Improved Distant Supervision Relation Extraction based on Edge-Reasoning Hybrid Graph Model

Shirong Shen, Shangfu Duan, Huan Gao and Guilin Qi*

School of Computer Science and Engineeriing, Southeast University, Nanjing, China

ARTICLE INFO

Keywords:
Distant supervision
Relation extraction
Heterogeneous information
Hybrid graph
Edge reasoning

ABSTRACT

Distant supervision relation extraction (DSRE) trains a classifier by automatically labeling data through aligning triples in the knowledge base (KB) with large-scale corpora. Training data generated by distant supervision may contain many mislabeled instances, which is harmful to the training of the classifier. Some recent methods show that relevant background information in KBs, such as entity type (e.g., Organization and Book), can improve the performance of DSRE. However, there are three main problems with these methods. Firstly, these methods are tailored for a specific type of information. A specific type of information only has a positive effect on a part of instances and will not be beneficial to all cases. Secondly, different background information is embedded independently, and no reasonable interaction is achieved. Thirdly, previous methods do not consider the side effect of the introduced noise of background information. To address these issues, we leverage five types of background information instead of a specific type of information in previous works and propose a novel edge-reasoning hybrid graph (ER-HG) model to realize reasonable interaction between different kinds of information. In addition, we further employ an attention mechanism for the ER-HG model to alleviate the side effect of noise. The ER-HG model integrates all types of information efficiently and is very robust to the noise of information. We conduct experiments on two widely used datasets. The experimental results demonstrate that our model outperforms the state-of-the-art methods significantly in held-out metric and robustness tests.

1. Introduction

Relation Extraction (RE) aims at extracting semantic relations between specified entity pairs from plain text. It plays a crucial role in a wide range of applications such as Knowledge Graph (KG) completion [1] and question answering [2]. For example, given the sentence #1 in Fig. 1, RE aims to identify the *FounderOf* relation between *Bill Gates* and *Bill & Melinda Gates Foundation*. The sentence #1 is called an instance of the entity pair <*Bill Gates, Bill & Melinda Gates Foundation*>. Since the relations to be extracted are predefined and finite, relation extraction can be regarded as a classification task. A general solution for classification is based on supervised learning. A supervised learning method trains a classifier on labeled data to predict the probability distribution of each category. However, these methods suffer from the lack of labeled data because manual annotation is very costly. Distant supervision (DS) [3, 4, 5] is a promising approach to solve this limitation. It can generate labeled data automatically by aligning KGs with text. All sentences containing the specified entity pair in a KG are treated as instances of the entity pair. All instances of an entity pair form a bag, and the corresponding relation of the bag is the relation between the specified entity pair in KG. As shown in Fig. 1, if the triplet <*Bill Gates, FounderOf, Bill & Melinda Gates Foundation*> exists in KG, all the sentences containing *Bill Gates* and *Bill & Melinda Gates Foundation* are taken as the training instances of *FounderOf* relation. The method of training relation extraction model using labeled data generated by DS is called Distant Supervision Relation Extraction (DSRE).

One of the keys to DSRE is to encode instances into feature vectors. Some methods used CNN [6] and RNN [7] to encode instances. Cai et al.[7] added the shortest dependency path (SDP) in the dependency grammar to RNN. Zhou et al. [8] added the attention mechanism after an encoder to further enhance the effect of instance information. Besides, deep CNN [9] and GCN [10] were used to extract sentence features. Some work used information of entity pair to enhance the instance coding process. Zeng proposed PCNN [11] for sentence segmentation modeling. Zhang et al. [12] used GCN to encode pruned dependency trees to simplify calculations and improve the robustness of relation extraction.

Unlike supervised learning, the labeled data used by DS contains a large number of mislabeled instances. These mislabeled instances bring the noise to DSRE. For example, according to the triplet <*Bill Gates, FounderOf, Bill & Melinda Gates Foundation*> in a KG, sentence #2 in Fig. 1 is labeled as an instance of relation *FounderOf*. But sentence #2 does not imply that the relation between the two entities is *FounderOf*. So this instance maybe brings the noise to model training. Recently, various models based on Deep Neural Network (DNN) [13, 14, 6, 15] have been proposed to improve DSRE. These methods mainly focused on dealing with the noise problem brought by mislabeled instances during DS and have achieved considerable improvement. The most direct way is to use the attention mechanism to weigh instances in a bag and choose helpful information automatically [16, 17]. Based on this idea, attention mechanisms for different granularities were used to reduce instance noise[18,

*Corresponding author
ssr@seu.edu.cn (S. Shen); sf_duan@seu.edu.cn (S. Duan); gh@seu.edu.cn (H. Gao); gqi@seu.edu.cn (G. Qi)
ORCID(s): 0000-0003-0641-8033 (S. Shen)

Figure 1: Examples of DS.

19]. Qin et al. [20] considered the selection of instance information as a decision process and trained a reinforcement learning(RL) model to reduce noise. Luo et al. [21] added a noise model on the basic model, expecting to eliminate error information in DS. In addition, there are also methods to generate labels through generators and use adversarial training (AT) to improve the accuracy of sample generation [22, 23, 24].

However, existing DNN-based methods do not consider sufficient background information for making predictions. For example, as shown in Fig. 1, sentence #1 and sentence #3 are two DS generated sentences, and they describe different relations. It is hard to predict their relations precisely because they both use *create* to express corresponding relations. Recent methods attempted to introduce more background information to enhance DSRE. Methods proposed in [25, 26] used the types and descriptions of entities to provide rich entity information to RE. Other researchers [27, 28, 29] carried out the representation learning to introduce the knowledge in KG to the DS model. Zeng et al. [15] incorporated inference information hidden in extra text and proposed a path-based model to enhance DSER. Tim et al. [30] attempted to introduce logical relationships between entities into the model as additional information. These methods have shown the effectiveness of introducing background information. However, there are still three major problems remaining to be addressed.

