.2 Model Overview

In this paper, we propose a general framework called *OntoED* with three modules: (1) Event Detection (Ontology Population), (2) Event Ontology Learning, and (3) Event Correlation Inference. Figure 2 shows the key idea of the three modules.
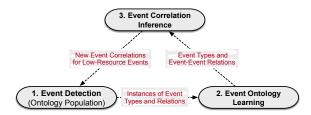

Figure 2: Overview of our proposed OntoED.

Event Detection aims at identifying the event trigger x_i^t and type e_i for each input tokens X_i, and then identify relations among event instances. The average instance embedding of each type is calculated as the primitive event prototype.

Event Ontology Learning aims to obtain event ontology embedding with the correlation of event prototypes, based on the relations among event types derived from instances.

---

[1] $(e_i, \text{COSUPER}, e_j)$: e_i and e_j has the same super type.
[2] (e_i, EQUAL, e_j): e_i and e_j happens simultaneously.

Event Correlation Inference seeks to infer new event correlations based on existing event-event relations, so as to obtain a solid event ontology.

The detailed architecture of OntoED with running examples is illustrated in Figure 3.

3.3 Event Detection (Ontology Population)

The input of ED is an initial event ontology with event types $\mathcal{E}$ and coarse corpus $\mathcal{T}$.

Instance Encoder. Given a token sequence $X_i = \{x_i^1, \cdots, x_i^L\}$ with trigger x_i^t, we use a pre-trained BERT (Devlin et al., 2019) to get a contextual representation $\boldsymbol{X}_i^t$ for x_i^t, and use the token embedding of [CLS] as the contextual representation $\boldsymbol{X}_i$ for X_i. Note that the instance encoder is pluggable, and can be replaced as other models followed by (Deng et al., 2020; Cui et al., 2020).

Class Encoder. We then represent event types as *prototypes* (Snell et al., 2017), as it is proven to be robust for low-resource ED (Deng et al., 2020).

Initially, event types have no correlation with others, thus we require to compute the *prototype* $\boldsymbol{P}_k$ for $e_k \in \mathcal{E}$ by averaging its instance embeddings:

$$\boldsymbol{P}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \boldsymbol{X}_i \qquad (1)$$

where N_k is the instance number of e_k. Afterward, event prototypes will be induced from the module of event correlation inference, as shown in Figure 3.

Event Detector. Given embeddings of a token sequence, we treat each token as an event trigger candidate and then compute probability of the corresponding event type for event trigger candidate x_i^t, denoted by:

$$P(y = e_k) = \frac{exp(-\|\boldsymbol{X}_i^t - \boldsymbol{P}_k\|)}{\sum_{j=1}^{N_e} exp(-\|\boldsymbol{X}_i^t - \boldsymbol{P}_j\|)} \qquad (2)$$

where $\| \cdot \|$ denotes Euclidean distance, and $N_e = |\mathcal{E}|$ denotes the number of event types.

As general, we adopt cross entropy as the loss function for event detection, denoted by:

$$\mathcal{L}_{ED} = -\sum_{k=1}^{N_e} y \log P(y = e_k) \qquad (3)$$

where y is the ground-truth label for x_i^t.

Instance Relation Extractor. For each event instance pair (X_i, X_j), we adopt a comprehensive way to model embedding interactions (Zhou et al., 2020), denoted by $\boldsymbol{X}_{ij}^p = [\boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{X}_i \odot \boldsymbol{X}_j, \boldsymbol{X}_i - \boldsymbol{X}_j]$, where $[\cdot, \cdot]$ denotes a vector concatenation, and $\odot$ is the element-wise Hadamard product.
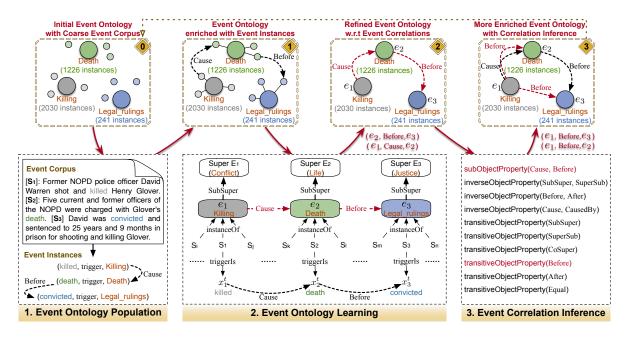
Figure 3: Detailed example for the process of OntoED. Note that we ignore instance nodes in No.2 and No.3 event ontology for space limit. **Step 1**: **Event Detection (Ontology Population)** connect event types with instances, given the initial event ontology with coarse corpus. **Step 2**: **Event Ontology Learning** establish correlations among event types, given the event ontology enriched with instances. **Step 3**: **Event Correlation Inference** induce more event correlations based on existing event-event relations, *e.g.*, $(e_1, \text{CAUSE}, e_2) \rightarrow (e_1, \text{BEFORE}, e_2)$, and $(e_1, \text{BEFORE}, e_2) \wedge (e_2, \text{BEFORE}, e_3) \rightarrow (e_1, \text{BEFORE}, e_3)$.

