HAEE: Low-Resource Event Detection with Hierarchy-Aware Event Graph Embeddings

Guoxuan Ding[1,2], Xiaobo Guo[1(✉)], Gaode Chen[1,2], Lei Wang[1], and Daren Zha[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{dingguoxuan,guoxiaobo,chengaode,wanglei,zhadaren}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. The event detection (ED) task aims to extract structured event information from unstructured text. Recent works in ED rely heavily on annotated training data and often lack the ability to construct semantic knowledge, leading to a significant dependence on resource. In this paper, we propose a hierarchy-aware model called HAEE by constructing event graph embeddings. We utilize two relations (*cause* and *subevent*) to help model events on two dimensions of polar coordinates, so as to distinguish events and establish event-event relations. Specifically, events under the *cause* relation are constructed at the same level of the hierarchy through rotation, while events under the *subevent* relation are constructed at different levels of the hierarchy through modulus. In this way, coexistence and interactions between relations in time and space can be fully utilized to enhance event representation and allow the knowledge to flow into the low-resource samples. The experiments show that HAEE has high performance in low-resource ED task, and the analysis of different dimensions of embeddings proves that HAEE can effectively model the semantic hierarchies in the event graph.

Keywords: Event detection · Low resource · Hierarchy-aware · Event graph

1 Introduction

Event detection [14] (ED) is an important task in natural language processing, as it aims to extract structured event information from unstructured text to support downstream tasks such as question answering systems [2,12], information retrieval [11,33], knowledge graph construction [1,28] and so on. For example, in the sentence "Stewart's first marriage to Alana Hamilton lasted about five years.", the event detection task requires identifying the event type "marry" where the word "marriage" triggers the event.

T. R. Payne et al. (Eds.): ISWC 2023, LNCS 14265, pp. 61–79, 2023.
https://doi.org/10.1007/978-3-031-47240-4_4

There are two major challenges in practical applications of ED. One such challenge is the problem of data scarcity [4]. ED requires a large amount of labeled data to train models. However, in real-world scenarios, some event types have limited samples, which can hinder models from effectively extracting event information. Therefore, there is a need to explore techniques that can facilitate learning knowledge from limited or even absent data.

On the other hand, learning existing knowledge is no longer a difficult problem due to the development of large-scale language models. Instead, constructing knowledge with semantic features is a more challenging problem [34]. Training models to capture profound semantic associations and extract potential information can enhance ED model performance. Therefore, it is necessary to investigate hierarchy-aware approaches for ED task.

Using more abstract event relations as an extra knowledge is a promising direction in low-resource ED scenarios. By learning event-event relations, it can provide a higher-level understanding of events that can be used to build better event representations and guide ED models. OntoED [5] is a classic and important model that formulates ED as an event ontology population task. By introducing three types of relations (*temporal*, *cause*, and *subevent*), it establishes associations between events to let the information flow into the low-resource events. To some extent, it addresses the problem of data scarcity. However, OntoED still has some limitations. Event relations can coexist and interact simultaneously in time and space, but 1) OntoED only considers and calculates each relation separately without discussing whether there are interactions between them. Additionally, 2) OntoED only involves one level of *subevent* without considering whether it has multi-level feature, which may result in the loss of relation knowledge.

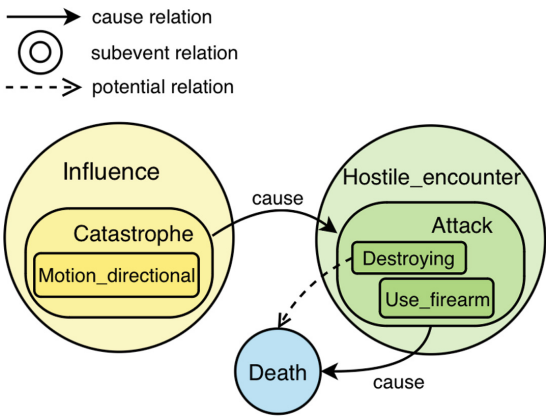


Fig. 1. Event graph with *cause* and *subevent*, potential relations are invisible in the knowledge set, but are real and can be mined. Color zones indicate events that are in the same time and space and are related through *subevent* (Color figure online)

To tackle these problems, we build a knowledge set with the *cause* and *subevent* relations and consider their characteristics of correlation. We do not consider *temporal* because it is relatively weak in terms of logical correlation. As illustrated in Fig. 1, *Cause* (e.g., *Attack* $\xrightarrow{cause}$ *Death*) means head events lead to the occurrence of tail events, and *subevent* (e.g., *Attack* $\xrightarrow{subevent}$ *Destroying*) indicates that child events are components of parent events, note that it can be a multi-level relation due to the transitivity of *subevent*. By modeling both *cause* and *subevent* together, we can observe the coexistence and interaction between relations: Events occurring in the same time and space may have an effect on events that occur in another time and space. Building *cause-subevent* relations not only makes the correlation closer but also helps to explore potential connections between events.

In this paper, we propose a novel hierarchy-aware model called HAEE (**H**ierarchy-**A**ware with **E**vent graph **E**mbedding) by combining *cause-subevent* relations and event graph embeddings. HAEE is expected to distinguish events and establish event-event relations by rotation and modulus, which operate at the same level and different levels of the hierarchy, respectively. For *cause*, we adopt a rotation-based approach [23] in which the event pairs are set at opposite positions on the circle. For *subevent*, we adopt a contrastive learning approach [20] where the distance between child events is as close as possible compare to the distance from child events to their parents. We perform convolution operations on embeddings to enhance the representation of events. In this way, further and hierarchical relation knowledge between events can be learned. We calculate multiple loss functions using Uncertainty to Weigh Losses [9] to obtain the final loss, which helps us balance the importance of each loss in the final ED task.