Firstly, previous methods only considered a specific type of background information. A specific type of background information only has a positive effect on some instances and will not benefit all instances. For example, in Fig. 1, the movie *Harry Potter* is written by *J. K. Rowling* and directed by *Chris Columbus*. *J. K. Rowling* and *Chris Columbus* use the same verb *create* but their relationship with *Harry Potter* is different. In this example, entity types are not helpful for relation extraction, which requires the incorporation of additional background information.

Secondly, previous methods encoded each type of information independently and directly concatenates all the information as the final features for classification. These methods lacked reasonable interaction between information and can only encode the surface features of background information. Furthermore, with the increase of background information types, the dimension of the final features will become large, which will lead to the increase of calculation cost. So previous methods were only suitable for introducing a small amount of background information.

Thirdly, previous methods did not consider alleviating the side effect of the introduced noise of background information. This problem is mainly reflected in two aspects. On the one hand, the introduced information may be wrong. For example, the method given in [25] obtained the entity types with NLP tools, which inevitably brought some errors and hurt the RE performance. On the other hand, background information does not obey the *expressed-at-least-once* assumption, which means some information may be missing for the entire bag. The traditional attention mechanism for sentence bags was useless to alleviate the side effect of the introduced noise of background information.

To solve these problems, three goals need to be achieved. Firstly, we need to construct multiple types of background information and establish the correlations between different

types of information. Secondly, we need to design a model that can encode multiple types of background information at the same time, and realize reasonable interaction between different information. Thirdly, we need to alleviate the side effect of introducing noisy information.

To achieve the first goal, we construct five types of background information for each entity pair and corresponding instances. Then we use a hybrid graph (HG) to represent all background information and the relations between different types of information. The HG has five types of nodes, and each type of node corresponds to one kind of additional information. The edge between two nodes in the HG represents the relationship between these two nodes.

For the second goal, we propose a novel graph-based model for DSRE, which uses various types of background information. We first transform all nodes in HG into vectors with various encoders as the initial state of HG. Then we use GCN[31, 32] to encode the nodes in HG according to the correlations between the background information. Traditional GCN can only encode nodes in a graph but cannot construct edge features. However, the features of the edge between the head entity and tail entity are essential for RE. As shown in Fig. 2, the relation between entities can be abstracted as the edge (represented by dotted line) between the nodes corresponding to the entities, where each entity has a node in HG. In order to construct the edge feature and make up for the defect that GCN cannot encode edges, we propose a new edge reasoning (ER) method. We first design the reasoning and updating methods for the edge feature, and inspired by Hao et al. [33], we used the edge feature to construct the transfer function in GCN. The ER method in this paper is a powerful extension of GCN, and GCN combined with ER can encode edges and nodes simultaneously under a unified framework.

For the last goal, we further employ an attention mechanism over the graph, and assign higher weight to the more critical background information. Notice that our model is highly robust to the missing information and flexible to integrate various types of information.

The contributions of our study can be summarized as follows:

- We construct five types of background information for entity pairs and corresponding instances and abstract all the background information into a hybrid graph by connecting related information by edges. This graph provides practical and robust support for DSRE.
- We propose a novel edge-reasoning hybrid graph (ER-HG) DS model, which incorporates heterogeneous background information in a unified framework and can flexibly integrate various kinds of information. Simultaneously, ER can enhance GCN in any other graph encoding tasks. In addition, we employ an attention mechanism over the ER-HG model to alleviate the effect of introducing noisy information.
- We conduct extensive experiments on two real-world DSRE datasets and a manually labeled DSRE dataset. The results show that our model is obviously superior to the most advanced methods through various evaluation indexes.

The remainder of this paper is structured as follows. In Section 2, we briefly introduce basic terms and neural network models used in our work. In Section 3, we introduce the implementation details of our ER-HG. In Section 4, we report the evaluations of our methods on three datasets and compare it with the previous methods. Besides, we conduct ablation experiments to analyze the function of each module. In Section 5, we introduced the related works of DSRE. Finally, in Section 6, we have summarized our work and put forward the prospect.

2. Background

In this section, we briefly introduce basic terms and neural network models used in our work, such as Bidirectional Gated Recurrent Unit(GRU) and graph Convolutional Network(GCN).

2.1. Basic Terms

2.1.1. Sentence Bag and Instance

An sentence bag $S_{(h_i,t_i)}$ is a collection of all sentences containing entity pair (h_i, t_i), where h_i and t_i are the target entities for relation extraction. Each sentence in a sentence bag is denoted as an instance $I_{(h_i,t_i)}$, which has the following form,

$$I_{(h_i,t_i)} = [w_1, w_2, ..., w_l] \tag{1}$$

where w_j is the j-th word index in $I_{(h_i,t_i)}$, l is the length of $I_{(h_i,t_i)}$. For example, in Fig. 1 $S_{(J.K.Rowling,HarryPotter)}$ consists of sentence #3, sentence #4 and other instances containing *J. K. Rowling* and *Harry Potter*. Sentence #3, sentence #4 are instances in $S_{(J.K.Rowling,HarryPotter)}$. The sentence bag is the basic information of DSRE. Under the setting of DSRE, the sentence bag contains at least one sentence which implies the semantic relationship between the target entity pair.

2.1.2. Distant Supervision Relation Extraction Dataset

We denote the DSRE dataset in this paper as $\mathcal{D}$, where $\mathcal{D} = \{(S_{(h_i,t_i)}, L_i)|, i = 1, 2, ...\}$, $S_{(h_i,t_i)}$ is a sentence bag of entity pair (h_i, t_i), L_i is the label of $S_{(h_i,t_i)}$, which is the relation between h_i and t_i in a KG. For example, the relation between *J. K. Rowling* and *Harry Potter* in KG is *AuthorOf*, so in the dataset, the sentence bag $S_{(J.K.Rowling,HarryPotter)}$'s label is *AuthorOf*.

