

EventKE: Event-Enhanced Knowledge Graph Embedding

Zixuan Zhang, Hongwei Wang, Han Zhao, Hanghang Tong, Heng Ji

Department of Computer Science

University of Illinois Urbana-Champaign

{zixuan11, hongweiw, hanzhao, htong, hengji}@illinois.edu

Abstract

Relations in most of the traditional knowledge graphs (KGs) only reflect static and factual connections, but fail to represent the dynamic activities and state changes about entities. In this paper, we emphasize the importance of incorporating *events* in KG representation learning, and propose an event-enhanced KG embedding model **EventKE**. Specifically, given the original KG, we first incorporate event nodes by building a heterogeneous network, where entity nodes and event nodes are distributed on the two sides of the network interconnected by event argument links. We then use entity-entity relations from the original KG and event-event temporal links to inner-connect entity and event nodes respectively. We design a novel and effective attention-based message passing method, which is conducted on entity-entity, event-entity, and event-event relations to fuse the event information into KG embeddings. Experimental results on real-world datasets demonstrate that events can greatly improve the quality of the KG embeddings on multiple downstream tasks.¹

1 Introduction

Knowledge graph (KG) is a kind of efficient and informative representation of structured knowledge. A typical KG consists of a collection of knowledge triples, where each triple (h, r, t) describes that the head entity h and tail entity t are connected through a relation r . Recently, extensive studies have been focusing on knowledge graph representation learning, which aims to learn low-dimensional entity and relation embeddings that are informative and scalable to use for many downstream applications, such as information retrieval (Yang, 2020), recommendation systems (Sun et al., 2020), machine reading comprehension (Qiu et al., 2019), and query-answering systems (Kacupaj et al., 2021;

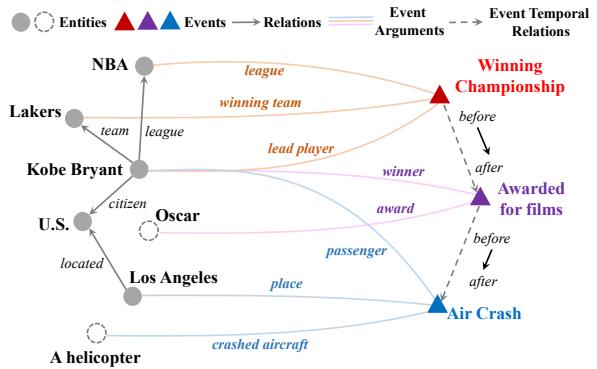


Figure 1: An example of representing an event-enhanced KG as a heterogeneous network.

Saxena et al., 2020). Typical KG embedding models, such as (Bordes et al., 2013; Dettmers et al., 2018; Sun et al., 2019), usually learn the model parameters by maximizing pre-defined score functions on ground-truth triples. One major limitation of such methods is that each knowledge triple is modeled locally and independently, without considering the global contextual information of KGs. To solve this problem, another line of approaches (Schlichtkrull et al., 2018; Vashisht et al., 2020) manages to model KGs as heterogeneous networks, and design message passing among entities using graph neural networks to better utilize global structural information.

However, the relations in knowledge graphs are mostly static factual connections between entities, which are still not sufficient for KG embedding models to learn fully rich and comprehensive entity representations. If we think of how humans understand a real-world entity, we usually consider not only its static facts and properties, but more importantly, the dynamic *events* associated with the entity as well. Based on this motivation, we hypothesize that *events* are essential for the model to understand entities and relations more comprehensively and improve the quality of KG embeddings, which can be beneficial for multiple knowledge-related downstream tasks.

¹Data and source codes are made publicly available at <https://github.com/zhangzx-uiuc/EventKE>.

In Information Extraction (IE), an *event* usually consists of an *event trigger* (a word or phrase that most directly indicates an event occurrence) and a set of *event arguments* (the entities participating in the event with different roles), such as the *Winning*, *Awarded*, and *Air Crash* events shown in Figure 1. In general, events can benefit KG modeling mainly in the following four aspects:

- Events contain much richer information compared to the entity-entity relations in KG. For example, in Figure 1, the *Awarded for films* event can give us a new aspect of the basketball player *Kobe Bryant* for his excellence in films, while the relations in regular KGs only contain his well-known facts on sports.
- Compared with the triples in KGs which can only involve two entities, events are able to capture relations among multiple entities (depending on the number of arguments of this event) and thus they have more expressive power than traditional knowledge triples. For example, in Figure 1, the event *Winning Championship* connects three entities (*NBA*, *Lakers*, and *Kobe Bryant*) together.
- Events can help improve the connectivity of KGs and mitigate the ubiquitous sparsity issue in most KGs. An event can reduce the distance between its two argument entities to 2 even if they are far away in original KGs. Events can also introduce more links to low-degree or isolated entities (e.g., *Oscar* and *A helicopter* in Figure 1) to enrich KGs.
- In general, events that are related to each other can be organized in chronological order. The event-event temporal relations can also be incorporated to inform the model with the dynamic state transitions of entities. For example, in Figure 1, the original KG only contains the outdated information when *Kobe Bryant* is still a basketball player. However, the event sequence *Winning* → *Awarded* → *AirCrash* can provide information of *Kobe*'s life at different stages, thus it will be greatly helpful for learning a more comprehensive entity representation.