Applying these ideas, HAEE can start from a tiny amount of labeled samples and gradually discover more and more potential knowledge contained in event graph (e.g., *Destroying* $\xrightarrow{relation}$ *Death*), thereby improving the performance of low-resource ED task. The experiments demonstrate that HAEE outperforms existing models in data-scarce scenarios. To further analyze the effectiveness of our model, we conduct a module-wise analysis that examines each component of the model and its contribution to the overall performance. We find that our model has good modeling and semantic hierarchies on event embeddings, which is a key factor in accurately extracting events from text.

In summary, our contributions are as follows:

- We propose a novel hiearchy-aware model called HAEE. By constructing the knowledge set with *cause* and *subevent*, it can discover more potential information and allow the knowledge to flow into the low-resource events to solve the problem of ED in low-resource scenarios.
- To fully utilize coexistence and interactions between relations in time and space, we leverage *cause-subevent* relations by learning event embedding in the polar coordinates, which lead to a hierarchical event graph on dimensions of rotation and modulus.
- The experiments demonstrate that HAEE can achieve better performance in low-resource ED task. We conduct a module-wise analysis that examines

each component of the model and its contribution to the overall performance, showing that our model has good modeling and semantic hierarchies on event graph.

2 Related Work

In low-resource scenarios, exploiting stronger models and data augmentation are two main directions to improve the ED task [4]. Exploiting stronger models is to design a model that can improve its learning ability, thereby making full use of a small amount of samples and reducing its dependence on data. Data augmentation is expected to enhance the quality of samples as well as contribute more semantic representations.

Compare to traditional machine learning methods that have difficulty handling complex event structures and semantic relations, most recent ED works are based on neural networks [16,18], which can automatically learning useful features from data. DMCNN [3] is designed with a vocabulary expression model for capturing sentence-level clues by extracting lexical and sentence-level features using CNNs. Nguyen et al. [19] propose an improved non-continuous CNN that uses a joint-based paradigm to learn potential feature representations. JRNN [21] is presented with a bidirectional RNNs to link grammar-related vocabulary without using vocabulary itself. Feng et al. [8] use LSTM's long-term memory to capture text sequence information at the document level. However, both RNN-based and CNN-based models only consider the semantic information of individual samples without considering their relations with other samples. The powerful text learning ability enables the pre-training models [31,36] to win a widely attention and achieve a rapid development. ED tasks have shifted from constructing lexical representations for identifying triggers to representing samples as a whole. For example, CLEVE [27] is designed with a contrastive learning method based on pre-training frameworks that makes full use of event knowledge in large amounts of unsupervised data to construct semantic structures. While these techniques have shown promise, they encounter obstacles such as the requirement for significant quantities of annotated data and the inability to explore more suitable strategies for learning with limited data.

Another research direction to improve model performance is through data augmentation [17]. PathLM [13] provides an event pattern representation semantic framework that connects events together through multiple paths. OntoED [5] uses an ontology-based event construction method to enhance data by introducing multiple event relations. K-HPN [6] embeds *cause* into a knowledge-aware hypersphere prototype network. Ye et al. [32] use external knowledge graphs to transform structured knowledge into text to address the problem of knowledge scarcity. However, these techniques have primarily focused on enhancing samples at a local level, without improving sample quality from a higher-level perspective. Recent works [30,35] on entity graphs provide the possibility of constructing event knowledge in a higher-dimensional space. By doing so, it may facilitate the investigation of events from a more abstract and holistic perspective.

3 Methodology

In this section, we formally present the proposed model HAEE. We first introduce a) *Event Detection* as the main task of our model. Afterwards, we introduce b) *Graph Embedding* including rotation and modulus part for hierarchy-aware event graph modeling. Finally, we introduce c) *Embedding Convolution* and d) *Loss Function* to complete the rest of our model.

3.1 Problem Formulation

For an ED task, the given input includes a sample set $\mathcal{T} = \{X_i | i = 1, \ldots, N\}$ and an event type set $\mathcal{E} = \{e_i | i = 1, \ldots, M\}$. Each sample X_i in $\mathcal{T}$ is a token sequence $\{x_i^1, \ldots, x_i^L\}$ with trigger x_i^k, where L is the maximum length of the sequence, and k is the position of the trigger in the sequence. The event type set $\mathcal{E}$ consists of different event types e, and each X_i belongs to an event type label e_i. The goal of the event detection task is to predict the event type label corresponding to each sample.

Event relation refer to the higher-level knowledge and more abstract connections between events. The relation set $\mathcal{R}$ includes the *cause* set $\mathcal{R}_c$ and the *subevent* set $\mathcal{R}_s$. The *cause* set $\mathcal{R}_c$ includes *cause* r_{cause} and *caused by* $r_{causedby}$, while the *subevent* set $\mathcal{R}_s$ includes *subevent* $r_{subevent}$ and *superevent* $r_{superevent}$.

Based on the event type set $\mathcal{E}$ and the relation set $\mathcal{R}$, the knowledge set $\mathcal{K}$ about events and relations is constructed, consisting of triples $(e_h, r, e_t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where e_h and e_t respectively refer to the head event and tail event under the relation r.

