A Hybrid Graph Model for Distant Supervision Relation Extraction

Shangfu Duan, Huan Gao, Bing Liu, and Guilin Qi(✉)

School of Computer Science and Engineering, Southeast University, Nanjing, China
{sf_duan,gh,liubing_cs,gqi}@seu.edu.cn

Abstract. Distant supervision has advantages of generating training data automatically for relation extraction by aligning triples in Knowledge Graphs with large-scale corpora. Some recent methods attempt to incorporate extra information to enhance the performance of relation extraction. However, there still exist two major limitations. Firstly, these methods are tailored for a specific type of information which is not enough to cover most of the cases. Secondly, the introduced extra information may contain noise. To address these issues, we propose a novel hybrid graph model, which can incorporate heterogeneous background information in a unified framework, such as entity types and human-constructed triples. These various kinds of knowledge can be integrated efficiently even with several missing cases. In addition, we further employ an attention mechanism to identify the most confident information which can alleviate the side effect of noise. Experimental results demonstrate that our model outperforms the state-of-the-art methods significantly in various evaluation metrics.

Keywords: Distant supervision · Relation extraction · Heterogeneous information · Hybrid graph

1 Introduction

Relation Extraction (RE) aims at extracting semantic relations from plain text. It plays a crucial role in a wide range of applications such as Knowledge Graph (KG) completion and question answering. For example, given the sentence #1 in Fig. 1, RE aims to identify the *FounderOf* relation between *Bill Gates* and *Bill & Melinda Gates Foundation*. Supervised approaches have achieved excellent performance in RE. However, they suffer from the lack of labeled data while manual annotation is very costly. Distant supervision (DS) [6,13,16] is a promising approach to solve this limitation. It can generate labeled data automatically by aligning KGs with text. As shown in Fig. 1, if the triplet *<Bill Gates, FounderOf, Bill & Melinda Gates Foundation>* exists in KG, then all the sentences containing *Bill Gates* and *Bill & Melinda Gates Foundation* are taken as the training instances of *FounderOf* relation.

P. Hitzler et al. (Eds.): ESWC 2019, LNCS 11503, pp. 36–51, 2019.
https://doi.org/10.1007/978-3-030-21348-0_3

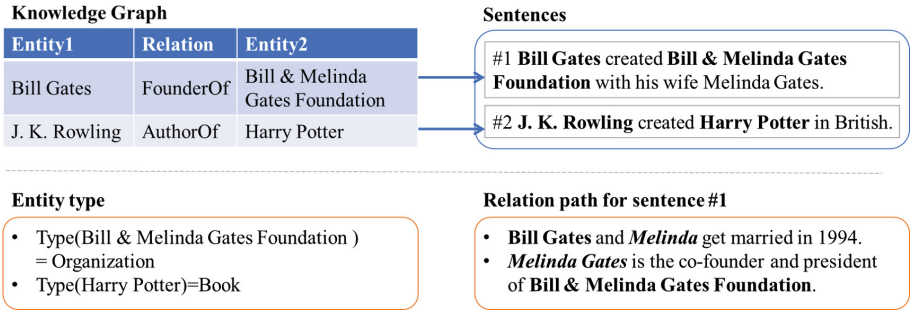


Fig. 1. Examples of DS.

Recently, various models based on Deep Neural Network (DNN) [14,23–25] have been proposed to improve Distant Supervision Relation Extraction (DSRE). These methods mainly focused on dealing with the noise problem brought by DS and have achieved a huge improvement. However, they still lacked sufficient background information for making predictions. For example, as shown in Fig. 1, there are two DS generated sentences and they describe different relations. It's hard to predict their relations exactly because they both use "*create*" to express corresponding relations. Recent studies have attempted to introduce more background information to enhance DSRE. Some methods proposed in [7,11] introduced the types and descriptions of entities which provide rich entity information to RE. Other researchers [5,22] carried out the representation learning of KG and the training of DS models jointly so as to introduce the knowledge in KG to the DS model. Zeng et al. [25] incorporated inference information that is hidden in extra text and proposed a path-based model to enhance DSER. These studies have shown the effectiveness of introducing background information. However, there are still two major problems remaining to be addressed.

On the one hand, the introduced background information is too sparse, especially for the long-tail relations. Previous methods only considered a single type of background information and it's not enough to cover the most cases. In addition, these methods always designed customized models to combine the corresponding knowledge which is limited to incorporate heterogeneous background information simultaneously. On the other hand, the previous studies did not consider alleviating the side effect of the introduced noise. For example, the methods [11] obtained the entity types with NLP tools and this inevitably brought some errors and hurt the RE performance.