2.1.3. Entity Type

For any entity e, we can get its type y_e from a KG. The knowledge graph divides the entities by domains and gives the fine-grained entity types in different domains. For example, the fine-grained entity type of Jackie Chan is *actor*, and the domain *Jackie Chan* belongs to is *film*. Fine-grained entity types [4] provide powerful constraints between an entity pair and a relation. For example, as shown in Fig. 1,

verb *create* indicates different relations such as *AuthorOf* and *FounderOf*. If we know that the type of *Harry Potter* in sentence #3 is *book*, it's easier to predict that the relation between *J.K.Rowling* and *Harry Potter* is *AuthorOf*, not *FounderOf*.

2.1.4. Entity Representation in KG

A Knowledge Graph (KG) [34] is a collection of triples (h_i, r_i, t_i), where r_i represents the relationship between entities h_i and t_i. The representation learning of KG [35, 36, 37] aims to learn a vector embedding of both entities and relations into a low-dimensional space. Translation-based model [36] treats the labeled relation embedding $\mathbf{r_i}$ as a translation of the embeddings of $\mathbf{h_i}$ and $\mathbf{t_i}$, i.e. $\mathbf{h_i} + \mathbf{r_i} \approx \mathbf{t_i}$. In the training process, the representation of the entity in KG is obtained by satisfying the above constraints. Compared with common word vectors, entity representation in KG can better reflect the relevance of entities in the real world.

2.1.5. Relation Path

A relation path describes the flow of resources between head entity h and tail entity t [38, 39, 40, 41, 42]. In DSRE, we define the relation path between h and t as the combination of an entity sequence and a relation sequence:

$$p = \{\{h, e_1, ...e_l, t\}, \{r_{(h,e_1)}, ..., r_{(e_l,t)}\}\} \tag{2}$$

The entity sequence in relation path starts with h and ends with t, and adjacent entities in p have a relation in KG. The relation sequence in relation path stores all the relations between each pair of adjacent entity. For example, Fig. 1 shows the relation path between $BillGates$ and $Bill\&MelindaGatesFoundation$. Intuitively, a relation path between h and t can be represented as $h \xrightarrow{r_{(h,e_1)}} e_1 \xrightarrow{r_{(e_1,e_2)}} \dots \xrightarrow{r_{(e_l,t)}} t$.

Compared with the entity representation in KG, the relation path between two entities can reflect the potential complex relationship between the entities, which is an important supplement to DSRE.

2.1.6. Entity Graph-Context

The graph context of a particular entity refers to all entities that co-occur with it in a sentence, which is important supplementary information. The essence of graph context is to enrich the relevant information of entities through the unlabeled corpus, which may be missing in KG. For example, in Fig. 1, the graph context of *Chris Columbus* can reflect the close relationship between *Chris Columbus* and the film industry. In relation extraction, graph context can enrich the semantics of entities, constrain the background of entities, and help extract more fine-grained relations. Formally, the graph context of the entity e is denoted as N_e.

2.1.7. Hybrid Graph of Distant Supervision Relation Extraction

In this article, the following types of information are used as the basis for DSRE: head entity, tail entity, the type of head entity, the type of tail entity, sentence bag, the relation path between the head entity and tail entity, the entity graph-context of the head entity and the entity graph-context of the tail entity. We treat each type of information as a node and pre-define six different edges to distinguish the interrelationships between nodes. These six edge types are defined as follows:

- **ET-edge**: the edge between entity and entity type.
- **EC-edge**: the edge between entity and graph-context.
- **HB-edge**: the edge between head entity and sentence bag.
- **HP-edge**: the edge between head entity and relation path.
- **TB-edge**: the edge between tail entity and sentence bag.
- **TP-edge**: the edge between tail entity and relation path.

These nodes and the edges between nodes constitute the hybrid graph for DSRE which is shown in Fig. 2. The hybrid graph organizes information around entities, including making full use of KG information and introducing information outside of KG, thus provides richer support for DSRE task.

2.2. Bidirectional Gated Recurrent Unit

Gated Recurrent Unit (GRU) is a variant of LSTM with a more concise structure and powerful effect. Given a feature sequence matrix $M \in \mathbb{R}^{L\times D}$, $M = [m_1, m_2, \dots, m_L]$. In each time step, GRU has the following form,

$$r_t = \sigma(W_r \cdot [\vec{h}_{t-1}, m_t]) \tag{3}$$

$$z_t = \sigma(W_z \cdot [\vec{h}_{t-1}, m_t]) \tag{4}$$

$$\tilde{h}_t = tanh(W_{\tilde{h}} \cdot [r_t\vec{h}_{t-1}, m_t]) \tag{5}$$

$$\vec{h}_t = (1 - z_t)\vec{h}_{t-1} + z_t\tilde{h}_t \tag{6}$$

where $\vec{h}_{t-1} \in \mathbb{R}^{D'}$ is the output state of the previous step, D' is the size of hidden state. $\vec{h}_0$ is a trainable parameter vector. $W_r \in \mathbb{R}^{D+D'}$, $W_z \in \mathbb{R}^{D+D'}$ are parameter vectors, σ is a non-linear activation function. The whole process is denoted as follows.

$$\vec{h}_t = GRU(\vec{h}_{t-1}, m_t) \tag{7}$$

According to the calculation method of GRU, the output of each time step can refer to the information of the previous time step, but the information of the later time step is inaccessible. Therefore, the definition of the reverse GRU operation is:

$$\overleftarrow{h}_t = GRU(\overleftarrow{h}_{t+1}, m_t) \tag{8}$$

where $\overleftarrow{h}_{L+1}$ is a trainable parameter. The forward and reverse states of each time step are added as the final state h_t.