In this paper, we propose EventKE, an event-enhanced knowledge graph embedding model to incorporate event information into KG representations. As shown in Figure 1, events are con-

sidered as an additional set of nodes beyond the original KG. Event nodes and entity nodes are distributed on two sides, and they are inter-connected by event argument links. Moreover, event nodes and entity nodes are inner-connected by KG relations and event-event temporal links, respectively. Inspired by graph attention networks (Velickovic et al., 2018), we design a four-stage attention-based information aggregation model to fuse event information into KG embeddings. Specifically, we first calculate event embeddings based on entity embeddings using a left-to-right graph attention mechanism on event argument links, and then update event embeddings based on event-event temporal links. After that, the entity representations are updated by first passing back the event information and then conducting information aggregation through entity-entity relations. The entire model is trained end-to-end by optimizing the convolutional-based distance function on ground-truth KG triples.

A practical challenge of applying event information to KG embedding is the scarcity of high-quality event annotations, since it is usually expensive to acquire manual event annotations that are relevant to KG entities. Fortunately, as the event extraction techniques have become mature recently, we can manage to use event extraction systems to extract our desired events from natural language texts with high quality. In this paper, in addition to the gold-standard event annotations such as those in the ACE-2005 dataset², we also use a state-of-the-art cross-document event extraction and tracking system (Wen et al., 2021a) to obtain events from news articles. This shows that our proposed EventKE can also be widely applicable to traditional KG datasets without manually-labeled events. We evaluate our trained KG embeddings on three typical tasks: *knowledge graph completion*, *entity classification* and *relation classification*, and the results demonstrate that the event information greatly improves knowledge graph representations.

Our contributions can be summarized as follows:

- We propose to incorporate event information into KG representation learning, and design a novel and effective attention-based bipartite information aggregation model to utilize the event information effectively.
- We demonstrate the effectiveness of EventKE by conducting experiments on multiple eval-

²<https://catalog.ldc.upenn.edu/LDC2006T06>

uation tasks. Experimental results show that incorporating events can significantly improve the learned KG embeddings compared with previous models. In specific, EventE achieves 13.4% average relative gain on KG completion, 3.5% absolute accuracy gain on entity classification, and 1.6% absolute accuracy gain on relation classification, respectively.

2 Problem Definition

A knowledge graph is denoted by $G = (V, E)$, where $v_i \in V$ denotes the i -th entity node and $(i, j) \in E$ denotes the relation between the i -th entity node and the j -th entity node. We use $r_{i,j} \in \mathcal{R}$ to denote the relation type for edge (i, j) , such as the relation types *team*, *league*, and *citizen* in Figure 1, where \mathcal{R} represents the pre-defined relation type category. Such a knowledge graph G can also be represented by a collection of decomposed knowledge triples $\mathcal{G} = \{(v_i, r_{i,j}, v_j)\}_{v_i, v_j \in V, r_{i,j} \in \mathcal{R}}$.

To model the events, we treat events as an additional set of event nodes with event temporal relations. We use \mathcal{E} to denote the set of event nodes, where each event $e_j \in \mathcal{E}$ is composed of an event trigger t_j , its event type $c_j \in \mathcal{C}$, and a set of event arguments A_j . The event trigger t_j is a word or phrase from the original sentence that most clearly indicates the event occurrence, e.g., “*wins*” for an event *Winning Championship*, which is already included when the events are annotated or extracted from the original texts. The event type c_j is from a pre-defined event type category \mathcal{C} , such as the *Winning Championship* and *Awarded for films* for the first two event nodes in Figure 1 respectively. The set of event arguments A_j is composed of a set of pairs of entities and event argument roles.

$$A_j = \{(v_{j,k}, z_{j,k})\}_{k=1}^{|A_j|},$$

where $|A_j|$ is the number of arguments for event e_j . $v_{j,k} \in V$ denotes the entity of the k -th argument for event e_j , and $z_{j,k} \in \mathcal{A}$ denotes the role type (also from a pre-defined collection of role types \mathcal{A}) for this entity in event e_j . For example, (“*Kobe Bryant*”, *winner*) and (“*Oscar*”, *award*) in Figure 1 are two event arguments for the event *Awarded*.

The objective of event-enhanced KG embedding is to learn a low-dimensional representation vector v_i for each node $v_i \in V$ with the help of the original knowledge graph G and the events \mathcal{E} .