To solve the above problems, we propose a novel graph-based model for DSRE, which can not only incorporate heterogeneous background information but also alleviate the side effect of introduced noisy information. We first transform different types of information into vectors with various encoders and treat each piece of information as a node in the graph. Then, we connect the related nodes and fuse all information with a Graph Convolutional Network (GCN). Notice that, our model is highly robust to the missing information and flexible

to integrate various types of information. To alleviate the effect of the introduced noisy information, we further employ an attention mechanism over the graph, which can assign a higher weight to more confident information.

The contributions of our study can be summarized as follows:

- We propose a novel graph-based DS model, which can incorporate heterogeneous background information in a unified framework and is flexible to integrate various kinds of knowledge.
- We employ an attention mechanism over the graph which can alleviate the effect of the introduced noise.
- We conduct extensive experiments on a real-world DS dataset and the results demonstrate that our model outperforms state-of-the-art methods significantly via various evaluation metrics.

2 Background

In this section, we give a brief introduction of related background information and the Graph Convolutional Network.

2.1 Various Types of Background Information

Sentence Bag. We denote a DS generated data set as $\mathcal{D} = \{S_{(h_i,t_i)}|, i = 1, 2, \ldots\}$ and a sentence bag $S_{(h_i,t_i)}$ is a collection of sentences that each sentence mentions both entities h and t.

Entity Representation in KG. A Knowledge Graph (KG) is a collection of triples (h_i, r_i, t_i) and the representation learning of KG aims to learn a vector embedding of both entities and relations into a low-dimensional space. Translation-based model [2] was proposed by treating the labeled relation embedding $\mathbf{r_i}$ as a translation of the embeddings of $\mathbf{h_i}$ and $\mathbf{t_i}$, i.e. $\mathbf{h_i} + \mathbf{r_i} \approx \mathbf{t_i}$.

Entity Type. For any entity e, we can get its type y_{e_i} from a KG. Fine-grained entity types [13] provide powerful constraints between an entity pair and a relation. For example, as shown in Fig. 1, the type of *Harry Potter* is *book*. With this background knowledge, it's much easy to predict the relation between *J.K.Rowling* and *Harry Potter* from *FounderOf* and *AuthorOf*.

Relation Path. A relation path describes the flow of resource between multiple entities [4,12,15,20,21]. In DSRE, we define a relation path as a set of entities, $p = \{h, e_1, ...e_l, t\}$, which means the resource flows from h to t through l entities. More specifically, the path is represented as $h \xrightarrow{S_{(h,e_1)}} e_1 \xrightarrow{S_{(e_1,e_2)}} \ldots \xrightarrow{S_{(e_l,t)}} t$.

2.2 Graph Convolutional Network

Graph Convolutional Networks (GCN) [8,17] is an efficient variant of Convolutional Neural Networks (CNNs) and aims at dealing with the graph structure data. GCN can extract local graph structure and learn a better representation of each node. Given a graph with two inputs: A feature matrix $X \in \mathbb{R}^{N \times D}$ where N is the number of nodes and D denotes the size of node features and an adjacency matrix $A \in \mathbb{R}^{N \times N}$, a typical GCN outputs a node-level representation $Z \in \mathbb{R}^{N \times F}$ where F is the size of output features. Each neural network layer has the following form

$$H^{(l+1)} = f(H^{(l)}, A), \tag{1}$$

where $H^{(0)} = X$ and $H^{(L)} = Z$, L is the number of layers.

In each layer, the layer-wise propagation are shown in Eq. 2.

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{l}) \tag{2}$$

To further solve the limitations of Eq. 2, Kipf et al. [8] uses a symmetric normalization and enforce a self-loops in the graph. In this paper, we will represent GCN as a transform function of X and A, as shown in Eq. 3, where $\hat{A} = A + I$, I is the identity matrix and $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$.

$$GCN(X, A) = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}XW\right) \tag{3}$$

3 Methodology

In this section, we present our hybrid graph model, which can fuse heterogeneous background information and alleviate the side effect of introduced noise.

3.1 Problem Definition

Given a DS generated dataset $D = \{S_{(h_i,t_i)}|, i = 1, 2, \ldots\}$, for each entity pair (h_i, t_i), we extend some background information from a KG, and denote the dataset as $\mathbb{I} = \{I_{(h_1,t_1)}, I_{(h_2,t_2)}, \ldots\}$. The label of each instance $I_{(h_i,t_i)}$ keeps same as the label of $S_{(h_i,t_i)}$. In this paper, our methods aim to predict the relation between an entity pair, and we finally learn a probability distribution $P(r_i|h_i, t_i; \theta)$ over all relations $r_i \in \mathcal{R}$, where θ denotes the parameters in our model.