$$h_t = \overrightarrow{h}_t + \overleftarrow{h}_t \tag{9}$$

After concatenating all states together as output $H_{GRU} = [h_1, h_2,$
$..., h_L]$, the Bidirectional GRU operation is in the following form,

$$H_{GRU} = BiGRU(M) \tag{10}$$

2.3. Graph Convolutional Network

A Graph Convolutional Networks (GCN) [43, 44] 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 $X \in \mathbb{R}^{N\times D}$ and an adjacency matrix $A \in \mathbb{R}^{N\times N}$, where N is the number of nodes and D denotes the size of node features. 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:

$$H^{(l+1)} = f(H^{(l)}, A) \tag{11}$$

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

In each layer, the layer-wise propagation is shown in Eq. 12.

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^l) \tag{12}$$

The above is the original GCN form. From Eq. 12, for the j-th node, the updating process in step $l+1$ is as follows,

$$h_j^{(l+1)} = \sigma(\sum_{i=1}^{N} A_{i,j} h_i^{(l)} W^l) \tag{13}$$

where $A_{j,i}$ is the (j,i) element in adjacency matrix A, $h_i^{(l)}$ is the i-th nodes state in the l-th layer. It can be seen that all nodes connected with h_j are equivalent. These nodes are added together by the same linear transformation to obtain a new h_j. This method cannot reflect the particularity of edges and the difference between nodes, so we introduce some variants[31].

In order to reflect the information difference of edges, different linear transformations are defined for each edge.

$$h_j^{(l+1)} = \sigma(\sum_{i\in\{i,A_{i,j}=1\}} h_i^{(l)} W_{(i,j)}^l + b_{(i,j)}^l) \tag{14}$$

where $W_{(i,j)}^l$ is the transition matrix of edges (i,j), $b_{(i,j)}^l$ is the corresponding bias term.

In addition, in order to reflect the role of each node and reduce the impact of noise, a scalar gate is calculated for each edge as follows,

$$g_{i,j}^{(l)} = sigmoid(h_i^{(l)} \hat{W}_{(i,j)}^l + \hat{b}_{(i,j)}^l) \tag{15}$$

where $\hat{W}_{(i,j)}^l$ is a matrix parameter and $\hat{b}_{(i,j)}^l$ is a bias, $sigmoid$ refers to sigmoid activation function. Then the propagation of update the j-th node has the following form,

$$h_j^{(l+1)} = \sigma(\sum_{i\in\{i,A_{i,j}=1\}} g_{i,j}^{(l)} (h_i^{(l)} W_{(i,j)}^l + b_{(i,j)}^l)) \tag{16}$$

After that, all $h_j^{(l+1)}$ are concatenated as $H^{(l+1)}$. After finite-step iterations, the feature vectors of all nodes are concatenated as the final output of GCN. This process is denoted as,

$$F = GCN(X, A) \tag{17}$$

3. Methodology

In this section, we introduce the encoders of background information and present our edge-reasoning hybrid graph model, which can effectively fuse rich heterogeneous background information and alleviate the side effect of introducing noise.

3.1. Problem Definition

Given an entity pair (h,t) , DSRE uses the sentence bag $S_{(h,t)}$ to extract the relation between h and t. In this paper, we train a DSRE model on $\mathcal{D} = \{(S_{(h_i,t_i)}, L_i)|, i = 1, 2, \dots\}$ to predict the relation between an entity pair from a given sentence bag. We finally learn a probability distribution $P(r|h,t,$
$S_{(h,t)};\theta)$ over all relations $r \in \mathcal{R}$, where θ denotes the parameters in our model. In addition, we extend entity type, relation path and graph context as background information, and train entity representation in KG as a supplement.

3.2. Overview

The overview of our approach is shown in Fig. 2. When carrying out the RE task, our model considers the sentence bag generated by DS and 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 consists of three stages. Firstly, all types of information are converted into vector representations by using corresponding encoders. Then, we construct a hybrid graph by treating each type of information as a node and connecting related information. Next, we propose an edge-reasoning graph convolution model to construct a more discriminative representation of each type of information and generate the relation feature between entities. Finally, we use an attention mechanism to extract 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 for heterogeneous background information.

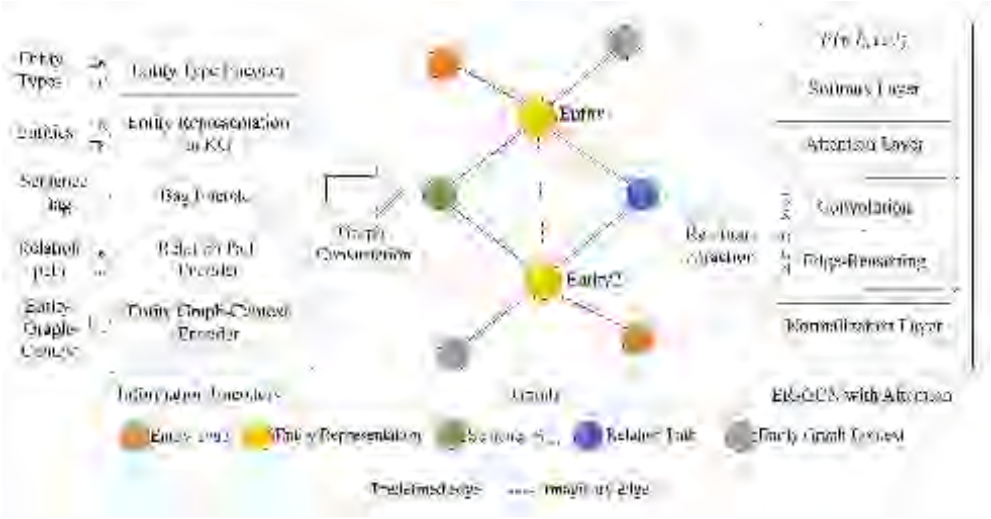


Figure 2: Overview of our method.