3 Model

3.1 Overview

Given a knowledge graph $G = (V, E)$ and a set of events \mathcal{E} , we consider events as an additional set of nodes, and build a heterogeneous network where the entity nodes V and event nodes \mathcal{E} are distributed on two sides. As shown in Figure 2, the entity nodes and event nodes are inter-connected by event argument links, while both of the entity set and event set are also inner-connected by relations from the original KG and event-event temporal links, respectively. We design an L -layer event-aware bipartite information aggregation model to enforce the entities to aggregate information from both knowledge graph neighbors and relevant events, where the illustration for each layer is shown in Figure 2. The whole model is trained end-to-end by optimizing the convolution based scoring function on ground-truth knowledge triples with output embeddings from the last layer.

3.2 Event-aware Information Aggregation

As shown in Figure 2, given the heterogeneous network with entity and event nodes, each layer of the event-aware bipartite information aggregation model consists of four stages. First, we compute the event representations using a graph attention mechanism on the event argument links. Then, we conduct message passing on the event nodes with temporal relations to update the event representations. After that, we aggregate the event information back to the entities based on another graph attention mechanism on event argument links. Finally, the entity representations are updated by incorporating the event information and the neighborhood message from the original KG relations.

We use v_i^l to denote the representation vector for the i -th entity in layer l , and the relation type embedding is denoted as $r_{i,j}$. We use t_j , c_j to denote the embedding vectors for the event trigger and event type of event e_j respectively. For event arguments, since each of them is also an entity, we use $v_{j,k}^l$ to represent the embedding for the k -th argument of event e_j , and the corresponding role type embedding is denoted as $z_{j,k}$. Note that only the entity embeddings are updated in each layer, while other embeddings are identical in each layer and uniformly optimized through end-to-end training to reduce the model size.

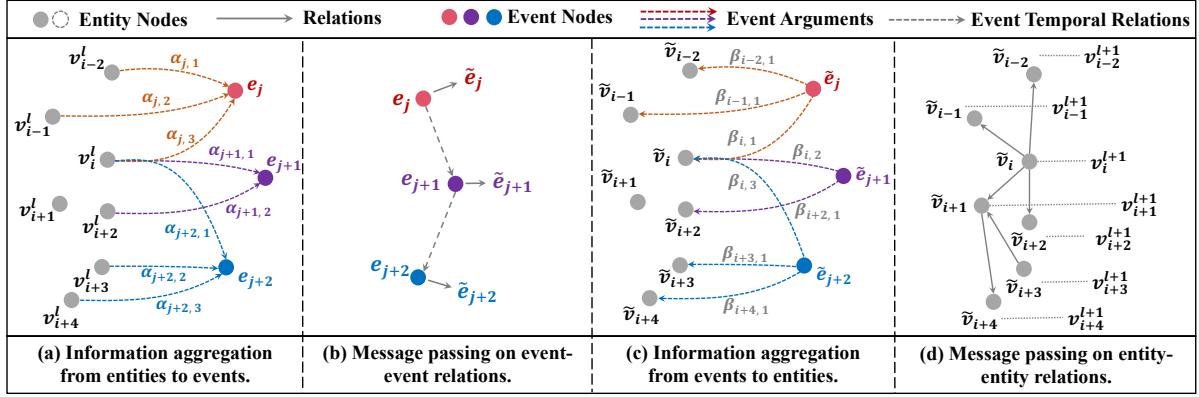


Figure 2: An illustration of the four-stage event-aware information aggregation model.

Information aggregation from entities to events. We first use an entity-to-event graph attention mechanism to aggregate entity information to events through the event argument links. For each event e_j with the number of arguments $|A_j|$, we first compute the attention weights $\alpha_{j,k}$ according to the concatenation of embeddings of event trigger, event type, entity, and role type:

$$\alpha_{j,k} = \frac{\exp \sigma \left(\mathbf{W}_\alpha \left[t_j, c_j, v_{j,k}^l, z_{j,k} \right] \right)}{\sum_{p=1}^{|A_j|} \exp \sigma \left(\mathbf{W}_\alpha \left[t_j, c_j, v_{j,p}^l, z_{j,p} \right] \right)}, \quad (1)$$

where $\sigma(\cdot)$ is a *LeakyReLU* activation function as in (Velickovic et al., 2018), and \mathbf{W}_α denotes a trainable parameter for linear transformation. Here, $v_{j,k}^l$ denotes the entity embedding for the k -th event argument for event j . Such an attention mechanism is capable of incorporating rich event information including event trigger, event type, and argument role types. Then the entity information λ_j passing to each event e_j from its arguments can be computed by the following weighted sum:

$$\lambda_j = \sum_{k=1}^{|A_j|} \alpha_{j,k} \sigma_e \left(\mathbf{W}_e v_{j,k}^l \right), \quad (2)$$

where $\sigma_e(\cdot)$ denotes the *ReLU* activation function and \mathbf{W}_e is a trainable matrix. We concatenate the event trigger embedding t_j , event type embeddings c_j , along with the entity information vector λ_j , as the event representation e_j :

$$e_j = [t_j, c_j, \lambda_j]. \quad (3)$$

Message passing on event-event relations. We then update the event representations e_j by conducting message passing on event-event temporal links. We use $\mathcal{N}(j)$ to denote the set of event nodes

that have temporal relations with e_j , then the event representation is updated similar to graph convolution network (Kipf and Welling, 2017).

$$\tilde{e}_j = e_j + \gamma \cdot \sigma_e \left(\frac{1}{|\mathcal{N}(j)|} \sum_{k \in \mathcal{N}(j)} e_k \mathbf{W}_t \right), \quad (4)$$

where γ is a hyper parameter controlling the amount of information to incorporate from event neighbors. Such a message passing procedure can efficiently inform the model with event-event temporal information.