We then calculate the probability $P(y = r_k)$ of relation $r_k \in \mathcal{R}$ between (X_i, X_j) by softmax. Generally, we adopt cross entropy as the loss function for instance relation extraction, denoted by:

$$\mathcal{L}_{RE} = -\sum_{k=1}^{N_r} y \log P(y = r_k) \qquad (4)$$

where y is the ground-truth for (X_i, X_j), and $N_r = |\mathcal{R}|$ denotes the number of event-event relations.

Overall, the loss function for event detection (ontology population) is calculated by:

$$\mathcal{L}_{OP} = \gamma \mathcal{L}_{ED} + (1 - \gamma) \mathcal{L}_{RE} \qquad (5)$$

where γ is a hyperparameter.

3.4 Event Ontology Learning

Ontology Completion. We complete event ontology $\mathcal{O}$ with both intra and inter structure of events. We normatively link event instances $\mathcal{T}$ to event types $\mathcal{E}$, and establish correlations among event types based on linkages among event instances.

Instance-to-class Linking. Given a sentence S_i (formalized as a token sequence X_i) with a trigger x_i^t of an event instance, we link these information to its corresponding event type e_i with normative triples: $(S_i, \text{triggerIs}, x_i^t)$ and $(S_i, \text{instanceOf}, e_i)$.

Class-to-class Linking. Given an event instance pair (X_i, X_j) with a relation r, we upgrade the

instance correlation to corresponding event types, denoted by (e_i, r, e_j). Besides, we link each event subtype to its corresponding supertype[3] with a SUBSUPER relation (SUPERSUB in reverse), and we link each event subtype pair having the same supertype with a COSUPER relation.

Ontology Embedding. We represent the event ontology considering both instances and correlations for each event type. Specifically, given a triple $\ell = (e_h, r, e_t) \in \mathcal{O}$, we propagate the prototype P_h of head event type e_h to prototype P_t of tail event type e_t with a relation transformation matrix $M_r \in \mathbb{R}^{d \times d}$. We select a matrix to embed r as it shows great robustness to model relations in low-resource senarios (Zhang et al., 2019). We then aggregate propagation from all head event types by

$$P_t^* = \sum_{(e_h^i, r_i, e_t) \in \mathcal{O}_\ell} P_h^i M_{r_i} \qquad (6)$$

where $\mathcal{O}_\ell$ is all one-hop neighbor triples of e_t in $\mathcal{O}$.

The prototype P_t of e_t in ℓ after propagation is a weighted average of P_t and P_t^* with weight $\lambda \in [0, 1]$, denoted by:

$$P_t = \lambda P_t + (1 - \lambda) P_t^* \qquad (7)$$

---

[3]The supertypes and its corresponding subtypes in this paper are pre-defined and will be introduced in appendix.

Object Property Axioms	Rule Form	According to Linear Map Assumption	Relation Constraint
$\texttt{subOP}(r_1, r_2)$	$(e_i, r_2, e_j) \leftarrow (e_i, r_1, e_j)$	$\boldsymbol{P}_i \boldsymbol{M}_{r_2} = \boldsymbol{P}_j, \ \boldsymbol{P}_i \boldsymbol{M}_{r_1} = \boldsymbol{P}_j$	$\boldsymbol{M}_{r_1} = \boldsymbol{M}_{r_2}$
$\texttt{inverseOP}(r_1, r_2)$	$(e_i, r_1, e_j) \leftarrow (e_j, r_2, e_i)$	$\boldsymbol{P}_i \boldsymbol{M}_{r_1} = \boldsymbol{P}_j, \ \boldsymbol{P}_j \boldsymbol{M}_{r_2} = \boldsymbol{P}_i$	$\boldsymbol{M}_{r_1} \boldsymbol{M}_{r_2} = \boldsymbol{I}$
$\texttt{transitiveOP}(r)$	$(e_i, r, e_k) \leftarrow (e_i, r, e_j), (e_j, r, e_k)$	$\boldsymbol{P}_i \boldsymbol{M}_r = \boldsymbol{P}_k, \ \boldsymbol{P}_i \boldsymbol{M}_r = \boldsymbol{P}_j, \ \boldsymbol{P}_j \boldsymbol{M}_r = \boldsymbol{P}_k$	$\boldsymbol{M}_r \boldsymbol{M}_r = \boldsymbol{M}_r$

Table 1: Three types of object property expression axioms. $\texttt{OP}$ is the short for $\texttt{ObjectProperty}$. $\boldsymbol{I}$ is an identity matrix.

We calculate the possibility that r is the relation between e_h and e_t with a truth value for (e_h, r, e_t): $\phi(e_h, r, e_t) = sim(\boldsymbol{P}_h \boldsymbol{M}_r, \boldsymbol{P}_t) = \sigma(\boldsymbol{P}_h^\top \boldsymbol{M}_r \boldsymbol{P}_t)$, where σ is sigmoid function, and the similarity between $\boldsymbol{P}_h \boldsymbol{M}_r$ and $\boldsymbol{P}_t$ is evaluated via dot product.