3.2 Overview

The overview of our approach is shown in Fig. 2. When carrying out the RE task, our model not only considers the sentence bag generated by DS, but also introduces various types of background information, such as the entity type, the entity representation in KG, the entity graph context and the relation path. The whole process includes three stages. Firstly, these types of information are

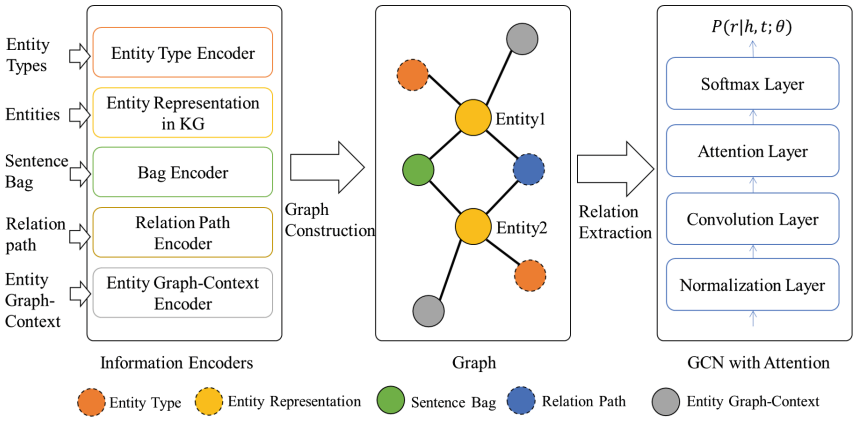


Fig. 2. Overview of our method.

converted into vector representations respectively using corresponding encoders. Then, we construct a hybrid graph by treating each piece of information as a node and connecting related information.

We utilize a graph convolutional methods to learn a more discriminative representation of each piece of information. Finally, an attention-based GCN model extracts features of the graph and outputs a probability distribution of relations. Notice that our model can fuse all types of information on the graph, even though some nodes may be missing.

3.3 Encoders

In this part, we will introduce different encoders that used for heterogeneous background information respectively.

Sentence Bag Encoder. For any sentence $s_i \in S_{(h,t)}$ with an entity pair (h,t), we apply a Bi-LSTM model [10] to obtain its vector representation $\mathbf{s}_i \in \mathbb{R}^{d_s}$ where d_s denotes the size of sentence embedding. It's also flexible to replace with other models as long as it can represent the semantics of the given sentence. Afterward, we get the sentence bag representation $\mathbf{S}_{(h,t)} \in \mathbb{R}^{d_s}$ by summing all sentence embeddings with different weight. There are plenty of methods to calculate the weigh and we follow the methods used in [25].

Entity Type Encoder. Given a entity e and its type $y_{e_i} \in \mathcal{T}$, where $\mathcal{T}$ denotes the type set of all entities. Here, we take the one-hot encoding as inputs. During the training process, we will learn a distribution representation of each entity type and update the representation matrix $\mathbf{M}_y \in \mathbb{R}^{|\mathcal{T}| \times d_t}$ dynamically, where d_t denotes the size of entity type embedding.

Entity Encoder. Each entity mention e_i will be mapped into a real-value vector $\mathbf{e}_i$ with dimension d_e. In which we use a pre-trained PTransE model [9] to get the embedding of all the entities. The reason why we adopt PTransE is that PTransE can capture the path information among the KG. In addition, most existing translation-based methods can be easily integrated into the framework.

Relation Path Encoder. We propose a relation path encoder to jointly inference among multiple paths with a selective attention. To represent the flow of information on a path, the entity embedding information also has been integrated into our relation path encoder.

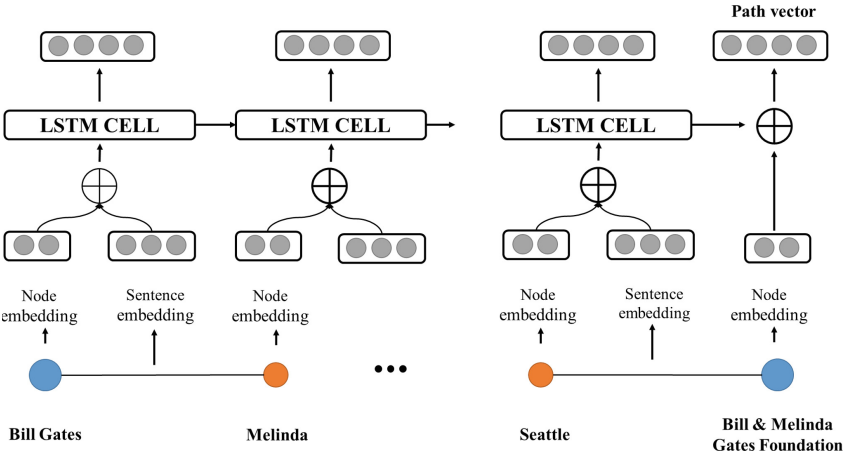


Fig. 3. The architecture of our path encoder.