3.2 Model Overview

In this paper, we propose a general model called HAEE with three modules: event detection, graph embedding and embedding convolution. The key information for each module is shown in Fig. 2. Graph embedding contains rotation part and modulus part, and each part including event detection can be divided into three steps: graph modeling, score mapping, and loss calculation.

For event detection, we obtain the contextual representation of the sample from BERT and calculate the probability between sample and its event to form the event detection loss.

For graph embedding, we model event embeddings in the polar coordinates, find events that are in the *cause* or *subevent* relation with the query event from the knowledge set and calculate corresponding distance scores to form the loss function.

For embedding convolution, we convolve distant events in a certain weight to learn further and hierarchical relation knowledge.

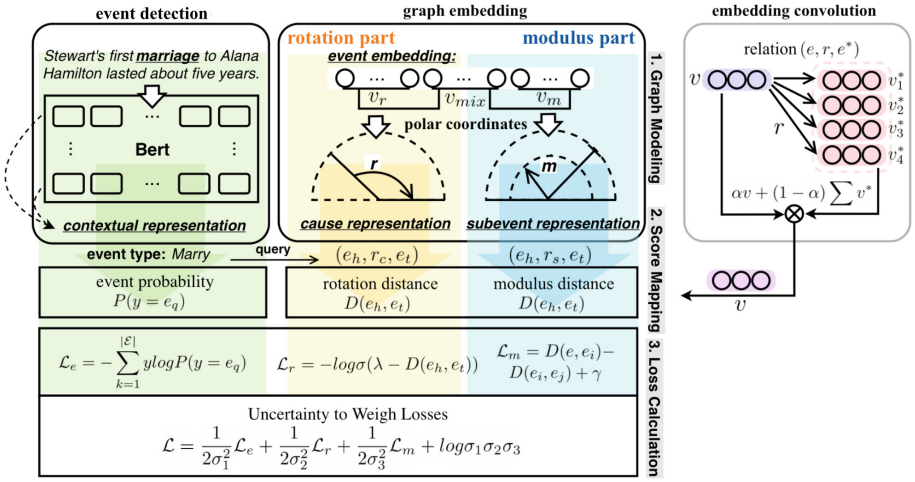


Fig. 2. Detailed example for the process of HAEE. Green zone represent the event detection. Yellow zone represent the rotation part in graph embedding. Blue zone represent the modulus part in graph embedding. Each space contains three steps: **Step 1 (Graph Modeling)** form sample representation and model event embedding to graph. **Step 2 (Score Mapping)** calculate event probability, rotation distance and modulus distance. **Step 3 (Loss Calculation)** form three loss functions and combine them to obtain the final loss (Color figure online)

3.3 Event Detection

To obtain the representation of each X_i, We use a pre-trained BERT model [7]. Specifically, we take the average of the word embeddings from the first and last layers of BERT's transformer to form the sentence representation. This approach maximally preserves semantic information for words and sentences in original samples, ultimately enhancing event representation [22,29].

We share the same formula to that of OntoED [5] for the calculation of event probability. For a given token sequence $\boldsymbol{X}_i$, the probability of it belonging to event type e_k is

$$P(y = e_k) = \frac{exp(-||\boldsymbol{X}_i - \boldsymbol{v}_k||)}{\sum_{j=1}^{|\mathcal{E}|} exp(-||\boldsymbol{X}_i - \boldsymbol{v}_j||)}, \tag{1}$$

and the event detection loss is

$$\mathcal{L}_e = -\sum_{k=1}^{|\mathcal{E}|} ylogP(y = e_q). \tag{2}$$

where e_q is the ground-truth label for $\boldsymbol{X}_i$.

3.4 Graph Embedding

Based on the characteristics of *cause* and *subevent*, we choose two dimensions of polar coordinates to model them. Polar coordinates [35] is a position system

in a two-dimensional space, consisting of rotation and modulus. By determining the rotation and modulus of a point, we can determine its position in the two-dimensional space.

To map event type to polar coordinates, we divide the embedding $\boldsymbol{v}$ of an event type e into rotation embedding $\boldsymbol{v}_r$, modulus embedding $\boldsymbol{v}_m$, and confuse embedding $\boldsymbol{v}_{mix}$ of equal size. The addition of the confuse embedding is to synthesize the characteristics of both dimensions and enhance the generalization ability of event representation. Specifically, we map $[\boldsymbol{v}_r, \boldsymbol{v}_{mix}]$ to the rotation as the *cause* representation and $[\boldsymbol{v}_m, \boldsymbol{v}_{mix}]$ to the modulus as the *subevent* representation. This modeling approach enhances interactions between two relations, helps discover potential relations between events, and ultimately enhances the model's semantic understanding ability.

Rotation Part. *Cause* refer to the occurrence of head events leading to the occurrence of tail events, indicating a obvious *cause* between them. Through the *cause* set R_c, we model events on the rotation of polar coordinates. Specifically, to represent *cause* between two events, we learn their opposite positions on the circle.

Given a query event e and *cause* r_c, we find all events e_c in the knowledge set $\mathcal{K}$ that satisfy (e, r_c, e_c) to form an event type set $\mathcal{E}_c$. For embeddings $\boldsymbol{v}$ and $\boldsymbol{v}_c$ of event types e and e_c, respectively, we have

$$(\boldsymbol{v} - \boldsymbol{v}_c)\, mod\, 2\pi = \pi, \tag{3}$$

where $\boldsymbol{v}, \boldsymbol{v}_c \in [0, 2\pi)$. We define the distance between events as

$$D(e_h, e_t) = \left\| sin((\boldsymbol{v}_h - \boldsymbol{v}_t + \pi)/2) \right\|_1. \tag{4}$$

The rotation loss [23] is then defined as

$$\mathcal{L}_r = \sum_{e_c \in \mathcal{E}_c} -log\sigma(\lambda - D(e, e_c)), \tag{5}$$

where $\sigma(\cdot)$ represents the sigmoid function and λ is a fixed threshold under rotation.