Overall, the loss fuction for event ontology learning is defined by:

$$\mathcal{L}_{OL} = - \sum_{(e_h, r, e_t) \in \mathcal{O}} y \log \phi(e_h, r, e_t) \quad (8)$$

and y denotes the ground-truth label for (e_h, r, e_t).

3.5 Event Correlation Inference

Given the event ontology with correlations among event types, we infer new event correlations based on existing ones. To be specific, we utilize the grounding g to infer new event correlation triples, which can be generalized as the following form:

$$(e_h^I, r^I, e_t^I) \leftarrow (e_h^1, r^1, e_t^1), \cdots, (e_h^n, r^n, e_t^n) \quad (9)$$

where the right side event triples $(e_h^k, r^k, e_t^k) \in \mathcal{O}$ with $k \in [1, n]$ have already existed in $\mathcal{O}$ and $(e_h^I, r^I, e_t^I) \notin \mathcal{O}$ is new inferred triples to be added.

To compute the truth value of the grounding g, we select three object properties ($\texttt{OP}$) of relations defined in OWL2[4] Web Ontology Language: $\texttt{subOP}$, $\texttt{inverseOP}$, and $\texttt{transitiveOP}$, and then learn matrics of relations from linear map assumption (Zhang et al., 2019), presented in Table 1. Wang et al. (2020a); Ning et al. (2018) have defined some conjunctive constraints of relations between the event pair, we translate them into object property axioms, shown in Table 2.

Object Property Axioms	Instances of Relation / Relation Pair
$\texttt{subOP}(r_1, r_2)$	(CAUSE, BEFORE)
$\texttt{inverseOP}(r_1, r_2)$	(SUBSUPER, SUPERSUB), (BEFORE, AFTER), (CAUSE, CAUSEDBY)
$\texttt{transitiveOP}(r)$	SUBSUPER, SUPERSUB, COSUPER, BEFORE, AFTER, EQUAL

Table 2: Groundings of three object properties in $\mathcal{O}$.

Assuming that $\boldsymbol{M}_r^\dagger$ and $\boldsymbol{M}_r^\ddagger$ denotes the relation set on left and right of Eq (9) respectively, they are

---

[4]https://www.w3.org/TR/owl2-profiles/

matrices either from a single matrix or a product of two matrices. As relation constraints are derived from ideal linear map assumption (the 3rd column in Table 1), $\boldsymbol{M}_r^\dagger$ and $\boldsymbol{M}_r^\ddagger$ are usually unequal but similar during training. Thus, the normalized truth value $\mathcal{F}_p$ of g can be calculated based on relation constraints (the 4th column in Table 1):

$$\mathcal{F}_p' = \|\boldsymbol{M}_r^\dagger - \boldsymbol{M}_r^\ddagger\|_F, \quad \mathcal{F}_p = \frac{\mathcal{F}_p^{max} - \mathcal{F}_p'}{\mathcal{F}_p^{max} - \mathcal{F}_p^{min}}$$

where $\| \cdot \|_F$ denotes Frobenius norm, and subscript p respectively denotes one of the three object properties. $\mathcal{F}_p^{max}$ and $\mathcal{F}_p^{min}$ is a the maximum and minimum Frobenius norm score. $\mathcal{F}_p \in [0, 1]$ is the truth value for the grounding g and the higher $\mathcal{F}_p$ means the more confident that g is valid.

The loss function for new event correlation inference is defined by:

$$\mathcal{L}_{ER} = - \psi_S \sum_{i \in \mathcal{G}^{(S)}} \log \mathcal{F}_p^i - \psi_V \sum_{j \in \mathcal{G}^{(V)}} \log \mathcal{F}_p^j$$
$$- \psi_T \sum_{k \in \mathcal{G}^{(T)}} \log \mathcal{F}_p^k$$

$$(10)$$

$\mathcal{G}^{(\cdot)}$ denotes all groundings w.r.t. $\texttt{subOP}$ (S), $\texttt{inverseOP}$ (V), and $\texttt{transitiveOP}$ (T). ψ_S, ψ_V, and ψ_T are hyperparameters for the loss of three object properties respectively.

As a whole, the final loss function for OntoED is denoted by:

$$\mathcal{L} = \alpha \mathcal{L}_{OP} + \beta \mathcal{L}_{OL} + \mathcal{L}_{ER} \quad (11)$$

where α and β are hyperparameters for the loss of event ontology population (Eq (5)) and event ontology learning (Eq (8)) respectively.

4 Experiments

The experiments seek to: (1) demonstrate that OntoED with ontology embedding can benefit both standard and low-resource ED, and (2) assess the effectiveness of different modules in OntoED and provide error analysis. To this end, we verify the effectiveness of OntoED in three types of evaluation: (1) *Overall Evaluation*, (2) *Few-shot Evaluation*, and (3) *Zero-shot Evaluation*.

4.1 Datasets

As none of present datasets for ED is annotated with relations among events, we propose a new ED dataset namely `OntoEvent` with event correlations. It contains 13 supertypes with 100 subtypes, derived from 4,115 documents with 60,546 event instances. The details of `OntoEvent` are introduced in appendix. We show the main statistics of `OntoEvent` and compare them with some existing widely-used ED datasets in Table 3.