3.3.1. Instance Encoder

In order to extract the relationship between entity pairs (h, t), the context information in instance $I_{(h,t)}$ is essential. Firstly, each word in a sentence is mapped to a d_w-dimensional word vector, and $I_{(h,t)}$ is embedded into embedding matrix E_w.

$$E_w = [e_1, e_2, ..., e_L] \tag{18}$$

where e_i is the embedding vector of w_i. In addition, we add the position embedding of each word relative to the entity. A word's relative position is the difference between its position in the sentence and the position of the entity. We get,

$$p_{h,i} = i - p_h \tag{19}$$

$$p_{t,i} = i - p_t \tag{20}$$

where p_h and p_t are the positions of h and t, $p_{h,i}$ and $p_{t,i}$ are the relative position of the i-th word to h and t. By embedding each relative position into a 5-Dimensional vector, we get,

$$E_{ph} = [e_{p_{h,1}}, e_{p_{h,2}}, ..., e_{p_{h,L}}] \tag{21}$$

$$E_{pt} = [e_{p_{t,1}}, e_{p_{t,2}}, ..., e_{p_{t,L}}] \tag{22}$$

The final embedding matrix is obtained by concatenation of E_w, E_{ph} and E_{pt}.

$$E = E_w \oplus E_{ph} \oplus E_{pt} \tag{23}$$

BiGRU [16] is used to encode the sentence embedding matrix. BiGRU is a classical RNN structure, which can effectively acquire sequence features. Based on the definition in section 2.2, the calculation of the encoder can be expressed as follows,

$$H_g = BiGRU(E) \tag{24}$$

The attention mechanism uses the weighted sum of each feature in the feature sequence to represent the information of the whole sequence. Assume that the input is $H \in \mathbb{R}^{L \times d_i}$, L is the length of H and d_i is the dimension of each feature vector, an attention mechanism operates as follows,

$$u_j = tanh(H_j W_k + b_k) \tag{25}$$

$$a_j = \frac{exp(u_j \cdot q)}{\sum_{i=1}^{L} exp(u_i \cdot q)} \tag{26}$$

$$E_I = \sum_{j=1}^{L} a_j H_j \tag{27}$$

where H_j is the j-th vector in H, $W_k \in \mathbb{R}^{d_i \times d'}$ is a parameter matrix, $b_k \in \mathbb{R}^{d'}$ is a bias term, $q \in \mathbb{R}^{d'}$ is the trainable query vector. $E_I \in \mathbb{R}^{d_i}$ is the embedding vector of input instance $I_{(h,t)}$. In summary, the instance encoder encodes the input $I_{(h,t)}$ to a embedding vector E_I of length d_i. It is also flexible to replace with other models as long as it can represent the semantics of the given sentence.

3.3.2. Sentence Bag Encoder

For any instance $I^j_{(h,t)} \in S_{(h,t)}$ with an entity pair (h, t), we obtain its d_i-dimensional feature vector by the approach in Instance Encoder. Afterward, we get the sentence bag representation $\mathbf{S}_{(h,t)} \in \mathbb{R}^{d_i}$ by summing all sentence embeddings with different weights. There are some methods to calculate the weights, and we follow the best-performing method used in [15]. Specifically, the embedding vectors of each $I^j_{(h,t)} \in S_{(h,t)}$ is E_{Ij}, we use an attention mechanism to aggregate all instances with different weights.

$$u^I_j = tanh(E_{Ij} W^I_k + b^I_k) \tag{28}$$

$$a_j^I = \frac{exp(u_j^I \cdot q^I)}{\sum_{i=1}^{|S_{(h,t)}|} exp(u_i^I \cdot q^I)} \tag{29}$$

$$\mathbf{S}_{(h,t)} = \sum_{j=1}^{|S_{(h,t)}|} a_j^I E_{I^j} \tag{30}$$

where $W_k^I \in \mathbb{R}^{d_i \times d'}$ is a parameter matrix, $b_k^I \in \mathbb{R}^{d'}$ is a bias term, $q^I \in \mathbb{R}^{d'}$ is the trainable query vector. $\mathbf{S}_{(h,t)} \in \mathbb{R}^{d_i}$ is the embedding vector of sentence bag $S_{(h,t)}$.

3.3.3. Entity Type Encoder.

Given the type set of all entities $\mathcal{T}$, $y_i \in \mathcal{T}$ is the i-th entity type in $\mathcal{T}$. We use a real vector as the feature of y_i. All these vectors form the representation matrix $\mathbf{M}_y \in \mathbb{R}^{|\mathcal{T}| \times d_t}$, where d_t denotes the size of entity type embedding. Given an entity pair(h, t), the entity type embedding of h is denoted as $\mathbf{y}_h$ and the entity type embedding of t is denoted as $\mathbf{y}_t$. $\mathbf{y}_h$ and $\mathbf{y}_t$ are selected from $\mathcal{T}$ by the entity types of h and t. In practice, the vector representation of each $y_i \in \mathcal{T}$ is initialized randomly. We will update the representation matrix $\mathbf{M}_y$ dynamically during the training process.

3.3.4. Entity Encoder

Each entity mention e_i will be mapped into a real-value vector $\mathbf{e}_i$ with dimension d_e. We use a pre-trained PTransE model[45] to get the embedding of all the entities. For a given entity pair (h, t), the entity embedding of h is denoted as $\mathbf{h}$ and the entity embedding of t is denited as $\mathbf{t}$. 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.