Modulus Part. *Subevent* means head events contain tail events in time and space, where tail events are components of head events. For the *subevent* set R_s, we assume that *subevent* have multi-level feature, meaning that events at the same level should have similar properties, while there is a certain distance between events and their parents. The *subevent* has transitivity, i.e., $(a, r, b), (b, r, c), then(a, r, c)$. We model this on the modulus of polar coordinates to represent events level. To achieve this goal, we use contrastive learning [20] to make events at the same level closer and events at different levels farther apart.

Given a query event e and *subevent* r_s, we find all events e_s in the knowledge set $\mathcal{K}$ that satisfy (e, r_s, e_s) to form an event type set $\mathcal{E}_s$. We define the different-level event pairs set $\mathcal{P}_d = \{(e, e_i) | e_i \in \mathcal{E}_s\}$ and the same-level event pairs set

$\mathcal{P}_s = \{(e_i, e_j)|e_i, e_j \in \mathcal{E}_s, i \neq j\}$. The distance between events as

$$D(e_h, e_t) = \|\boldsymbol{v}_h - \boldsymbol{v}_t\|_2 . \tag{6}$$

Then the modulus loss is defined as

$$\mathcal{L}_m = \sum_{(e,e_i)\in\mathcal{P}_d} \sum_{(e_i,e_j)\in\mathcal{P}_s} max(D(e, e_i) - D(e_i, e_j) + \gamma, 0). \tag{7}$$

where γ represents an interval threshold under modulus.

3.5 Embedding Convolution

For an event relation chain $(a, r, b), (b, r, c) \in \mathcal{K}$, and $(a, r, c) \notin \mathcal{K}$, the event c may play a role in learning the relation (a, r, b) for the event a. To enhance the representation of events, obtain further and hierarchical knowledge in relation learning, given a query event e, a relation r, and the event relation knowledge (e, r, e_r), we find all events e^* in the knowledge set $\mathcal{K}$ that satisfy (e_r, r, e^*) to form an event set $\mathcal{E}^*$. We then calculate embedding of e_r as

$$\boldsymbol{v} = \alpha\boldsymbol{v} + (1 - \alpha)\frac{1}{|\mathcal{E}^*|}\sum_{e^*\in\mathcal{E}^*} \boldsymbol{v}^*, \tag{8}$$

where $\boldsymbol{v}^*$ is a embedding of e^* and $\alpha \in [0, 1]$ is the weight for distant events. Through this approach, when calculating relations between events e and e_r, more knowledge can be learned.

3.6 Loss Function

For each event that undergoes relation calculation, its relation distribution may not be uniform, resulting in some events having many relations while others have few. To balance three losses, we introduce the Uncertainty to Weigh Losses multi-task optimization strategy [9]. By introducing weight coefficients into losses and adjusting their weights based on the prediction uncertainty of different module, we can express the final loss function as

$$\mathcal{L} = \frac{1}{2\sigma_1^2}\mathcal{L}_e + \frac{1}{2\sigma_2^2}\mathcal{L}_r + \frac{1}{2\sigma_3^2}\mathcal{L}_m + log\sigma_1\sigma_2\sigma_3, \tag{9}$$

where $\sigma_1, \sigma_2, \sigma_3$ are learnable parameters.

4 Experiments

In this section, we first introduce two datasets (OntoEvent and MAVEN-Few) and the baseline methods. Then we report the experimental results on ED task in different scenarios including a) *Overall Evaluation* and b) *Low-resource Evaluation.*

4.1 Datasets

Two datasets are used for the model evaluation: the OntoEvent dataset and the selected MAVEN-ERE dataset, called MAVEN-Few. Both of these datasets are composed of English samples. OntoEvent [5] is an event dataset with event relations, used to demonstrate that introducing ontology helps with event detection. MAVEN-ERE [24] is a unified large-scale dataset for event relation detection based on the original MAVEN [26] dataset, containing a large number of event types and their relations. To facilitate comparison with OntoED, we select event types from MAVEN-ERE that exist in OntoEvent, which has 71 event types, and their corresponding samples to conduct MAVEN-Few dataset for the evaluation. For the construction of knowledge set, we selected all involved events relations from MAVEN-ERE, which has 277 *cause* and 83 *subevent* for experiment. Compared to OntoEvent's 9 *cause* and no multi-level *subevent* involved, relations in MAVEN-Few are richer and it may be much more easier to mine more potential associations between events. In this paper, we will use these two datasets and the knowledge set to evaluate the performance of HAEE (Table 1).

Table 1. Statistics of OntoEvent and MAVEN-Few datasets. (Doc: document, Train: training set, Valid: validation set, Test: test set, Class: event types, Caus-Rel: *cause* relations, Sub-Rel: muti-level *subevent* relations.)