Dataset	#Doc	#Ins	#SuperT	#SubT	#E-E Rel
ACE 2005	599	4,090	8	33	None
TAC KBP 2017	167	4,839	8	18	None
FewEvent	-	70,852	19	100	None
MAVEN	4,480	111,611	21	168	None
OntoEvent	4,115	60,546	13	100	3,804

Table 3: Statistics of `OntoEvent` compared with existing widely-used ED datasets. (Doc: document, Ins: instance, SuperT: supertype, SubT: subtype, E-E Rel: event-event relation.)

`OntoEvent` is established based on two newly proposed datasets for ED: MAVEN (Wang et al., 2020b) and FewEvent (Deng et al., 2020). They are constructed from Wikipedia documents or based on existing event datasets, such as ACE-2005[5] and TAC-KBP-2017[6]. In terms of event-event relation annotation in `OntoEvent`, we jointly use two models: TCR (Ning et al., 2018) is applied to extract temporal and causal relations, and JCL (Wang et al., 2020a) is used for extract hierarchical relations. The code of OntoED and `OntoEvent` dataset can be obtained from Github[7].

4.2 Baselines

For *overall evaluation*, we adopt CNN-based model DMCNN (Chen et al., 2015), RNN-based model JRNN (Nguyen et al., 2016), and GCN-based model JMEE (Liu et al., 2018). Besides, we adopt BERT-based model AD-DMBERT (Wang et al., 2019) with adversarial imitation learning. We also adopt graph-based models OneIE (Lin et al., 2020) and PathLM (Li et al., 2020) which generate graphs from event instances for ED.

For *few-shot evaluation* and *zero-shot evaluation*, we adopt some metric-based models for few-shot ED, such as MatchNet (Lai et al., 2020), ProtoNet (Snell et al., 2017) and DMBPN (Deng et al., 2020). We also adopt knowledge-enhanced model EKD (Tong et al., 2020) and BERT-based models QAEE (Du and Cardie, 2020) as well as RCEE (Liu et al.,

---

[5] http://projects.ldc.upenn.edu/ace/
[6] https://tac.nist.gov/2017/KBP/Event/index.html
[7] https://github.com/231sm/Reasoning_In_EE

2020) based on machine reading comprehension. Besides, we adopt ZSEE (Huang et al., 2018) especially for zero-shot ED.

4.3 Experiment Settings

With regard to settings of the training process, SGD (Ketkar, 2014) optimizer is used, with 30,000 iterations of training and 2,000 iterations of testing. The dimension of token embedding is 50, and the maximum length of a token sequence is 128. In OntoED, a dropout rate of 0.2 is used to avoid over-fitting, and the learning rate is 1×10^{-3}. The hyperparameters of γ, λ, α, and β are set to 0.5, 0.5, 1.5 and 1 respectively. ψ_S, ψ_V, and ψ_T are set to 0.5, 0.5 and 1 respectively. As the dataset is unbalanced, we evaluate the performance of ED with macro precision (P), Recall (R) and adopt micro F1 Score (F) following (Chen et al., 2015). Detailed performance can be found in Github[7].

4.4 Overall Evaluation

Setting. We follow the similar evaluation protocol of standard ED models, *e.g.*, DMCNN (Chen et al., 2015). Event instances are split into training, validating, and testing subset with ratio of 0.8, 0.1 and 0.1 respectively. Note that there are no new event types in testing set which are not seen in training.

As seen from Table 4, OntoED achieves larger gains compared to conventional baselines, *e.g.*, DMCNN, JRNN and JMEE. Moreover, OntoED still generally excel BERT-based AD-DMBERT. This implies the effectiveness of ED framework with ontology embedding, which can leverage and propagate correlations among event types, so that reduce the dependence on data to some extent. Especially, OntoED also outperform graph-based models, *i.e.*, OneIE and PathLM. The possible reason is that although they both convert sentences into instance graphs, and PathLM even connects event types with multiple entities, the event correlations are still implicit and hard to capture. OntoED can explicitly utilize event correlations and directly propagate information among event types.

4.5 Few-shot Evaluation

Setting. We follow the similar evaluation protocol and metrics of data-scarce ED models, *i.e.*, RCEE (Liu et al., 2020), which train models with partial data. We randomly sample nearly 80% event types for training, 10% for validating, and 10% for testing. Differently from overall evaluation, the event types in testing set are not exsiting in training set.