As shown in Fig. 3, given an entity path $p = \{h, e_1, ..., e_l, t\}$ between an entity pair (h, t), we use the LSTM model to predict the relation by capturing the flows from h to t. For any LSTM cell at timestep k, we firstly concatenate corresponding entity embedding and sentence bag embedding, as shown in Eq. 4, where $W_e \in \mathbb{R}^{d_p \times (d_e + d_s)}$, $b_e \in \mathbb{R}^{d_p}$ are model parameters and the $\oplus$ means the concatenation operation. Here, d_p denotes the size of path embedding.

$$\mathbf{x}_k = W_e(\mathbf{e}_k \oplus \mathbf{S}_{(e_k, e_{k+1})}) + b_e, \tag{4}$$

Afterward, each process step is defined in Eq. 5, where $LSTM(x, h, c)$ denotes a standard LSTM cell [14], and at each timestep $k+1$, the hidden state h_{k+1} is a function of the current input embedding x_{k+1} with the last step's hidden status h_k and cell state C_k. We use the last hidden stages to represent the relation path vector $\mathbf{p} \in \mathbb{R}^{d_p}$.

$$h_{k+1} = LSTM(x_{k+1}, h_k, C_k) \tag{5}$$

Following the MIL framework [6,10], we use a selective attention over each path and get the embedding $\mathbf{P}_{(h,t)} \in \mathbb{R}^{d_p}$ which combines all information of relation paths.

Entity Graph-Context Encoder. For any entity e, we collect all entities that occur with it in a sentence as its graph-context N_e and encode the graph-context into a vector representation $\mathbf{N}_e \in \mathbb{R}^{d_e}$, as shown in Eq. 6.

$$\mathbf{N}_e = \sigma\left(\frac{1}{|N_e|}\sum_{e' \in N_e} \mathbf{e}'\right) \tag{6}$$

3.4 Graph Convolutional Network with Attention

One crucial challenge for fusing heterogeneous information is that they have different embeddings. Hence, we propose a hybrid graph model and use the adjacent matrix to represent the correlation between each piece of information. For a given instance $I_{(h,t)}$ and each embeddings of various background information, we first transform the varying structure into a fixed structure with an adjacent matrix. Then, we apply a GCN to extract high-level features and get a new representation of each node. Finally, we utilize an attention layer to combine heterogeneous information together with different weights. Specially, we will calculate the similarity with a dot product operation between the embedding of each node and an approximation representation of relation r.

Normalization Layer. We propose an adaptive framework to transform various embeddings into a graph structure. More specifically, we represent our hybrid graph with a node matrix $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_{d_g}\}$ and an adjacency matrix $\mathbf{A} \in \mathbb{R}^{d_g \times d_g}$. Here, each element in $\mathbf{X}$ denotes a kind of information and $\mathbf{A}_{i,j} = 1$ indicates the i_{th} and j_{th} information are correlated. In this paper, for an entity pair (h,t), we build the hybrid graph $\mathbf{X} = \{\mathbf{S}_{(h,t)}, \mathbf{P}_{(h,t)}, \mathbf{h}, \mathbf{y}_h, \mathbf{t}, \mathbf{y}_t, \mathbf{N}_h, \mathbf{N}_t\}$, where $\mathbf{h}$ and $\mathbf{t}$ are embeddings of the given entity pair, $\mathbf{y}_h$ and $\mathbf{y}_t$ are corresponding entity type embeddings, $\mathbf{N}_h$ and $\mathbf{N}_t$ are entity context embeddings. $\mathbf{S}_{(h,t)}$ and $\mathbf{P}_{(h,t)}$ are the sentence bag embedding and relation path embedding which connect both $\mathbf{h}$ and $\mathbf{t}$, $\mathbf{N}_h$ and $\mathbf{y}_h$ connect $\mathbf{h}$, $\mathbf{N}_t$ and $\mathbf{y}_t$ connect $\mathbf{t}$. In order to keep simplicity and convenient, we set the size of all embeddings as $d_i = d_s = d_p = d_e = d_t$, thus the node matrix can be denoted as $\mathbf{X} \in \mathbb{R}^{d_g \times d_i}$.