Dataset	#Doc	#Train	#Valid	#Test	#Class	#Caus-Rel	#Sub-Rel
OntoEvent [5]	4115	48436	6055	6055	100	9	–
MAVEN-Few	–	4416	552	551	71	277	83

4.2 Experiments Settings

We test the results of the validation set under different dimensions in {50,100,200, 500}, and finally choose 100 as the dimension of event embedding. The maximum token sequence length of the training sample is set to 128, and a dropout ratio of 0.2 is set to prevent overfitting. The learning rate is 1×10^3, and the initial values of uncertainty values $\sigma_1, \sigma_2, \sigma_3$ are set to -0.5. In terms of hyperparameter selection, we use a grid search on the validation set. Specifically, convolution weight $\alpha \in [0.2, 0.8]$ with a step size of 0.2, rotation threshold $\lambda \in [0.06, 0.12]$ with a step size of 0.02, and modulus threshold $\gamma \in [0.06, 0.12]$ with a step size of 0.02 are searched on the validation set to find optimal values for α, λ, γ, which are ultimately set to 0.4, 0.08, and 0.08 respectively for testing purposes. We randomly divide dataset into training set (80%), validation set (10%), and test set (10%), using the SGD optimizer [10] with the batch size of 42 samples per training iteration for a total of 5000 iterations to obtain the final result. We evaluate the performance of model with Precision, Recall and F1 Score for sample classification to the correct event label [5].

4.3 Baselines

For overall evaluation, we compare the proposed model with the following baseline methods in ED task.

- AD-DMBERT [25] is an adversarial training model that enhances distantly supervised event detection models and automatically constructs more diverse and accurate training data for semi-supervised event detection models.
- OneIE [15] is a joint neural model for information extraction. It explicitly models cross-subtask and cross-instance inter-dependencies and predicts the result as a unified graph instead of isolated knowledge elements.
- PathLM [13] is an auto-regressive language model and is designed to induce graph schemas that connect two event types through multiple paths involving entities.
- OntoED [5] formulates ED as an event ontology population task, and inferred more enriched samples and event-event relations with ontology learning.

For low-resource evaluation, we simply select OntoED [5] model for comparison due to its superior performance compared to other models and its utilization of event relations as extra knowledge, similar to HAEE.

4.4 Results

Overall Evaluation. From Table 2, it can be seen that on the OntoEvent dataset, HAEE obtains a better result than other models on all three indicators, such as BERT-based AD-DMBERT. This indicates the effectiveness of the HAEE framework built on BERT, which can better establish the connection between events. It also has a better performance than graph-based OneIE and PathLM, which only convert sentences into instance graphs and ignore potential relations between events. For ontology-based OntoED, it only explains event relations at the semantic level without considering whether there are interactions between relations and has weak modeling of *cause* and *subevent*. It is due to the comprehensive consideration of both *cause* and *subevent* between events and the implicit associations in the whole event graph in HAEE.

Table 2. Evaluation of event detection with overall OntoEvent dataset. †: results are produced with codes referred to Deng et al. [5]; ‡: results are produced with official implementation.

Model	Precision	Recall	F1 Score
AD-DMBERT† [25]	0.6735	0.7346	0.7189
OneIE† [15]	0.7194	0.6852	0.7177
PathLM† [13]	0.7351	0.6874	0.7283
OntoED‡ [5]	0.7756	0.7844	0.78
HAEE	**0.8882**	**0.8868**	**0.8875**

Low-Resource Evaluation. In low-resource scenarios, HAEE still maintains better performance. From Table 3, it can be seen that when the training samples of the OntoEvent dataset are reduced to 50%, OntoED's F1 score drops by 6.5% while HAEE only drops by 1.3%. When the training set is reduced to 25%, OntoED's F1 score drops by 16% while HAEE only drops by 2.4%. Even when the training set is reduced to 10%, HAEE only drops by 7.6%, with an F1 score of 0.831, which is much higher than OntoED's F1 score of 0.78 on the full training set.

Table 3. F1 Score of event detection on different ratios of OntoEvent and MAVEN-Few training data.

Model	OntoEvent				MAVEN-Few			
	Full	50%	25%	10%	Full	50%	25%	10%
OntoED [5]	0.78	0.7154	0.6198	0.4989	0.7725	0.6034	0.5195	0.2534
HAEE	**0.8875**	**0.8747**	**0.8634**	**0.831**	**0.8722**	**0.8577**	**0.8165**	**0.5993**

In the low-resource MAVEN-Few dataset, HAEE still performs well. The F1 values in each scenario are higher than those of OntoED. In terms of model stability, OntoED has already shown a significant decline when the training set is reduced to 50%, while for HAEE, when the training set is reduced to 50% and 25%, F1 only decreases by 2.5% and 5.6%, respectively. Only when it is reduced to 10%, does F1 show a significant decrease of 27%. However, compared with OntoED, HAEE still has a significant advantage at this point as its F1 value at a data ratio of 25% is higher than that of OntoED on the full dataset by 4.4%.

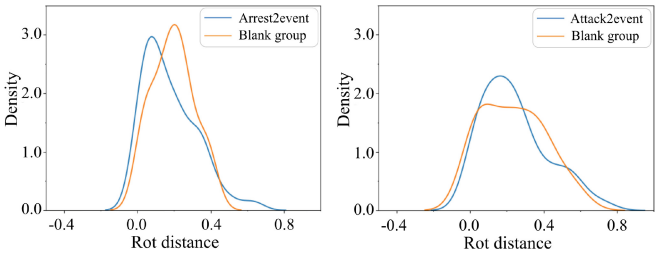
It can be seen that HAEE can maintain great and stable performance in low-resource scenarios while OntoED relies solely on semantic relations, which means that explicit event relations are not enough to support event representation in low-resource scenarios due to not considering implicit interactions between relations. Owing to modeling events in graph, HAEE can fully utilize event relations to enhance event representation in low-resource scenarios, thereby helping the model perform better in event detection task and enhancing its robustness.