Model	Trigger Identification			Event Classification		
	P	R	F	P	R	F
DMCNN (Chen et al., 2015)	64.65 ± 0.89	64.17 ± 0.94	64.15 ± 0.91	62.51 ± 1.10	62.35 ± 1.12	63.72 ± 0.99
JRNN (Nguyen et al., 2016)	65.94 ± 0.88	66.67 ± 0.95	66.30 ± 0.93	63.73 ± 0.98	63.54 ± 1.13	66.95 ± 1.03
JMEE (Liu et al., 2018)	70.92 ± 0.90	57.58 ± 0.96	61.87 ± 0.94	52.02 ± 1.14	53.80 ± 1.15	68.07 ± 1.02
AD-DMBERT (Wang et al., 2019)	74.94 ± 0.95	72.19 ± 0.91	73.33 ± 0.97	67.35 ± 1.01	$\mathbf{73.46 \pm 1.12}$	71.89 ± 1.03
OneIE (Lin et al., 2020)	74.33 ± 0.93	71.46 ± 1.02	73.68 ± 0.97	71.94 ± 1.03	68.52 ± 1.05	71.77 ± 1.01
PathLM (Li et al., 2020)	75.82 ± 0.85	72.15 ± 0.94	74.91 ± 0.92	73.51 ± 0.99	68.74 ± 1.03	72.83 ± 1.01
OntoED	$\mathbf{77.67 \pm 0.99}$	$\mathbf{75.92 \pm 0.92}$	$\mathbf{77.29 \pm 0.98}$	$\mathbf{75.46 \pm 1.06}$	70.38 ± 1.12	$\mathbf{74.92 \pm 1.07}$

Table 4: Evaluation of event detection with overall instances. $P(\%)$, $R(\%)$ and $F(\%)$ stand for precision, recall, and F1-score respectively.

Model	1%	5%	10%	15%	20%
MatchNet (Lai et al., 2020)	7.09 ± 2.10	14.22 ± 2.18	20.59 ± 2.11	26.34 ± 2.07	30.93 ± 1.98
ProtoNet (Snell et al., 2017)	8.18 ± 2.51	15.25 ± 2.27	21.74 ± 2.02	27.75 ± 2.01	32.21 ± 1.66
DMBPN (Deng et al., 2020)	11.25 ± 2.13	20.03 ± 1.99	27.69 ± 1.95	33.13 ± 1.91	38.06 ± 1.54
EKD (Tong et al., 2020)	35.82 ± 2.02	44.51 ± 1.83	49.64 ± 1.77	52.79 ± 1.26	55.95 ± 1.34
QAEE (Du and Cardie, 2020)	41.17 ± 1.85	48.69 ± 1.84	54.03 ± 1.81	58.97 ± 1.81	62.04 ± 1.67
RCEE (Liu et al., 2020)	42.58 ± 1.94	50.06 ± 1.81	56.51 ± 1.71	60.79 ± 1.13	63.98 ± 1.08
OntoED	$\mathbf{44.98 \pm 1.98}$	$\mathbf{52.19 \pm 1.74}$	$\mathbf{58.53 \pm 1.75}$	$\mathbf{62.47 \pm 1.52}$	$\mathbf{65.51 \pm 1.17}$

Table 5: F1 score (%) of event classification on extremely sparse intances for few-shot evaluation.
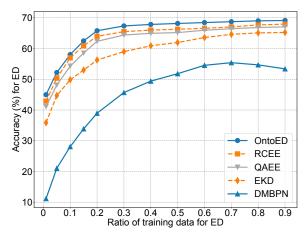

Figure 4: Results on different ratios of ED training data.

As seen from Table 5, we demonstrate F1 score results in extremely low-resource scenarios (training with less than 20% data, with the similar setting to Liu et al. (2020)). Obviously, OntoED behaves tremendous advantages in low-resource ED. For example, OntoED obtains 44.98% F1 with 1% data, in comparison to 7.09% in MatchNet and 8.18% in ProtoNet. We also illustrate accuracy results with different ratios of training data followed by Liu et al. (2020), show in Figure 4. As seen, OntoED demonstrates superior performance with less data dependence than baselines. Especially comparing with DMBPN and EKD, which require 60% training data to closely achieve the best results, while OntoED only uses 20%. Besides, we find that the performance on DMBPN increases first and then slightly decreases as the ratio of training data increases, the possible reason may lie in data noise and redundancy. In low-resource scenarios, more data are not always better. Particularly for some merely data-driven ED models, such as DMBPN, may obtain a worse effect instead if added data are dirty or duplicated. But for OntoED, as it utilizes correlation knowledge in the event ontology and has less dependence on event instances, making it more robust to noisy and redundant data. Furthermore, OntoED also outperforms than BERT-based model with regarding each event instance as a question, *i.e.*, QAEE and RCEE. This implies that event ontology learning with event type knowledge may resolve low-resource ED more advantageously than training merely with event instances.

4.6 Zero-shot Evaluation

Setting. We follow the similar evaluation protocol and metrics of zero-shot ED models, *i.e.*, ZSEE (Huang et al., 2018), and comply with the same dataset segmentation policy as few-shot evaluation, thus there are also new unseen event types for testing. Differently, ED data are completely banned for training, meaning that we train models only with event types other than instances.

Table 6 demonstrates the results regarding zero-shot ED. We can see that OntoED achieves best precision and F1 score as well as comparable recall results in comparison to baselines. This illustrates the effectiveness of OntoED handling new unseen event types without introducing outsourcing data.

Model	P	R	F
EKD (Tong et al., 2020)	32.58	31.77	32.17
QAEE (Du and Cardie, 2020)	36.69	37.33	37.01
RCEE (Liu et al., 2020)	37.45	36.83	37.14
ZSEE (Huang et al., 2018)	40.92	**44.18**	43.02
OntoED	**42.13**	44.04	**43.06**

Table 6: Comparisons of performance on zero-shot ED.