Convolutional Layer. We propose a variant of graph convolutional network [8, 17] which takes the inputs of the node matrix $\mathbf{X}$ and the corresponding adjacency matrix $\mathbf{A}$. The output is a matrix represents new features of each node. It's efficiently for our convolutional layer to capture the structure pattern with shared parameters. Specifically, we do not need to train large parameters over the whole graph and it's space-efficient and time-efficient. As shown in Eq. 7, the output

matrix $\mathbf{G} \in \mathbb{R}^{d_g \times d_o}$ represents all background information, where d_o is the size of output features.

$$\mathbf{G} = GCN(\mathbf{X}, \mathbf{A}) \tag{7}$$

Attention Layer. It's straightforward that some extra knowledge may express irrelevant or noisy information on the given entity pair. Thus, we propose a graph attention layer to learn a discriminative representation among all background knowledge embeddings. Inspired by translation-based model, we use an approximation representation of the relation between entity pair (h, t), as shown in Eq. 8. Then, we calculate the similarity between each background knowledge embedding $\mathbf{G} = \{\mathbf{g}_1, ..., \mathbf{g}_{d_g}\}$ with the approximation relation representation $\mathbf{r}_{h,t}$, shown in Eq. 10.

Afterward, we apply a weighted sum operation overall features on the graph and assign a high weight to more relevant feature to alleviate the effect of noise, as shown Eq. 11:

$$\mathbf{r}_{h,t} = \mathbf{t} - \mathbf{h} \tag{8}$$

$$\mathbf{u}_j = tanh(\mathbf{W}_a \mathbf{g}_j + \mathbf{b}_a), \tag{9}$$

$$a_j = \frac{exp(\mathbf{r}_{h,t} \cdot \mathbf{u}_j)}{\sum_{k=1}^{d_g} exp(\mathbf{r}_{h,t} \cdot \mathbf{u}_k)}, \tag{10}$$

$$\mathbf{r}_g = \sum_{j=1}^{d_g} a_j \mathbf{g}_j, \tag{11}$$

where $\mathbf{W}_a \in \mathbb{R}^{d_i \times d_o}$ and $\mathbf{b}_a \in \mathbb{R}^{d_i}$ are the model parameters, a_j denotes the weight of each background feature.

Finally, we calculate the probability of each relation r in Eq. 12,

$$\mathbf{o} = \mathbf{M}\mathbf{r}_g, \tag{12}$$

where $\mathbf{M}$ is the score matrix to calculate the scores of each relation $r' \in \mathcal{R}$ and $\mathbf{o} \in \mathbb{R}^{d_r}$ denotes the probabilities of all relations, i.e. $\mathbf{o} = (o_{r_1}, o_{r_2}, ...)$. For a given entity pair (h, t) and the corresponding instance $I_{(h_1,t_1)}$, we conduct the conditional probability $P(r|h, t, I_{(h_1,t_1)})$ as shown in Eq. 13.

$$P(r|h, t, I_{(h_1,t_1)}) = \frac{\exp(o_r)}{\sum_{r' \in \mathcal{R}} \exp(o_{r'})} \tag{13}$$

3.5 Optimization

For an entity pair (h, r), the overall objective function is defined in Eq. 14, where θ denotes the parameters in our model.

$$\ell_\theta(r|h, t; \theta) = \log(P(r|h, t, I_{(h_i,t_i)})) \tag{14}$$

We use mini-batch stochastic gradient descent (SGD) to maximize our objective function.

$$\min_{\theta} \mathbb{E}\left[\sum_{(h_i,t_i)\in\mathbb{I}_{batch}} \frac{\ell_\theta\left(r_i|h_i,t_i,\theta\right)}{|\mathbb{I}_{batch}|}\right] \tag{15}$$

4 Experiments

Our experiments aim to measure (1) whether the introduction to additional background information can improve the prediction performance in DSRE, and (2) whether our model can reduce the effect of the introduced noisy data. Our source code and dataset are available at GitHub[1].

4.1 Dataset and Metrics

The most commonly used benchmark dataset for DSRE was proposed in [16], and this dataset was generated by aligning Freebase [1] relations with the New York Times (NYT) corpus. Due to the sparse and scattered connection of different entities, this dataset does not have enough relation paths to measure the effect of this type of information. Therefore, we apply an up-to-date dataset proposed in [25] which addresses the above issue by aligning Wikidata[2] relations with the NYT corpus. Compared to the Freebase dataset, this dataset contains more instances and the detailed statistics are shown in Table 1.

Table 1. Statistics of the benchmark dataset.

Set	#Sentences	#Entity pairs	#Facts
Train	647,827	266,118	50,031
Valid	234,350	121,160	5,609
Test	235,609	121,837	5,756

Following the previous work [13], we measure our model based on the held-out metric. This metric provides an approximation measurement of precision automatically by comparing the predicted relations with corresponding facts in Wikidata. We compare our model with the baseline methods and other approaches which employing part of background information. In addition, we also evaluate the robustness and capacity of different percentage of noisy data. To further demonstrate the effect of our hybrid graph model, we also report the Precision@N results and F1 score with different noise ratio.