Information aggregation from events to entities.

After we obtain the event representations \tilde{e}_j , we use another graph attention mechanism to aggregate event information back to entities. For each entity v_i , we use \mathcal{E}_i to denote the set of events that are connected with v_i through event arguments:

$$\mathcal{E}_i = \{e_j \mid \exists 1 \leq k \leq |A_j|, v_i = v_{j,k}\}. \quad (5)$$

For example, in Figure 2(c), the \mathcal{E}_i for node v_i should be $\{e_j, e_{j+1}, e_{j+2}\}$. We manage to fuse the event information back to involved entities using an attention mechanism, where the attention weight $\beta_{i,k}$ for event e_k is computed by

$$\beta_{i,k} = \frac{\exp \sigma \left(\mathbf{W}_\beta \left[\tilde{e}_k, v_i^l \right] \right)}{\sum_{p=1}^{|\mathcal{E}_i|} \exp \sigma \left(\mathbf{W}_\beta \left[\tilde{e}_p, v_i^l \right] \right)}, \quad (6)$$

where $\sigma(\cdot)$ and \mathbf{W}_β denote the *LeakyReLU* activation function and the linear transformation parameter respectively. Then we compute the event information \tilde{v}_i for entity node v_i by weighted sum:

$$\tilde{v}_i = v_i^l + \epsilon \cdot \sum_{k=1}^{|\mathcal{E}_i|} \beta_{i,k} \mathbf{W}_v \tilde{e}_k. \quad (7)$$

Similar to Eq. (4), we also adopt a hyper parameter ϵ to control the message passing level from events to entities. The event information vector $\tilde{\mathbf{v}}_i$ is supposed to contain rich information from events, including event triggers, event types, argument roles and also the event temporal relations.

Message passing on entity-entity relations. In the final stage, we conduct message passing on the original entity-entity relations from the original KG to further incorporate information from the original static relations between entities. To handle the different relation types, we adopt Composition GCN (Vashishth et al., 2020) to model the relation types using different relation type embeddings. We use $\mathcal{N}(i)$ to denote the set of entity nodes that are connected with v_i through the original entity-entity relations in knowledge graphs, and the entity representations are updated by

$$\mathbf{v}_i^{l+1} = \sigma_e \left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \mathbf{W}_r \phi(\tilde{\mathbf{v}}_j, \mathbf{r}_{i,j}) \right), \quad (8)$$

where $\sigma_e(\cdot)$ is the *ReLU* activation function, $\mathbf{r}_{i,j}$ is the relation type embedding, and $\phi(\cdot)$ denotes a circular correlation operator³ (Nickel et al., 2016) between two vectors. Finally, we obtain the updated entity representation \mathbf{v}_i^{l+1} that incorporates both the information from the original KGs and the rich event information from event nodes. We use the final layer output \mathbf{v}_i^L for model optimization and knowledge graph representation learning.

3.3 Optimization

After we obtain the entity representations \mathbf{v}_i^L and relation type embeddings $\mathbf{r}_{i,j}$, the model is optimized by maximizing the convolution-based score function for each knowledge triple $(v_i, r_{i,j}, v_j)$. Basically given each head entity v_i and the relation $r_{i,j}$, we use \mathcal{V}^+ to denote the set of all possible tail entities, and use \mathcal{V}^- to denote the set of random sampled negative entities. Then the loss for such a triple can be computed by binary cross-entropy:

$$\begin{aligned} \mathcal{L}(\mathbf{v}_i^L, r_{i,j}) = & - \sum_{v_+ \in \mathcal{V}^+} \log \sigma_s(\text{CONV}(\mathbf{v}_i^L, \mathbf{r}_{i,j}, \mathbf{v}_+^L)) \\ & - \sum_{v_- \in \mathcal{V}^-} \log (1 - \sigma_s(\text{CONV}(\mathbf{v}_i^L, \mathbf{r}_{i,j}, \mathbf{v}_-^L))) \end{aligned}$$

³The circular correlation $\phi(a, b)$ between two vectors a and b is defined as $[\phi(a, b)]_k = \sum_{i=0}^{d-1} a_i b_{(k+1) \bmod d}$, where d denotes the dimension of the vectors.

where σ_s is the sigmoid function, and $\text{CONV}(\cdot)$ is the convolution based scoring function defined in (Dettmers et al., 2018). Specifically, given a knowledge triple (s, r, t) , the score function can be computed by $\text{CONV}(s, r, t) = f(\text{vec}(f([\bar{s}, \bar{r}]) * \omega)) \mathbf{W} t$, where \bar{s} and \bar{r} denote the 2D reshaping for vectors and $*$ denotes the convolution operator.