5 Analysis

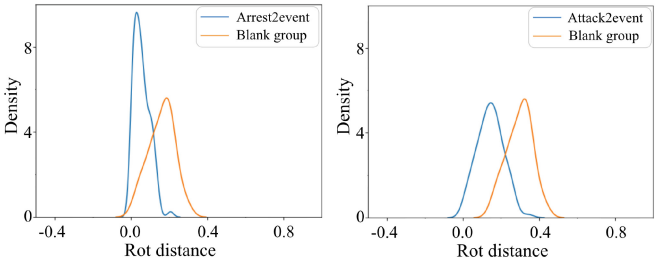
In this section, we analysis the semantic hierarchies and performance of HAEE from different perspectives including a) *Cause Representation*, b) *Subevent Representation* and c) *Ablation Studies*. We also provide possible explanations for our findings.

5.1 Cause Representation

Due to complex one-to-many and many-to-one *cause*, event pairs in relation may not fully reflect opposite position in polar coordinates modeling. To study



(a) 100 dimension.



(b) 500 dimension.

Fig. 3. Rotation distance density of two events (*Arrest* and *Attack*) in different embedding dimensions. Blue lines represent distance to other events. Red lines represent distance to blank control group. (Color figure online)

the effect of *cause* representation on rotation, we select two events, *Attack* and *Arrest*, to calculate their *cause* distance density to other events and set up 30 blank events as a control group. The blank events are only established during model initialization and do not participate in relation calculations. By comparing the distance to other events and the distance to the blank group, it can not only reflect the effect of *cause* calculation on events but also reflect the *cause* information of events themselves based on the blank control group.

Figure 3 shows the *cause* distance density of *Attack* and *Arrest* at 100 and 500 dimensions. It can be seen that as the dimension increases, the variance of blank control group tends to be more similar, indicating that *cause* representation becomes more stable, while event *cause* features become more prominent, means the distinction between events becomes obvious. This proves that the model does have an effect on modeling *cause*.

To further analyze the semantic hierarchy of *cause* representation, we select 20 positive events and 20 negative events from 100 events, as well as the aforementioned 30 blank control events, to form their respective event groups. We calculate the average distance from each event to the positive group, negative group, and blank control group and test the model's clustering effect on positive and negative events in *cause* modeling.

From the Table 4, it can be seen that positive events have a more unified clustering tendency, with 16 positive events tending to be far from negative

Table 4. The average rotation distance from each event to different groups. (p.g: positive group, b.g: blank control group, n.g:negative group. The bold numbers represent groups with a greater distance.)

Positive Events	Distance			Negative Events	Distance		
	p.g	b.g	n.g		p.g	b.g	n.g
come_together	0.45	0.55	**0.59**	destroying	**0.34**	0.23	0.24
elect	0.34	0.42	**0.47**	kidnapping	**0.30**	0.22	0.23
committing_crime	0.19	0.21	**0.28**	violence	**0.28**	0.21	0.22
employment	0.19	0.19	**0.23**	theft	**0.45**	0.32	0.33
award	**0.27**	0.20	0.21	robbery	0.18	0.19	**0.24**
arriving	0.31	0.38	**0.44**	hostile_encounter	0.18	0.20	**0.26**
contact	**0.51**	0.36	0.35	killing	0.23	0.29	**0.37**
recovering	0.21	0.19	**0.22**	terrorism	0.28	0.37	**0.45**
commerce_sell	0.28	0.34	**0.40**	conquering	**0.26**	0.20	0.22
exchange	0.19	0.20	**0.26**	arrest	0.19	0.19	**0.23**
marry	0.37	0.46	**0.51**	divorce	0.18	0.19	**0.25**
cure	0.25	0.30	**0.37**	bodily_harm	**0.38**	0.23	0.25
breathing	0.18	0.20	**0.25**	military_operation	**0.26**	0.20	0.22
communication	0.20	0.19	**0.22**	catastrophe	0.19	0.19	**0.23**
education_teaching	0.18	0.19	**0.24**	prison	**0.58**	0.47	0.46
traveling	0.19	0.21	**0.28**	use_firearm	**0.72**	0.60	0.59
resolve_problem	0.20	0.19	**0.22**	confronting_problem	**0.35**	0.23	0.25
be_born	**0.24**	0.20	0.21	death	0.41	0.53	**0.60**
placing	0.19	0.20	**0.24**	damaging	0.27	0.36	**0.44**
sending	**0.57**	0.42	0.40	revenge	**0.36**	0.24	0.26

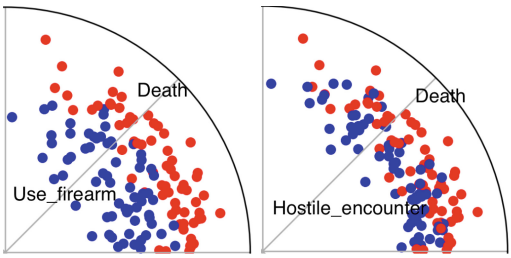
events, and only 4 positive events showing the opposite result. This means that if an event shows obvious positivity, the model's accuracy in predicting the polarity of the event can reach up to 80%. For negative events, the clustering tendency is not particularly obvious, but it can be observed from the data in the table that compared with the blank control group, the clustering effect of events is significant. This means that the farther away an event's *cause* distance is from negative events, the more obvious its unified positivity or negativity is.