3.3.5. Relation Path Encoder.

We propose a relation path encoder for joint inference among multiple paths. The entity embedding and relation embedding are integrated into our relation path encoder to represent the flow of information on a path.

As shown in Fig. 3, given an entity path p between an entity pair (h, t), we use the LSTM model to encode the relation path by capturing the flows from h to t. For any LSTM cell at timestep k, we firstly concatenate corresponding entity embedding and relation embedding, as shown in Eq. 31,

$$x_k = W_e(\mathbf{e}_k \oplus \mathbf{r}_{(e_k, e_{k+1})}) + b_e, \tag{31}$$

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 $\oplus$ is the concatenation operation. Here, d_p denotes the size of path embedding.

Afterward, each process step is written as Eq. 32, where $LSTM(x, h, c)$ denotes a standard LSTM cell [13], 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 $\mathbf{p} \in \mathbb{R}^{d_p}$.

$$h_{k+1} = LSTM(x_{k+1}, h_k, C_k) \tag{32}$$

Following the MIL framework [3, 16], 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.

3.3.6. Entity Graph-Context Encoder

For any entity e, we collect all entities that occur with it in a sentence as its graph-contex N_e and encode the graph-context into a vector representation $\mathbf{N}_e \in \mathbb{R}^{d_e}$, as shown in Eq. 33.

$$\mathbf{N}_e = \sigma\left(\frac{1}{|N_e|} \sum_{e' \in N_e} \mathbf{e}'\right) \tag{33}$$

where $\mathbf{e}'$ is the vector representation of entity e' generated in Section 3.3.4.

3.4. Edge-Reasoning Graph Convolutional Network

One crucial challenge for fusing heterogeneous information is that they have different embeddings methods. Moreover, the interrelationships between different information are different. Hence, we add specific edges between these pieces of information to form a hybrid graph to describe the relationship between them and use the adjacent matrix to represent the correlation between each piece of information. For a given entity pair (h, t) and embeddings of various background information, we apply an edge-reasoning GCN to extract high-level features of each node and generate relation features between target entities. Then we use an attention layer to aggregates all these features. Finally, the model calculates the probability of each relation r with a non-linear classifier.

3.4.1. Normalization Layer

We propose an adaptive framework to transform various embeddings into a graph structure of the hybrid graph. 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 background information and $\mathbf{A}_{i,j} = 1$ indicates that i_{th} and j_{th} elements are correlated. In this paper, as shown in Fig. 2, there are eight nodes in the hybrid graph of an entity pair (h, t). We build the node matrix $\mathbf{X} = \{\mathbf{S}_{(h,t)}, \mathbf{P}_{(h,t)}, \mathbf{h}, \mathbf{y}_h, \mathbf{t}, \mathbf{y}_t, \mathbf{N}_h, \mathbf{N}_t\}$ by the encoders in Section 3.3, where $\mathbf{h}$ and $\mathbf{t}$ are embeddings of the given entity pair, $\mathbf{y}_h$ and $\mathbf{y}_t$ are corresponding entity type emebddings, $\mathbf{N}_h$ 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 $d_i = d_p = d_e = d_t$, thus the node matrix can be denoted as $\mathbf{X} \in \mathbb{R}^{d_g \times d_i}$.

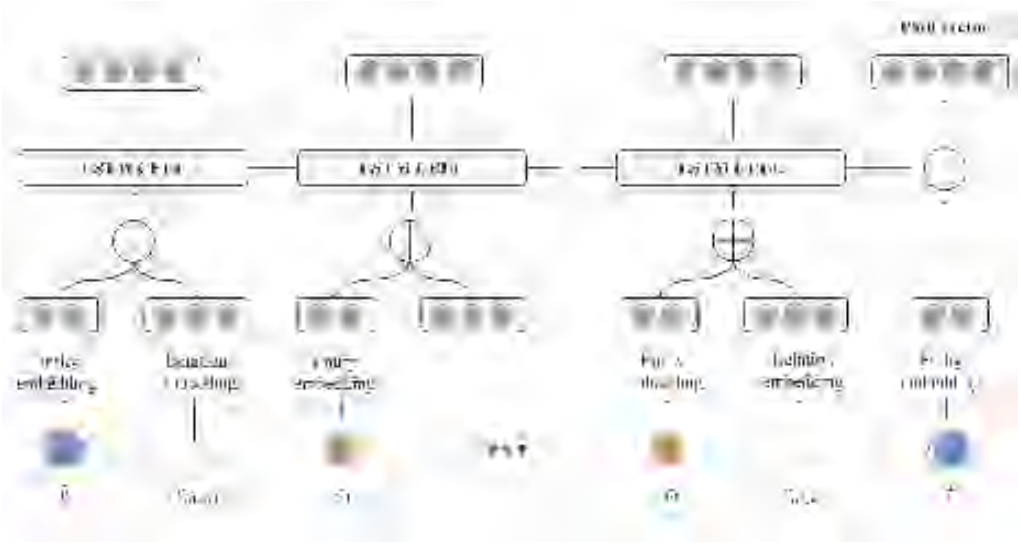


Figure 3: The architecture of our path encoder.

3.4.2. Convolutional Layer

We propose a variant of graph convolutional network[43, 44] which takes the node matrix $\mathbf{X}$ and the corresponding adjacency matrix $\mathbf{A}$ as input and generates new node features through a graph convolution operation. Edges of the same type in the hybrid graph share the same transition matrix mentioned in section 2.3. So that it is efficient 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 is space-efficient and time-efficient. As shown in Eq. 34, 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.

$$\mathbf{G} = GCN(\mathbf{X}, \mathbf{A}) \tag{34}$$

3.4.3. Edge-Reasoning

However, the traditional GCN only re-encodes the nodes and does not reason about the relation between entities. The relation between entities can be seen as an edge connecting two entities. Intuitively, we can use other information in the graph to construct the feature of this edge. Furthermore, we hope that the new edge can help node encoding as well as the predefined edge. Therefore, we design an edge reasoning method to extend the GCN to encode edges and nodes simultaneously in a unified framework.