4 Experiments

4.1 Dataset

We first evaluate our model on the Automatic Content Extraction (ACE 2005) dataset,⁴ which provides document-level entity, relation, and event annotations. We use this dataset because it has gold-standard event annotations, which is convenient to evaluate the impact of incorporating events into KG embeddings. We perform automatic entity linking (Wu et al., 2020) to link the entities to Wikidata (dumped in August 2019), and merge the entities with the same Wikidata entries into one single node. We use the most recent event-event temporal relation extraction system (Wen et al., 2021b) to obtain the event-event temporal relations.

However, gold-standard event annotations for real-world KGs are usually expensive or even intractable. To demonstrate that our proposed model can also be applied to KGs without gold-standard event annotations, we choose another large-scale dataset, WikiEvents (Li et al., 2021), which contains news articles from Wikipedia references about complex events. We use the most recent cross-document event extraction (Lin et al., 2020; Li et al., 2021), coreference resolution (Lai et al., 2021) and temporal tracking system (Wen et al., 2021a) to automatically extract events from the news articles.

The statistics of the extracted KGs from the two datasets are shown in Table 1.

Dataset	Entities	Rels	Events	Args
ACE-2005	7,376	7,441	3,071	5,587
WikiEvents	104,942	151,253	39,092	112,972

Table 1: The statistics of the extracted knowledge graphs from the two datasets, where *Rels* and *Args* denote the number of entity-entity relations and event argument links respectively.

⁴<https://catalog.ldc.upenn.edu/LDC2006T06>

4.2 Evaluation Tasks

We evaluate our trained knowledge graph embeddings on three typical evaluation tasks: *knowledge graph completion*, *entity classification*, and *relation classification*.

Knowledge Graph Completion KG completion (Bordes et al., 2013) is a typical task for evaluating knowledge graph embeddings, which aims to predict the tail entity t given a pair of head entity h and relation r . Since the number of entities in KG is usually large, we use the ranking based metrics (Bordes et al., 2013): mean ranking (MR), mean reciprocal ranking (MRR), and Hits@ k to evaluate the KG completion performance. For WikiEvents dataset with extremely large number of entities, it is not scalable to evaluate among all entities, therefore, we randomly selected a fixed number of negative samples and calculate the rankings of the correct tail entity.

Entity Classification The entity classification task aims to classify each entity node into a pre-defined entity type category. We divide all entities in the KG into 80% training, 10% validation, and 10% testing examples. After we obtain our trained KG embedding, we adopt a two-layer feed-forward neural network with *ReLU* activation on the hidden layer on the top of our model to fine-tune for entity classification task on the training entities, and select the best model on the validation set, then report the entity classification accuracy on the testing entities. There are totally 45 fine-grained entity types in *ACE-2005* dataset and 9 types in *WikiEvents*.

Relation Classification The relation classification task aims to identify the relation type between each given pair of entities. Similar to entity classification, we also divide all the entity-entity relations into training, validation, and testing sets and use a two-layer feed-forward neural network for relation classification, and use the accuracy as the evaluation metric. There are totally 18 relation types in *ACE-2005* dataset and 9 types in *WikiEvents*.

4.3 Baselines and Implementation Details

Baselines We first adopt the most typical previous models *TransE* (Bordes et al., 2013) and *CompGCN* (Vashisht et al., 2020) as the representatives of translation based and graph neural network based KG embedding models respectively. In both of these two methods, we only use the initial KG and the entity-entity relations without

any events, and the entity node embeddings are randomly initialized before training. To show the influence of incorporating the pretrained language model BERT (Devlin et al., 2019), in *CompGCN-BERT*, we use the *bert-base-uncased* model checkpoint⁵ to initialize entity node embeddings before conducting message passing using CompGCN. For our event-enhanced models, *EventKE* represents our proposed event-enhanced knowledge graph embedding model. In addition, to show the effects of each individual part of our model, we also introduce two model variants for ablation study. *EventKE-w/o-templink* denotes the model without information aggregation on event-event temporal links. In *EventKE-w/o-events*, instead of using annotated events or event extraction results, we randomly initialize all events (including the event triggers, event types, and event arguments) and the use such “random events” to train the model. Note that the number of model parameters in *EventKE-w/o-events* and *EventKE* are exactly the same, we want to evaluate whether the performance gains come from the valuable event information, or merely come from the larger capacity of the whole model.