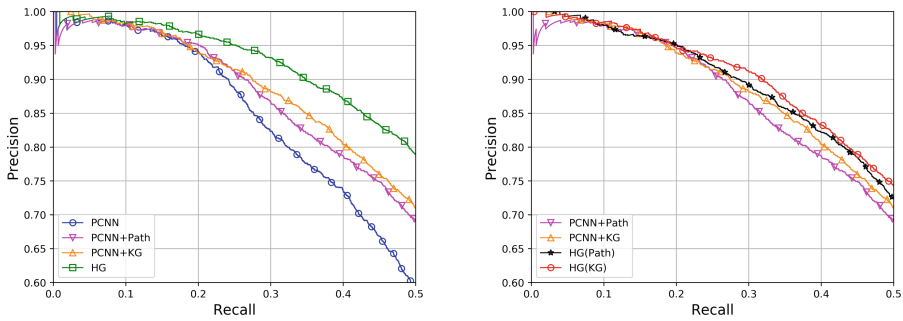
[1] https://github.com/Apeoud/HG-DSRE.git.
[2] www.wikidata.org/.

4.2 Experimental Settings

We apply different combination of parameters in validation dataset and the optimal parameters are shown as follows. We use a pre-trained word embeddings on NYT corpus with different embedding size $d_w = \{50, \mathbf{100}, 200, 300\}$. For entity embedding, we train a PTransE model [9] with the dimension d_e among $\{\mathbf{50}, 100, 200, 300\}$. We select the learning rate λ for SGD among $\{\mathbf{0.01}, 0.1, 1.2\}$. For training, the best mini-batch size B is $\{30, \mathbf{50}, 100, 300\}$. We also apply dropout on the last layer to avoid overfitting with dropout rate 0.5. Other parameters have little effect and we follow the settings used in [25].

4.3 Precision-Recall Curve Comparison

To demonstrate the performance of our model, we compare it with other methods via the held-out evaluation. Figure 4(a) demonstrates the overall evaluation results which include our method with the comparison to three baseline approaches. (1) PCNN represents the Piecewise-CNN model with multi-instances learning used in [24]. (2) PCNN+path [25] incorporated relation path with PCNN model. (3) PCNN+KG [5] jointly learned an embedding of word in text and entity mention in KG. (4) HG is our method which introducing four additional information, the entity embeddings in KG, the entity type, the entity graph-context and the relation paths. Figure 4(b) introduce two variant of our method, HG(Path) and HG(KG). HG(Path) takes a limitation of the additional knowledge and only consider relation path information while HG(KG) only incorporate KG embeddings. From the aggregate precision/recall curves, we observe that:



(a) Overall results. (b) Effectiveness of the graph model.

Fig. 4. Aggregate PR curves upon the held-out evaluation. (a) shows the comparison between our methods and other baselines. (b) demonstrates the noise immunity when incorporating the same extra knowledge for our methods and others.

Overall Results. (1) When the recall is very small (0–0.1), all methods keep nearly equivalent performance. As the recall grows, the precision of PCNN drops rapidly. This observation indicates that only a small part of the instance contains reasonable information to make sufficient prediction and it's necessary to introduce external information to avoid this problem. (2) We also observe that the methods which utilize background information have a slow decay with the recall increase, this phenomenon indicates that incorporating background information can really improve the prediction accuracy. (3) Our methods (HG) outperform all methods and achieve the best performance with 20% improvement in precision to PCNN when recall equals 0.5 and 10% improvement to PCNN+Path. It indicates that our methods can take advantages of multiple background information.

Effectiveness of Our Hybrid Graph. While previous experiments demonstrate the effectiveness of incorporating multiple background information, there still lacks sufficient evidence to prove that our methods can reduce the effect of noisy information. Thus, we compare two variations of HG, named HG(Path) and HG(KG) which only take single background information (Path/KG) respectively. (1) Comparison between HG(path) and PCNN+Path as well as HG(KG) and PCNN+KG all indicate that even under the same background information, our methods still obtain at least 5% improvement at 0.5 recall. The main reason is that since our methods utilize the graph attention layer to filter out the effect of noisy information in relation path. (2) It seems that introducing KG information performs better than relation path information, either in our model or in PCNN. We observe that KG constructed by human usually contains few noisy data. Instead, the relation path is generated with an unsupervised way which may incorporate much more noise. (3) HG obtains the best performance among all HG variation, as shown in Fig. 4(a) and (b). This demonstrates that different information can complement each other and a fusion model can leverage all information to enhance the performance. Further more, our model is capable to fuse more information efficiently, such as entity description [7], we will take these information into account for the future work.

Table 2. P@N and F1 with different percentage of no-relation facts.