Traditional models, such as EKD and RCEE, require to adopt other datasets, *e.g.*, WordNet (Miller et al., 1990) (where words are grouped and interlinked with semantic relations) and FrameNet (Baker, 2014) (where frames are treated as meta event types) to increase the persuasiveness of results. In contrast, OntoED naturally models the structure of event types with an event ontology, thus even for a new unseen event type without instance data, we can also obtain its representation through the event-event correlation. Moreover, OntoED is also beneficial to resolve zero-shot ED than ZSEE. This may due to OntoED modeling with both intra and inter structures of events while ZSEE merely considering the intra-structure.

5 Further Analysis

5.1 Ablation Study

To assess the effect of event ontology learning and correlation inference, we remove the two modules in OntoED, and evaluate F1 score shown in Figure 5. From the results, we observe that OntoED outperforms the two baselines in all evaluation settings, indicating that event ontology learning and correlation inference facilitate ED, as they utilize knowledge among event types and has less dependence on instance data. Furthermore, in terms of performance degradation compared to OntoED, F1 score of OntoED merely without event correlation inference (*e.g.*, 10.9%↓) drops more seriously than that without event ontology learning (*e.g.*, 6.6%↓), and the phenomenon is more obvious in few-shot and zero-shot evaluation (*e.g.*, 10.9%↓ v.s. 15.9%↓ and 28.1%↓). This illustrates that event correlation inference is more necessary in OntoED, as it establishes more correlations among event types, thereby knowledge can be propagated more adequately, especially from data-rich to data-poor events.

5.2 Error Analysis

We further conduct error analysis and provide some representative examples. (1) One typical error relates to similar event-event structures in the event ontology. As OntoED considers event correlations,
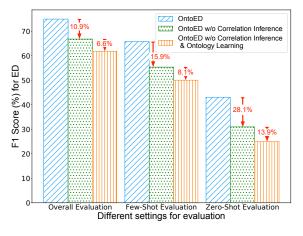

Figure 5: Effect of different modules in OntoED.

event types with similar neighbor triples can be indistinguishable. For example, *Robbery* and *Kidnapping* have the same supertype *Crime*, and they both have the neighbor triples of (∗, CAUSE, *Arrest*). (2) The second error relates to wrong instance relations. As the instance relation extraction directly influence the establishment of event correlations, wrong instance relations will cause error propagation. (3) The third error relates to the same event mention for different event types. For example, '*Of the 126 people aboard, 47 died and 74 sustained serious injuries.*' both mentions *Die* and *Injure*.

6 Conclusion and Future Work

This paper proposes a novel event detection framework with ontology embedding called *OntoED*. We revisit the ED task by linking each event instance to a specific type in a target event ontology. To facilitate the linkage, we enrich the event ontology with event-event relations, such as temporal, causal and hierarchical correlation, and induce more event correlations based on existing ones. The key insight is that event ontology can help to reduce model dependence on instance data, especially in low-resource scenarios. As data-rich event types can propagate correlation knowledge to data-poor ones, and new event types can establish linkages to the event ontology. We demonstrate the effectiveness of OntoED in three settings: overall, few-shot as well as zero-shot, and experiments show that OntoED excels previous methods with great robustness.

In the future, we intend to extend our work in several aspects. First, we would improve the event ontology and consider more event correlations. Second, we would explore if low-resource ED can also boost to identify event correlation. Third, we would develop more neuro-symbolic methods for ED.

Acknowledgments

Broader Impact Statement

A broad goal of event detection is to extract structured knowledge from unstructured texts to facilitate knowledge acquisition. For example, it is valuable in the medical domain and provides social benefits to analyze dispensatory details as well as electronic health records. Furthermore, a solid ED system can also be applied to many society issues, such as anti-terrorist and public opinion analysis.

In this paper, we present a new dataset `OntoEvent` for ED with event-event correlations. The event data are all collected from existing datasets (*i.e.*, ACE 2005) or open source databases (*e.g.*, Wikipedia), and the annotation are generated from existing models with citations. In experiments, we detailedly describe how to evaluate the newly-proposed `OntoEvent` and provide specific analysis. The code and dataset are both available.

Our approach to ED can leverage only a few event corpus to establish the linkage between event types and event instances *w.r.t.* event correlations. In addition, this work is also a brand-new attempt to combine information extraction and symbolic reasoning, based on ontology embedding. Our intention is to develop an ontology-based ED system for the NLP community, and wish our innovation can become a small step in this direction.

References

Collin F Baker. 2014. Framenet: A knowledge base for natural language processing. In *Proceedings of Frame Semantics in NLP: A workshop in honor of Chuck Fillmore (1929-2014)*, pages 1–5.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL (1)*, pages 167–176. The Association for Computer Linguistics.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *EMNLP (Findings)*, pages 2329–2339. Association for Computational Linguistics.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *WSDM*, pages 151–159. ACM.

Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z. Pan, and Huajun Chen. 2019. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In *WWW (Companion Volume)*, pages 678–685. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *EMNLP (1)*, pages 671–683. Association for Computational Linguistics.