Implementation Details We train the KG embedding models for a maximum of 200 epochs and apply an early stopping strategy of 10 epochs (if the validation loss is not lower than the previous best for 10 consecutive epochs, we will stop training the model). We train the models on NVIDIA Tesla V100 GPUs using the Adam (Kingma and Ba, 2015) optimizer with a learning rate of 10^{-4} . The average runtime is about 3 hours for the *ACE-2005* dataset and 5 days for the large KG in the *WikiEvents* dataset. The detailed hyper-parameter choices are shown in Appendix A.

4.4 Results

Table 2 and Table 3 show the performance on knowledge graph completion, entity classification, and relation classification, respectively. In general, our model significantly outperforms all baseline competitors. By incorporating event information (compared with *CompGCN-BERT*), our model can achieve 13.4% average relative gain on KG completion MRR, and 3.5% and 1.6% absolute gain on entity and relation classification accuracy.

Effects of incorporating events. The evaluation results demonstrate that incorporating event infor-

⁵<https://huggingface.co/bert-base-uncased>

Dataset		ACE-2005				WikiEvents			
Metrics		MRR	MR	Hits@10	Hits@20	MRR	MR	Hits@10	Hits@20
TransE (Bordes et al., 2013)		0.095	112	0.181	0.223	0.067	213	0.111	0.153
CompGCN (Vashisht et al., 2020)		0.102	88.8	0.201	0.256	0.074	186	0.134	0.171
CompGCN-BERT		0.178	79.5	0.322	0.419	0.149	168	0.210	0.259
EventKE (Ours)		0.203	67.5	0.378	0.480	0.168	147	0.257	0.322
- w/o temporal links		0.196	78.5	0.362	0.465	0.166	149	0.249	0.312
- w/o events		0.181	78.9	0.111	0.408	0.146	168	0.210	0.242

Table 2: Knowledge graph completion performance. We set the number of negative samples as $K = 500$.

Dataset	ACE-2005		WikiEvents	
Model	Ents	Rels	Ents	Rels
TransE	70.25	56.32	79.63	72.29
CompGCN	71.02	55.99	80.04	73.06
BERT-Only	71.13	58.64	82.11	76.01
CompGCN-BERT	73.92	60.81	82.56	76.58
EventKE (Ours)	77.47	62.60	85.61	77.94
- w/o temporal links	75.81	62.20	84.32	78.11
- w/o events	73.32	61.06	82.20	76.29

Table 3: Accuracy (%) of entity and relation typing.

mation can greatly improve the quality of entity and relation embeddings. From the results shown in Table 2 and Table 3, we can also find that event-event temporal links can also show effectiveness especially on the task of KG completion and entity classification. We can also see that the performance of *EventKE-w/o-event* is evidently lower than *EventKE* on all tasks, even if it has the same number of model parameters with *EventKE*. Such results demonstrate that the performance gains come from the event information instead of larger model capacity.

Effects of using BERT. From the results, we can also find that *CompGCN-BERT* performs much better than *CompGCN*, which demonstrates that initializing the entity and event node embeddings using pretrained language models can significantly improve the performance. This is probably because the BERT embeddings contain sentence-level contextual information of entity mentions and event triggers, which are not captured in pretraining in the random initialization setting.

4.5 Qualitative Analysis

To further inspect how events help better understand the knowledge graph, we analyze the output results from the validation set of KG completion task, and show two typical examples in Table 4. Given an input head entity and a relation, we show the rankings of the correct tail entity before and

after incorporating event information in the last column of Table 4.

In the first example, the model is required to predict where the *Republican Guard* is *located in*, however, the entity-entity relations (in the training set) from the knowledge graph do not contain such information, which makes the model output a low ranking for the correct entity *Baghdad*. After incorporating event information, the entities *Republican Guard* and *Baghdad* are connected through a *Conflict:Attack* event node, which indicates that *Republican Guard* is currently located in *Baghdad* to participate this attacking event. As a result, the number of hops between these two entities is greatly reduced, and the model can successfully predict the correct tail entity based on such event information.

Similarly, in the second example, the model is asked to predict the *membership* of *Al-Rantissi*. We can see that there is also a *Conflict:Attack* event indicating that *Isarel* targets to retaliate on *Hamas* and *Al-Rantissi*. Although such an attack event can not directly indicate that *Al-Rantissi* is a member of *Hamas*, it can inform the model that *Al-Rantissi* and *Hamas* are strongly related since they have both been targeted in one single attack event. Therefore, the ranking of *Hamas* is greatly increased from 117 to 3, which contributes to the overall performance gain of KG completion.