(Noise)	75%				85%				95%			
P@N (%)	10%	20%	50%	F1	10%	20%	50%	F1	10%	20%	50%	F1
PCNN	86.0	68.5	38.3	57.2	85.4	67.6	37.7	56.5	84.4	66.0	36.6	54.8
PCNN+Path	89.0	71.5	39.8	59.6	89.0	71.4	39.6	59.4	88.6	71.0	39.1	59.1
HG(Path)	88.9	73.5	41.2	58.4	88.8	72.9	41.0	57.8	87.9	71.5	40.7	57.1
HG(KG)	90.1	**77.8**	40.2	61.4	90.0	**76.9**	40.0	60.9	89.7	**76.4**	42.5	60.2
HG	**92.1**	76.5	**43.2**	**63.1**	**92.1**	76.4	**43.6**	**62.6**	**92.1**	76.2	**42.9**	**62.2**

4.4 Model Robustness

In DSRE, the generated dataset consists of plenty of noisy instances. More specifically, the instances labeled "NA" which means there exist no relation between two entities are viewed as a kind of noise and the RE models might not distinguish these noisy data very well. Following the settings proposed in [25], we evaluate those models with the same relational facts and different percentages of "NA" sentences to verify the robustness. There are three groups of experiments with noise percentages are 75%, 85%, and 95%. In each experiment, we extract top 20,000 predicting relational facts according to the predicting probabilities, and report the P@N for @top 10%, @top 20%, @top 50% and F1 score in Table 2.

Evaluations in Table 2 demonstrate: (1) Our model achieves the best performance among different noise percentage and has a very slow decay with the noise percentage increase. (2) In some special cases, the models that incorporating more knowledge worse that those methods with less knowledge. This phenomenon indicates that our model can not eliminate the effects of noise completely, and we leave this issue for future work.

4.5 Case Study

Table 3 shows two representative examples of testing dataset. In each case, we use the change of scores to demonstrate the effect of background information. The first case indicates that when the sentence lacks adequate information to make

Table 3. Representative cases in testing dataset.

ex#1	**manhattan** ?country_of **America**		Score
	sentence	...marked **America**...like downtown **manhattan**...	0.242
	path_1 path_2	...**manhattan** charged the garbage haulers in **new_york**... ...**new_york**... city in **America**...	0.564
	(inference)	*manhattan* $\xrightarrow{located\_in}$ *new_york* $\xrightarrow{captial\_of}$ *America*	
	type	e_1 type: *location*, e_2 type: *country*	0.861
ex#2	**new_south_wales** ?located_in **australia**		score
	sentence	...in the **new_south_wales** in **australia**...	0.766
	path_1 path_2	in the southern alps of **victoria** and **new_south_wales** .. the_national_gallery of **australia** in **victoria**...	0.464
	(inference)	*new_south_wales* $\xrightarrow{shares\_border}$ *victoria* $\xrightarrow{country}$ *australia*	
	type	e_1 type: *state*, e_2 type: *country*	0.737

the prediction, incorporating various background information may enhance the performance significantly. For the second case, the sentence between the entity pair *new_south_wales* and *australia* includes sufficient evidence to predict the correct relation while the relation path underlies between the entity pair guide to the wrong direction and damage the capacity of classification. On the contrary, our model can capture the noise and utilize the constraints on the entity type to alleviate the effect of noisy information and correct to right relation. Generally, our method can learn a discriminative and powerful representation even with some noisy information.

5 Related Work

5.1 Distant Supervision

Distant Supervision (DS) was first proposed in [3] which focused on extracting binary relations by using a protein KG. With the development of DS, Mintz et al. [13] aligned Freebase [1] relations with New York Times (NYT) corpus to automatically generate the training data. Riedel et al. [16] proposed an assumption that there is at least one instance express the labeled relation among all sentences contains the entity pair. Following this *expressed-at-least-once* assumption, other researchers improved the original paradigm with kinds of methods. Surdeanu et al. [19] and Hoffmann et al. [6] utilized probabilistic graphics models to improve DSRE in MIL framework. Zheng et al. [26] built the inter information of aggregated inter-sentence to enhance DSRE performance. As these methods depend on the features obtained from NLP tools, so the errors derived from NLP tools will prorogate to DSRE system and effect their performance.

5.2 Neural Network Methods

With the great breakthrough of deep neural networks (DNNs), some researchers applied it for DSRE and obtained a promising result. Zeng et al. [24] firstly proposed a convolutional neural network (CNN) for relation classification. Zhou et al. [27] and Socher et al. [18] proposed to utilize bidirectional long short-term memory (Bi-LSTM) networks to model the sentence with sequential information with all words. Moreover, with the rise of attention mechanism, Lin et al. [10] employed sentence-level attention to reducing the weight of noisy data and achieves state-of-the-art. All theses methods only considered the sentence bag information and cannot deal with the case where all the sentences containing the same entities are wrongly labeled.