Compared with distant event groups, close event groups do not have particularly distinction from blank control group, but for some specific events, they can also effectively distinguish close groups. For example, *Death* has a distance difference of 0.12 between close group and blank group while that of only 0.07 between distant group and blank group.

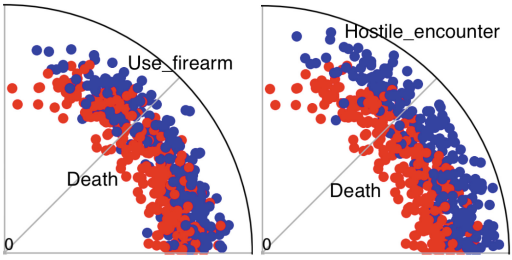
Cause representation calculated through rotation can cluster event polarity. One possible explanation for this phenomenon is that *cause* representation

carry *subevent* information through confuse embedding, and *subevent* have obvious clustering effects on events of the same type through learning distant relation of events. On the other hand, learning through a large number of samples also affects event representation. This ultimately leads to *cause* having polarity clustering effects as well. This means that the model can fully combine the characteristics of *cause-subevent* relations and event samples to enhance event representation, ultimately enhance semantic understanding of events by the model.

5.2 Subevent Representation



(a) 100 dimension.



(b) 500 dimension.

Fig. 4. Modulus distance of two event pairs (*Death*, *Use_firearm*) and (*Death*, *Hostile_encounter*) in different embedding dimensions. Note that the modulus in 100 dimension does not start from 0.

To demonstrate the hierarchy of *subevent* representation, we model other events based on the modulus of *Attack*. Specifically, we calculate the *subevent* distance between *Attack* and other events as the modulus of each event, while compressing the rotation to better demonstrate the effect of representation modeling. We select three events for graph embedding mapping: *Death*, *Use_firearm* and *Hostile_encounter*. It should be noted that *Death* and the other two events do not have a direct *subevent* in the knowledge set and can only indirectly reflect their *subevent* through *Attack*.

As shown in Fig. 4, in different dimensions, it can be seen that *Death* has a clear hierarchical distinction from the other two events. This proves that through

cause information in confuse embedding, model can utilize *cause* to learn more event knowledge in *subevent* and ultimately enhance event representation.

5.3 Ablation Studies

To study the effect of each module on the model, we remove modulus part, rotation part, and use only the basic architecture of the model for testing in different resource scenarios on the OntoEvent [5] dataset. As shown in the Table 5, when there is sufficient data, the improvement of modules on accuracy is not significant. This may be because the sample set itself already contains enough information, which makes it difficult to demonstrate the effects of rotation and modulus. However, when resources are limited, combining rotation and modulus can effectively improve F1 score. It can be observed that under extremely low-resource conditions (10%), using only rotation can increase F1 score by 2.2% compared to not using graph embedding at all. But if we combine rotation and modulus, it can increase F1 score by 7.3%. Also, it can be observed that modulus provides more gain when there is sufficient data, while rotation provides more gain under low-resource conditions. By combining rotation and modulus, HAEE can fully utilize the relations between events to enhance representation.

Table 5. Effect of different modules of HAEE in low-resource scenarios. (rot: rotation part, mod: modulus part).

Model	Full	50%	25%	10%
HAEE	0.8875	0.8747	0.8634	0.831
HAEE w/o mod	0.8696	0.8593	0.8477	0.7808
HAEE w/o rot	0.8801	0.8641	0.8495	0.7578
HAEE w/o mod & rot	0.8741	0.8637	0.8474	0.7579

6 Conclusions and Future Work

This paper proposes a new hierarchy-aware model HAEE, which establishes *cause-subevent* relations by rotation and modulus, and maps event embeddings to polar coordinates to enhance event representation. To learn further and hierarchical relation knowledge, we convolve distant events together. We combine three loss functions by balance the importance of each loss in the final ED task. Through experiments, we demonstrate that HAEE has high performance under low-resource conditions, and analyze the modeling effects of each module, proving that the model has good modeling and semantic effects on event graph. In the future, we intend to enhance our work by more event relations and more complicated structures, and extend it to other information extraction tasks.

Supplemental Material Statement: Source code for HAEE and the MAVEN-Few dataset is available from Github at https://github.com/cdmelon/HAEE. Source code for OntoED and the OntoEvent dataset is available from Github at https://github.com/231sm/Reasoning_In_EE.

References

1. Bosselut, A., Le Bras, R., Choi, Y.: Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4923–4931 (2021)
2. Boyd-Graber, J., Börschinger, B.: What question answering can learn from trivia nerds. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7422–7435. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.662. https://aclanthology.org/2020.acl-main.662
3. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, pp. 167–176. Association for Computational Linguistics (2015). https://doi.org/10.3115/v1/P15-1017. https://aclanthology.org/P15-1017
4. Deng, S., et al.: Low-resource extraction with knowledge-aware pairwise prototype learning. Knowl.-Based Syst. **235**, 107584 (2022)
5. Deng, S., et al.: OntoED: low-resource event detection with ontology embedding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2828–2839. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.acl-long.220. https://aclanthology.org/2021.acl-long.220
6. Deng, S., Zhang, N., Xiong, F., Pan, J.Z., Chen, H.: Knowledge extraction in low-resource scenarios: survey and perspective. arXiv preprint abs/2202.08063 (2022). https://arxiv.org/abs/2202.08063
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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), Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423
8. Feng, X., Huang, L., Tang, D., Ji, H., Qin, B., Liu, T.: A language-independent neural network for event detection. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, pp. 66–71. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/P16-2011. https://aclanthology.org/P16-2011
9. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018, pp. 7482–7491. IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00781. https://openaccess.thecvf.com/content_cvpr_2018/html/Kendall_Multi-Task_Learning_Using_CVPR_2018_paper.html