Suppose we want to reason about the relation between nodes i and j, then we construct an edge connecting these two nodes and set $\mathbf{A}_{i,j} = 1$. As shown in Fig. 2, two entities are connected with an imaginary edge in the graph. The embedding vector $\mathbf{X}_i, \mathbf{X}_j$ corresponding to nodes i, j are used to generate the initial feature vector of the constructed edge as follows,

$$eg_{i,j}^{(0)} = G(\mathbf{X}_i, \mathbf{X}_j) \tag{35}$$

the generation function G is defined as,

$$G(\mathbf{X}_i, \mathbf{X}_j) = \sigma([\mathbf{X}_i; \mathbf{X}_j - \mathbf{X}_i; \mathbf{X}_j]W_g + b_g) \tag{36}$$

where $W_g \in \mathbb{R}^{3d_i \times d_i}$ and $b_g \in \mathbb{R}^{d_i \times 1}$ are parameters of generation function. By applying the node update strategy, the edge feature update method in step l is given as follows,

$$\hat{eg}_{i,j}^{(l+1)} = G(h_i^{(l)}, h_j^{(l)}) \tag{37}$$

$$ge_{i,j}^{(l+1)} = sigmoid(W_{eg_{i,j}}^{l} \hat{eg}_{i,j}^{(l+1)} + b_{eg_{i,j}}^{l}) \tag{38}$$

$$eg_{i,j}^{(l+1)} = (1 - ge_{i,j}^{(l+1)}) eg_{i,j}^{(l)} + ge_{i,j}^{(l+1)} \hat{eg}_{i,j}^{(l+1)} \tag{39}$$

where $ge_{i,j}^{(l+1)}$ is a gate unit which controls the weight of each step update. Only the features of the head node and tail node are used to reason with edge features, because during the iterative process, the head node and tail node can contain information of other neighboring nodes through the information transmission of GCN. This concise way can effectively leverage all information related to entities in the graph.

The information transfer of the nodes at the sides of each pre-defined edge is achieved through a specific transition matrix in GCN. We use the method of parameter generation given in [33] to construct the transition matrix $W_{i,j}^{(l)}$ by $eg_{i,j}^{(0)}$. The inferred edges can participate in the node encoding just like other edges under the unified framework. The features of edges obtained by iterative reasoning are taken as an output of GCN in the same way as node encoding in $\mathbf{G}$, so we get $d_g + 1$ vectors in $\mathbf{G}$. In addition, ER can generate multiple edge features at the same time, which is suitable for any graph model.

3.4.4. Attention Layer

During the RE process, some extra information is invalid and may mislead the model. For example, in Fig. 1, the relation path {*Bill Gates, Melinda, Bill & Melinda Gates Foundation*} has no positive effect on extracting relation between

Bill Gates and *Bill & Melinda Gates Foundation*. Thus, it is important to filter the features generated by the graph model. We use an attention layer to learn a discriminative representation among all background knowledge embeddings and the inferred edge feature. Inspired by the translation-based models, we use an approximation representation of the relation between entity pair (h, t) as the query of attention mechanism, as shown in Eq. 40. Then, we calculate the similarity between each background knowledge embedding and the inferred edge feature $\mathbf{G} = \{\mathbf{g}_1, ..., \mathbf{g}_{d_g+1}\}$ with the approximation relation representation $\mathbf{r}_{h,t}$, shown in Eq. 42.

Afterward, we apply a weighted sum operation over all features on the graph and assign a high weight to more relevant features to alleviate the effect of noise, which is formulated by Eq. 43:

$$\mathbf{r}_{h,t} = \mathbf{t} - \mathbf{h} \tag{40}$$

$$\mathbf{u}_j = tanh(\mathbf{W}_a \mathbf{g}_j + \mathbf{b}_a), \tag{41}$$

$$a_j = \frac{exp(\mathbf{r}_{h,t} \cdot \mathbf{u}_j)}{\sum_{k=1}^{d_g+1} exp(\mathbf{r}_{h,t} \cdot \mathbf{u}_k)}, \tag{42}$$

$$\mathbf{r}_g = \sum_{j=1}^{d_g+1} a_j \mathbf{g}_j, \tag{43}$$

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, $\mathbf{r}_g$ is the aggregate feature of the graph.

Finally, for a given entity pair (h, t) and the corresponding sentence bag $S_{(h,t)}$, we calculate the probability of each relation r as follows,

$$\mathbf{o} = \mathbf{M}\mathbf{r}_g, \tag{44}$$

$$P(r|h, t, S_{(h,t)}) = \frac{\exp\left(o_r\right)}{\sum_{r' \in \mathcal{R}} \exp\left(o_{r'}\right)} \tag{45}$$

where r is the gold label of $S_{(h,t)}$, $\mathbf{M}$ 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 scores of all relations. d_r is the number of candidate relations. Eq. 45 is the normalization operation, which outputs the probability corresponding to each relation.

3.5. Optimization

For an entity pari (h, r), the overall objective function is defined in Eq. 46, where θ denotes the parameters in our model.

$$\ell_\theta\left(r|h, t; \theta\right) = \log(P\left(r|h, t, S_{(h,t)}\right)) \tag{46}$$

Set	#Sentences	#Entity pairs	#Facts
Train	570,088	233,064	156,664
Valid	234,350	58,635	15,104
Test	172,448	96,678	6,444

Table 1: Statistics of the NYT-Freebase dataset proposed by Riedel et al.[5].