Rujun Han, Yichao Zhou, and Nanyun Peng. 2020. Domain knowledge empowered structured neural net for end-to-end event temporal relation extraction. In *EMNLP (1)*, pages 5717–5729. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *ACL (1)*, pages 2160–2170. Association for Computational Linguistics.

Nikhil Ketkar. 2014. Stochastic gradient descent. *Optimization*.

Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. 2020. Extensively matching for few-shot learning event detection. In *NUSE@ACL*, pages 38–45. Association for Computational Linguistics.

Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. 2019. Biomedical event extraction based on knowledge-driven tree-lstm. In *NAACL-HLT (1)*, pages 1421–1430. Association for Computational Linguistics.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare R. Voss. 2020. Connecting the dots: Event graph schema induction with path language modeling. In *EMNLP (1)*, pages 684–695. Association for Computational Linguistics.

Xin Liang, Dawei Cheng, Fangzhou Yang, Yifeng Luo, Weining Qian, and Aoying Zhou. 2020. F-HMTC: detecting financial events for investment decisions based on neural hierarchical multi-label text classification. In *IJCAI*, pages 4490–4496. ijcai.org.

Rinaldo Lima, Bernard Espinasse, and Fred Freitas. 2018. Ontoilper: an ontology- and inductive logic programming-based system to extract entities and relations from text. *Knowl. Inf. Syst.*, 56(1):223–255.

Rinaldo Lima, Bernard Espinasse, and Fred Freitas. 2019. A logic-based relational learning approach to relation extraction: The ontoilper system. *Eng. Appl. Artif. Intell.*, 78:142–157.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *ACL*, pages 7999–8009. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *EMNLP (1)*, pages 1641–1651. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *EMNLP*, pages 1247–1256. Association for Computational Linguistics.

Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang, and Huajun Chen. 2021. Mlbinet: A cross-sentence collective event detection network. In *ACL*. Association for Computational Linguistics.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *HLT-NAACL*, pages 300–309. The Association for Computational Linguistics.

Maria Nefeli Nikiforos, Spiridon Vergis, Andreana Stylidou, Nikolaos Augoustis, Katia Lida Kermanidis, and Manolis Maragoudakis. 2020. Fake news detection regarding the hong kong events from tweets. In *AIAI Workshops*, volume 585 of *IFIP Advances in Information and Communication Technology*, pages 177–186. Springer.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *ACL (1)*, pages 2278–2288. Association for Computational Linguistics.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *EMNLP*, pages 392–402. The Association for Computational Linguistics.

Shirong Shen, Guilin Qi, Zhen Li, Sheng Bi, and Lusheng Wang. 2020. Hierarchical chinese legal event extraction via pedal attention mechanism. In *COLING*, pages 100–113. International Committee on Computational Linguistics.

Shirong Shen, Tongtong Wu, Guilin Qi, Yuan-Fang Li, Gholamreza Haffari, and Sheng Bi. 2021. Adaptive knowledge-enhanced bayesian meta-learning for few-shot event detection. In *Findings of ACL*. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *ACL*, pages 5887–5897. Association for Computational Linguistics.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020a. Joint constrained learning for event-event relation extraction. In *EMNLP (1)*, pages 696–706. Association for Computational Linguistics.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *NAACL-HLT (1)*, pages 998–1008. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020b. MAVEN: A massive general domain event detection dataset. In *EMNLP (1)*, pages 1652–1671. Association for Computational Linguistics.

Xing David Wang, Leon Weber, and Ulf Leser. 2020c. Biomedical event extraction as multi-turn question answering. In *LOUHI@EMNLP*, pages 88–96. Association for Computational Linguistics.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: event adversarial neural networks for multi-modal fake news detection. In *KDD*, pages 849–857. ACM.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *EMNLP/IJCNLP (1)*, pages 5765–5769. Association for Computational Linguistics.

Wenlin Yao, Zeyu Dai, Maitreyi Ramaswamy, Bonan Min, and Ruihong Huang. 2020. Weakly supervised subevent knowledge acquisition. In *EMNLP (1)*, pages 5345–5356. Association for Computational Linguistics.

Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Iteratively learning embeddings and rules for knowledge graph reasoning. In *WWW*, pages 2366–2377. ACM.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *ACL*, pages 7579–7589. Association for Computational Linguistics.

A Hierarchical Event Type Schema (Part 1)