5 Related Work

Knowledge Graph Embedding The most traditional translation-based KG embedding models are first proposed by (Bordes et al., 2013), which manage to model each knowledge triple as translation and train the model by minimizing the distance function between head entity, relation, and tail entity. Previous translation-based models are mainly diverse in how to model the distance for each triple. Typical methods include using relation hyperplanes (Wang et al., 2014; Lin et al., 2015),

Sentence with events	Event Structure	Task Input	Ranking Change
Well, when they do finally enter Baghdad , U.S. and coalition troops could face urban combat with the Republican Guard , some of whom may have withdrawn into the city.		$\text{PHYS:Located} \rightarrow ?$ "Republican Guard"	<i>Rank: 1</i> <i>Rank: 45</i>
Israel retaliated on Hamas , namely Al-Rantissi , it missed him and killed civilians.		$\text{ORG-AFF Membership} \rightarrow ?$ "Al-Rantissi"	<i>Rank: 3</i> <i>Rank: 117</i>

Table 4: Qualitative examples from the validation set demonstrating how events can help on KG completion.

projection matrix decomposition (Ji et al., 2015, 2016), relational mappings (Fan et al., 2014), multiple relation projections (Do et al., 2018), Gaussian distributions (He et al., 2015; Xiao et al., 2016), neural networks (Yang et al., 2015; Trouillon et al., 2016), 2D convolutions (Dettmers et al., 2018; Jiang et al., 2019), and rotations (Sun et al., 2019). However, such translation-based methods model each knowledge triple individually without considering the global context for entities in the KG. To solve this problem, another line of work attempts to model the KG as a unified heterogeneous graph and use graph neural networks to capture the global interactions between entity nodes, such as (Nickel et al., 2016; Schlichtkrull et al., 2018; Zhang et al., 2019; Vashishth et al., 2020; Hu et al., 2020; Shui and Karypis, 2020). Recent studies also focus on using additional information to improve KG embedding, such as entity type information (Niu et al., 2020) and text information (Wang et al., 2021).

Event Extraction and Utilization Events can help the KG embedding model to get a more comprehensive understanding of entities. A variety of event extraction models are proposed to extract events from natural language texts, such as (Huang et al., 2016, 2018; Luan et al., 2019; Wadden et al., 2019; Lin et al., 2020; Li et al., 2021). The dynamic event information is also well utilized in various areas, such as pretrained language models (Yang et al., 2019), event temporal information prediction (Yang et al., 2020; Wen et al., 2021b), event schema induction (Li et al., 2020), and event network representation learning (Zeng et al., 2021). To the best of our knowledge, we are the first to focus on improving the entity and relation embeddings using event information in KG representation learning.

Comparison and Discussion Our approach is highly related to the global event network embedding model (Zeng et al., 2021), however, there are a few essential differences. First, (Zeng et al., 2021) focuses on learning the embedding vectors for events to use for event-centric downstream tasks such as event type classification, argument extraction, and event coreference resolution. Our model focus on using event information to improve entity and relation understanding in knowledge graphs. For model design, (Zeng et al., 2021) treats the entity and event nodes as a unified network, and adopts a relational GCN module to encode the entire network. On the contrary, we regard the entities and events as two separate sets of nodes and conduct pairwise message passing between the two node sets. In addition, compared to (Zeng et al., 2021), we manage to make use of a wider range of event information including the event types and event temporal relations.

6 Conclusion and Future Work

Relations in traditional KGs usually reflect only static and factual connections, which fail to incorporate dynamic activities and state changes about entities. In this paper, we focus on incorporating *events* into KG representation learning, and building a heterogeneous network by introducing an additional event node layer. We design a novel and effective attention-based message passing method on entity-entity, event-entity, and event-event relations, to fuse the event information into KG embeddings. Experimental results on multiple evaluation tasks show the compelling effectiveness of incorporating events in KGs. For future work, we will consider applying EventKE to more challenging event-related tasks, such as event schema induction. In addition, using pretrained language models

to jointly learn KG and text embeddings is also a promising direction.

Acknowledgement

This research is based upon work supported by U.S. DARPA KAIROS Program No. FA8750-19-2-1004, U.S. DARPA AIDA Program No. FA8750-18-2-0014, Air Force No. FA8650-17-C-7715, LORELEI Program No. HR0011-15-C-0115, the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract No. FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. [Translating embeddings for modeling multi-relational data](#). In *Advances in Neural Information Processing Systems NIPS 2013*, pages 2787–2795.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. [Convolutional 2d knowledge graph embeddings](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 1811–1818.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186.

Kien Do, Truyen Tran, and Svetha Venkatesh. 2018. [Knowledge graph embedding with multiple relation projections](#). In *24th International Conference on Pattern Recognition, ICPR 2018*, pages 332–337.

Miao Fan, Qiang Zhou, Emily Chang, and Thomas Fang Zheng. 2014. [Transition-based knowledge graph embedding with relational mapping properties](#). In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, PACLIC*, pages 328–337.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. [Learning to represent knowledge graphs with gaussian embedding](#). In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015*, pages 623–632.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. [Heterogeneous graph transformer](#). In *The Web Conference 2020 WWW*, pages 2704–2710.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. [Liberal event extraction and event schema induction](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. [Zero-shot transfer learning for event extraction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. [Knowledge graph embedding via dynamic mapping matrix](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 687–696. The Association for Computer Linguistics.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. [Knowledge graph completion with adaptive sparse transfer matrix](#). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI 2016*, pages 985–991.