5.3 Methods with Background Information

The *expressed-at-least-once* assumption is often too strong in practice and previous methods are hard to deal with the cases when all sentences are not expressing the labeled relation. Therefore, some researchers introduced some background

information to expand missing information. The types and descriptions of entities from additional corpus and KG have been introduced to DS in [7,11]. Other works [5,22] attempted to combine a human constructed KG into DSRE and propose a joint representation learning framework. In addition, Zeng et al. [25] built an inference chain between two target entities via intermediate entities from the text and proposed a path-based neural relation extraction model to encode the relational semantics from both direct sentences and inference chains. These methods all achieved promising success but also suffered the low coverage problem.

6 Conclusion and Future Work

In this paper, we proposed a novel graph-based model to fuse heterogeneous information for DSRE. Firstly, we converted various pieces of information into vector representations via corresponding encoders. Then, we constructed a hybrid graph and treated each piece of information embedding into a node and connected related node with an edge. In addition, an attention mechanism was proposed to alleviate the noisy information by incorporating structured triples in a KG. We evaluated our model on a real-world dataset and results demonstrated that our model achieves huge improvement both in enhancing accuracy with heterogeneous background information and reducing the side effect of noisy information.

Our hybrid graph model is convenient to capture the relevance of various types of data, including the unlabeled data. In the future, we will generalize our model to large unlabeled text and try to learn more confident knowledge in the unsupervised relation extraction task. Moreover, the effect of noisy knowledge has not been eliminated completely and we will consider designing a more efficient method to solve this problem.

Acknowledgement. This work was supported by National Natural Science Foundation of China Key (U1736204) and National Key R&D Program of China (2018YFC0830200).

References

1. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of SIGMOD, pp. 1247–1250 (2008)
2. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NIPS, pp. 2787–2795 (2013)
3. Craven, M., Kumlien, J.: Constructing biological knowledge bases by extracting information from text sources. In: Proceedings of ISMB, pp. 77–86 (1999)
4. Guu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space. In: Proceedings of EMNLP 2015, pp. 318–327 (2015)

5. Han, X., Liu, Z., Sun, M.: Neural knowledge acquisition via mutual attention between knowledge graph and text. In: Proceedings of 8th AAAI and (EAAI-2018), pp. 4832–4839 (2018)
6. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L.S., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of ACL, pp. 541–550 (2011)
7. Ji, G., Liu, K., He, S., Zhao, J.: Distant supervision for relation extraction with sentence-level attention and entity descriptions. In: Proceedings of AAAI, pp. 3060–3066 (2017)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of ICLR (2016)
9. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: Proceedings of EMNLP, pp. 705–714 (2015)
10. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of ACL, pp. 2124–2133 (2016)
11. Liu, Y., Liu, K., Xu, L., Zhao, J.: Exploring fine-grained entity type constraints for distantly supervised relation extraction. In: Proceedings of COLING, pp. 2107–2116. ACL (2014)
12. McCallum, A., Neelakantan, A., Das, R., Belanger, D.: Chains of reasoning over entities, relations, and text using recurrent neural networks. In: Proceedings of EACL, pp. 132–141 (2017)
13. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of ACL, pp. 1003–1011 (2009)
14. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of ACL (2016)
15. Neelakantan, A., Roth, B., McCallum, A.: Compositional vector space models for knowledge base completion. In: Proceedings of ACL, pp. 156–166 (2015)
16. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Proceedings of ECML PKDD, pp. 148–163 (2010)
17. Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Proceedings of ESWC, pp. 593–607 (2018)
18. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of EMNLP-CoNLL, pp. 1201–1211 (2012)
19. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: Proceedings of EMNLP-CoNLL, pp. 455–465 (2012)
20. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing text for joint embedding of text and knowledge bases. In: Proceedings of EMNLP 2015, pp. 1499–1509 (2015)
21. Toutanova, K., Lin, V., Yih, W., Poon, H., Quirk, C.: Compositional learning of embeddings for relation paths in knowledge base and text. In: Proceedings of ACL, pp. 1434–1444 (2016)
22. Weston, J., Bordes, A., Yakhnenko, O., Usunier, N.: Connecting language and knowledge bases with embedding models for relation extraction. In: Proceedings of EMNLP, pp. 1366–1371 (2013)
23. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of EMNLP, pp. 1753–1762 (2015)

24. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING, pp. 2335–2344 (2014)
25. Zeng, W., Lin, Y., Liu, Z., Sun, M.: Incorporating relation paths in neural relation extraction. In: Proceedings of EMNLP, pp. 1768–1777 (2017)
26. Zheng, H., Li, Z., Wang, S., Yan, Z., Zhou, J.: Aggregating inter-sentence information to enhance relation extraction. In: Proceedings of AAAI, pp. 3108–3115 (2016)
27. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of ACL, pp. 207–212 (2016)