10. Ketkar, N., Ketkar, N.: Stochastic gradient descent. Deep learning with Python: a hands-on introduction, pp. 113–132 (2017)
11. Kuhnle, A., Aroca-Ouellette, M., Basu, A., Sensoy, M., Reid, J., Zhang, D.: Reinforcement learning for information retrieval. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2669–2672 (2021)
12. Li, F., et al.: Event extraction as multi-turn question answering. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 829–838. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.73. https://aclanthology.org/2020.findings-emnlp.73
13. Li, M., et al.: Connecting the dots: event graph schema induction with path language modeling. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 684–695. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.emnlp-main.50. https://aclanthology.org/2020.emnlp-main.50
14. Li, Q., et al.: A survey on deep learning event extraction: approaches and applications. IEEE Trans. Neural Netw. Learn. Syst. (2022)
15. Lin, Y., Ji, H., Huang, F., Wu, L.: A joint neural model for information extraction with global features. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7999–8009. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.713. https://aclanthology.org/2020.acl-main.713
16. Liu, X., Luo, Z., Huang, H.: Jointly multiple events extraction via attention-based graph information aggregation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 1247–1256. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/D18-1156. https://aclanthology.org/D18-1156
17. Lu, Y., et al.: Text2Event: controllable sequence-to-structure generation for end-to-end event extraction. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2795–2806. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.acl-long.217. https://aclanthology.org/2021.acl-long.217
18. Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, pp. 300–309. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/N16-1034. https://aclanthology.org/N16-1034
19. Nguyen, T.H., Grishman, R.: Modeling skip-grams for event detection with convolutional neural networks. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, pp. 886–891. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/D16-1085. https://aclanthology.org/D16-1085
20. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015, pp. 815–823. IEEE Computer Society (2015). https://doi.org/10.1109/CVPR.2015.7298682
21. Sha, L., Qian, F., Chang, B., Sui, Z.: Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI

Conference on Artificial Intelligence, (AAAI 2018), The 30th Innovative Applications of Artificial Intelligence (IAAI 2018), and The 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI 2018), New Orleans, Louisiana, USA, 2–7 February 2018, pp. 5916–5923. AAAI Press (2018). www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16222
22. Su, J., Cao, J., Liu, W., Ou, Y.: Whitening sentence representations for better semantics and faster retrieval. arXiv preprint abs/2103.15316 (2021). arxiv:2103.15316
23. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019. OpenReview.net (2019). https://openreview.net/forum?id=HkgEQnRqYQ
24. Wang, X., et al.: MAVEN-ERE: a unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. arXiv preprint abs/2211.07342 (2022). arxiv:2211.07342
25. Wang, X., Han, X., Liu, Z., Sun, M., Li, P.: Adversarial training for weakly supervised event detection. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, pp. 998–1008. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/N19-1105. https://aclanthology.org/N19-1105
26. Wang, X., et al.: MAVEN: a massive general domain event detection dataset. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1652–1671. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.emnlp-main.129. https://aclanthology.org/2020.emnlp-main.129
27. Wang, Z., et al.: CLEVE: contrastive pre-training for event extraction. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 6283–6297. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.acl-long.491. https://aclanthology.org/2021.acl-long.491
28. Wu, X., Wu, J., Fu, X., Li, J., Zhou, P., Jiang, X.: Automatic knowledge graph construction: a report on the 2019 ICDM/ICBK contest. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 1540–1545. IEEE (2019)
29. Wu, X., Gao, C., Zang, L., Han, J., Wang, Z., Hu, S.: ESimCSE: enhanced sample building method for contrastive learning of unsupervised sentence embedding. arXiv preprint abs/2109.04380 (2021). arXiv:2109.04380
30. Yang, J., et al.: Learning hierarchy-aware quaternion knowledge graph embeddings with representing relations as 3D rotations. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 2011–2023 (2022)
31. Yang, S., Feng, D., Qiao, L., Kan, Z., Li, D.: Exploring pre-trained language models for event extraction and generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 5284–5294. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/P19-1522. https://aclanthology.org/P19-1522
32. Ye, H., et al.: Ontology-enhanced prompt-tuning for few-shot learning. In: Proceedings of the ACM Web Conference 2022, pp. 778–787 (2022)
33. Zhang, W., Zhao, X., Zhao, L., Yin, D., Yang, G.H.: DRL4IR: 2nd workshop on deep reinforcement learning for information retrieval. In: Proceedings of the 44th

International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2681–2684 (2021)
34. Zhang, W., et al.: Iteratively learning embeddings and rules for knowledge graph reasoning. In: Liu, L., et al. (eds.) The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019, pp. 2366–2377. ACM (2019). https://doi.org/10.1145/3308558.3313612
35. Zhang, Z., Cai, J., Zhang, Y., Wang, J.: Learning hierarchy-aware knowledge graph embeddings for link prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3065–3072 (2020)
36. Zheng, S., Cao, W., Xu, W., Bian, J.: Doc2EDAG: an end-to-end document-level framework for Chinese financial event extraction. 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), Hong Kong, China, pp. 337–346. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1032. https://aclanthology.org/D19-1032