Xiaotian Jiang, Quan Wang, and Bin Wang. 2019. [Adaptive convolution for multi-relational learning](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 978–987, Minneapolis, Minnesota. Association for Computational Linguistics.

Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. 2021. [Conversational question answering over knowledge graphs with transformer and graph attention networks](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics EACL*, pages 850–862.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas N. Kipf and Max Welling. 2017. [Semi-supervised classification with graph convolutional networks](#). In *5th International Conference on Learning Representations, ICLR 2017*.

Tuan Lai, Heng Ji, Trung Bui, Quan Hung Tran, Franck Dernoncourt, and Walter Chang. 2021. A context-dependent gated module for incorporating symbolic semantics into event coreference resolution. In *Proc. The 2021 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT2021)*.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. 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)*, pages 684–695, Online. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *The North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2021)*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence AAAI 2015*, pages 2181–2187.

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

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

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence AAAI 2016*, pages 1955–1961.

Guanglin Niu, Bo Li, Yongfei Zhang, Shiliang Pu, and Jingyang Li. 2020. Autoeter: Automated entity type representation with relation-aware attention for knowledge graph embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020*, pages 1172–1181. Association for Computational Linguistics.

Delai Qiu, Yuanzhe Zhang, Xinwei Feng, Xiangwen Liao, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. 2019. Machine reading comprehension using structural knowledge graph-aware network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5896–5901. Association for Computational Linguistics.

Apoory Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics ACL*, pages 4498–4507.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Extended Semantic Web Conference ESWC 2018*, pages 593–607.

Zeren Shui and George Karypis. 2020. Heterogeneous molecular graph neural networks for predicting molecule properties. In *20th IEEE International Conference on Data Mining, ICDM 2020*, pages 492–500.

Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. 2020. Multi-modal knowledge graphs for recommender systems. In *International Conference on Information and Knowledge Management, CIKM 2020*, pages 1405–1414.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations ICLR 2019*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, pages 2071–2080.

Shikhar Vashisht, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations ICLR 2020*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. **KEPLER: A unified model for knowledge embedding and pre-trained language representation**. *Transactions on Association of Computational Linguistics (TACL)*, 9:176–194.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. **Knowledge graph embedding by translating on hyperplanes**. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence AAAI 2014*, pages 1112–1119.

Haoyang Wen, Ying Lin, Tuan M. Lai, Xiaoman Pan, Sha Li, Xudong Lin, Ben Zhou, Manling Li, Haoyu Wang, Hongming Zhang, Xiaodong Yu, Alexander Dong, Zhenhailong Wang, Yi R. Fung, Piyush Mishra, Qing Lyu, Dídac Surís, Brian Chen, Susan W. Brown, Martha Palmer, Chris Callison-Burch, Carl Vondrick, Jiawei Han, Dan Roth, Shih-Fu Chang, and Heng Ji. 2021a. **Resin: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system**. In *The North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2021 Demo Track)*.

Haoyang Wen, Yanru Qu, Heng Ji, Qiang Ning, Jiawei Han, Avi Sil, Hanghang Tong, and Dan Roth. 2021b. **Event time extraction and propagation via graph attention networks**. In *The North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2021)*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. **Scalable zero-shot entity linking with dense entity retrieval**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. **TransG : A generative model for knowledge graph embedding**. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2316–2325, Berlin, Germany. Association for Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. **Embedding entities and relations for learning and inference in knowledge bases**. In *3rd International Conference on Learning Representations ICLR 2015*.

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

Zonglin Yang, Xinya Du, Alexander Rush, and Claire Cardie. 2020. **Improving event duration prediction via time-aware pre-training**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3370–3378, Online. Association for Computational Linguistics.

Zuoxi Yang. 2020. **Biomedical information retrieval incorporating knowledge graph for explainable precision medicine**. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020*, page 2486.

Qi Zeng, Manling Li, Tuan Lai, Heng Ji, Mohit Bansal, and Hanghang Tong. 2021. **Gene: Global event network embedding**. In *In the TextGraphs Workshop at NAACL 2021*.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. **Heterogeneous graph neural network**. In *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019*, pages 793–803.

A Hyper Parameter Settings

Hyper-parameters	Values
Entity embedding dimensions	768
Relation embedding dimensions	768
Event embedding dimensions	768
Event type embedding dimensions	200
Role type embedding dimensions	200
Event-event passing level γ	0.5
Entity-event passing level ϵ	0.8
Num of layers for FFNNs	2
FFNN hidden dimensions for entity classification	200
FFNN hidden dimensions for relation classification	200
FFNN dropout rate	0.3
GCN dropout rate	0.4
Hidden activation function	<i>ReLU</i>
Learning rate	1e-4
Pretraining batch size	128
Batch size for entity typing	2,560
Batch size for relation typing	2,560

Table 5: Detailed settings for model hyper-parameters. Note that the message passing level γ and ϵ are determined by grid search in the range $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$.