SuperType	SubType	#Instance
Business	Collaboration	272
	Declare-Bankruptcy	95
	End-Org	115
	Merge-Org	41
	Start-Org	162
Cause-Effect	Causation	3387
	Cause-Change-Of-Position-On-A-Scale	1138
	Cause-Change-Of-Strength	1180
	Cause-To-Amalgamate	374
	Cause-To-Be-Included	850
	Cause-To-Make-Progress	448
	Influence	812
Commerce	Carry-Goods	62
	Commerce-Buy	64
	Commerce-Pay	114
	Commerce-Sell	153
	Manufacturing	411
Conflict	Attack	3550
	Bearing-Arms	115
	Besieging	312
	Conquering	1754
	Defending	923
	Escaping	956
	Hostile-Encounter	3620
	Killing	2030
	Military-Operation	1257
	Protest	185
	Quarreling	215
	Releasing	143
	Rescuing	162
	Revenge	64
	Sending	512
	Terrorism	309
	Use-Firearm	538
	Violence	419
Contact	Broadcast	638
	Come-Together	400
	Communication	582
	Contact	237
	Correspondence	197
	Telling	195
Crime	Kidnapping	109
	Robbery	70
	Theft	24
Justice	Acquit	27
	Appeal	125
	Arrest	379
	Committing-Crime	265
	Convict	260
	Criminal-Investigation	277
	Execute	70
	Extradition	14
	Fine	84
	Justifying	34
	Legal-Rulings	241
	Pardon	10
	Prison	69
	Release-Parole	95
	Sentence	321
	Sue	196
Life	Award	71
	Be-Born	128
	Bodily-Harm	1255
	Breathing	7
	Cure	88
	Death	1226
	Divorce	65
	Education-Teaching	132
	Marry	205
	Name-Conferral	760
	Recovering	337

Table 7: Hierarchical event types in `OntoEvent`.

B Hierarchical Event Type Schema (Part 2)

SuperType	SubType	#Instance
Movement	Arriving	1542
	Body-Movement	140
	Departing	488
	Motion-Directional	869
	Placing	815
	Transport-Artifact	241
	Transport-Person	446
	Traveling	937
Natural-Disaster	Catastrophe	3974
	Damaging	1227
	Destroying	1445
Personnel	Change-Of-Leadership	933
	Elect	577
	Employment	156
	End-Position	887
	Nominate	45
	Start-Position	506
Proecess	Confronting-Problem	225
	Process-End	1730
	Process-Start	3275
	Resolve-Problem	157
Transaction	Earnings-And-Losses	1104
	Exchange	79
	Getting	1067
	Giving	695
	Receiving	362
	Renting	21
	Supply	618
	Transaction	50

Table 8: Hierarchical event types in `OntoEvent`.

C Overview of Temporal & Causal Event-Event Correlations

E-E Relation	Head Event Type	Tail Event Type
BEFORE	Personnel.Start-Position	Personnel.End-Position
	Personnel.Nominate	Personnel.Elect
	Commerce.Commerce-Sell	Commerce.Commerce-Buy
	Commerce.Manufacturing	Commerce.Carry-Goods
	Life.Marry	Life.Divorce
	Life.Be-Born	Life.Name-Conferral
	Justice.Arrest	Justice.Prison
	Justice.Sue	Justice.Criminal-Investigation
	Justice.Criminal-Investigation	Justice.Legal-Rulings
	Transaction.Transaction	Transaction.Earnings-And-Losses
	Proecess.Confronting-Problem	Proecess.Resolve-Problem
	Justice.Justifying	Justice.Committing-Crime
	Justice.Convict	Justice.Execute
	Justice.Convict	Justice.Fine
	Justice.Convict	Justice.Extradition
	Justice.Convict	Justice.Sentence
	Justice.Sentence	Justice.Release-Parole
	Justice.Acquit	Justice.Pardon
AFTER	Movement.Arriving	Movement.Transport-Artifact
	Movement.Arriving	Movement.Departing
	Movement.Arriving	Movement.Transport-Person
	Business.End-Org	Business.Start-Org
	Life.Death	Life.Be-Born
	Life.Cure	Life.Bodily-Harm
	Proecess.Process-End	Proecess.Process-Start
	Justice.Appeal	Justice.Sue
	Conflict.Escaping	Conflict.Besieging
	Conflict.Bearing-Arms	Conflict.Use-Firearm
EQUAL	Commerce.Commerce-Pay	Commerce.Commerce-Buy
	Business.Declare-Bankruptcy	Business.End-Org
	Business.Merge-Org	Business.Collaboration
	Transaction.Getting	Transaction.Receiving
	Transaction.Giving	Transaction.Supply
	Transaction.Renting	Transaction.Exchange
	Movement.Traveling	Movement.Transport-Person
	Natural-Disaster.Damaging	Natural-Disaster.Destroying
	Movement.Body-Movement	Movement.Traveling
	Life.Cure	Life.Recovering
	Justice.Extradition	Justice.Legal-Rulings
	Conflict.Revenge	Conflict.Hostile-Encounter
	Conflict.Protest	Conflict.Quarreling
	Conflict.Protest	Conflict.Use-Firearm
	Conflict.Protest	Conflict.Violence
	Conflict.Protest	Conflict.Attack
	Conflict.Protest	Conflict.Killing
	Conflict.Protest	Conflict.Besieging
	Conflict.Protest	Conflict.Conquering
	Conflict.Protest	Conflict.Defending
CAUSE	Cause-Effect.Causation	Cause-Effect.Influence
	Natural-Disaster.Catastrophe	Natural-Disaster.Damaging
	Conflict.Attack	Life.Bodily-Harm
	Conflict.Killing	Life.Death
CAUSEDBY	Justice.Arrest	Crime.Kidnapping
	Justice.Arrest	Crime.Robbery
	Justice.Arrest	Crime.Theft
	Justice.Arrest	Conflict.Attack
	Justice.Arrest	Conflict.Killing

Table 9: Overview of temporal & causal event-event correlations in `OntoEvent`.