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

Table 2: Statistics of the NYT-Wikidata dataset proposed by Zeng et al.[15].

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

$$\min_\theta \mathbb{E}\left[\sum_{S_{(h,t)} \in \mathbb{S}_{batch}} \frac{\ell_\theta\left(r|h, t, \theta\right)}{|\mathbb{S}_{batch}|}\right] \tag{47}$$

where $\mathbb{S}_{batch}$ is a set of sentence bag. $|\mathbb{S}_{batch}|$ is the number of sentence bag in $\mathbb{S}_{batch}$.

4. Experiments

Our experiments aim to verify (1) whether introducing additional background information can improve the prediction performance in DSRE, and (2) whether our edge-reasoning hybrid graph model can extract more valuable features and 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 is proposed in [5], and this dataset is generated by aligning Freebase[46] relations with the New York Times(NYT) corpus. The detailed statistics of NYT-Freebase are shown in Table. 1. Due to the sparse background information in Freebase, experiments on NYT-Freebase cannot reflect the effect of background information on DSRE. Therefore, we used another up-to-date dataset proposed in [15], which addresses the above issue by aligning Wikidata[2] relations with the NYT corpus. Compared to the NYT-Freebase dataset, this dataset contains more instances. The detailed statistics of NYT-Wikidata are shown in Table. 2.

Following the previous work [4], we evaluated our model based on the held-out metric. This metric provides an approximation measurement of precision by comparing the predicted relations with corresponding facts in KG. In addition, we also evaluated the robustness and capacity of different percentages of noisy data to illustrate the effectiveness of our model in reducing noise edge effects. We adjusted the

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

Set	#Sentences	#Bags
Train	107,093	17,515
Test(DS)	9,955	3,735
Test(Human)	5,202	2277

Table 3: Statistics of the NYT-H dataset proposed by Zhu et al.[47]. DS means the test data generated by distant supervision, Human means the test data generated by human annotation.

noise ratio on NYT-Wikidata, compared it with other models using the same background information, and reported the Precision@N results and F1 score with different noise ratios. The reason for the comparison on NYT-Wikidata instead of the other data is that the NYT-Wikidata dataset has richer background information, so the corresponding noise ratio adjustment space is relatively large. Moreover, the information coverage of the other data is relatively low, so adjusting the noise ratio will not have a significant effect on the experimental results.

In addition, because the distant supervision dataset contains much noise, traditional testing methods are also tested in a noisy environment, which cannot reflect the model's actual performance. So we use the new DSRE dataset NYT-H [47] to evaluate our method. NYT-H provides the **M**anual **A**nnotation as **G**round **T**ruth (MAGT) method which uses the human-annotated labels as truths. Since NYT-H uses the method of manually labeling the test set, it supports two evaluation strategies. The first is the bag2bag track commonly used by DSRE, which evaluates the methods trained with sentence bag and tested with sentence bag. The second is the bag2sent track which is more biased towards practical applications. Bag2sent track evaluates the methods that are trained with sentence bag and tested with a single sentence. The detailed statistics of NYT-Wikidata are shown in Table. 3.

4.2. Experimental Settings

We applied different combinations of parameters in a validation dataset, and the optimal parameters are shown as follows. We used a pre-trained word embeddings on NYT corpus with different embedding size $d_w = \{50, \mathbf{100}, 200, 300\}$. For entity embedding, we trained a PTransE model[45] with the dimension d_e among $\{\mathbf{50}, 100, 200, 300\}$. We selected 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 applied dropout on the last layer to avoid overfitting with a dropout rate 0.5. Other parameters have little effect, and we followed the settings used in [15].

4.3. Precision-Recall Curve Comparison

To demonstrate the performance of our model, we compared it with other methods via the held-out evaluation. The models used for comparison include:

- PCNN represents the Piecewise-CNN model with multi-instance learning used in [6].
- GCN_PDT [12] is a method that encodes pruned dependency trees of instances by GCN.
- RA_BAG_ATT [19] is a method that uses a multi-granular attention mechanism (relation-aware and inter-bag) to reduce introduced noise.
- PCNN+KATT [29] it a method that uses a knowledge-aware attention mechanism to integrate the relational knowledge into relation extraction model.
- ToHRE [48] is the method that uses a top-down classification to strengthen the relation extraction model.
- ER-HG is our method with all background information and enhanced edge-reasoning GCN model.

Fig. 4 shows the P-R curves upon the held-out evaluation, where (a) and (c) demonstrate the overall evaluation results, (b) and (d) show the results of the ablation experiment reflecting the role of each part of our method.

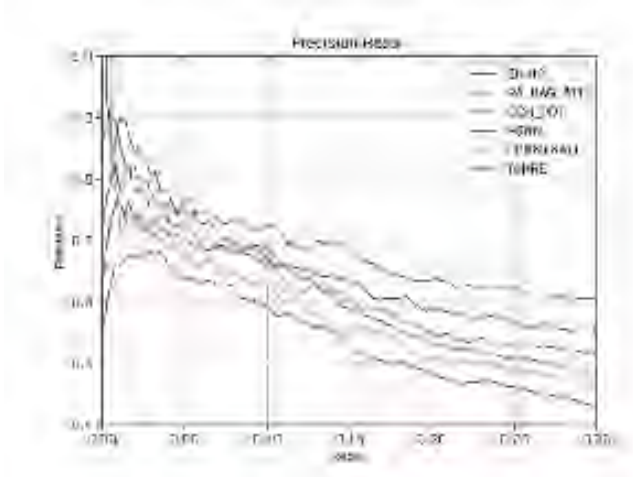
4.3.1. Overall Results

The overall precision/recall curves on NYT-Freebase and NYT-Wikidata are presented in Fig. 4 (a) and (c), we observe that:

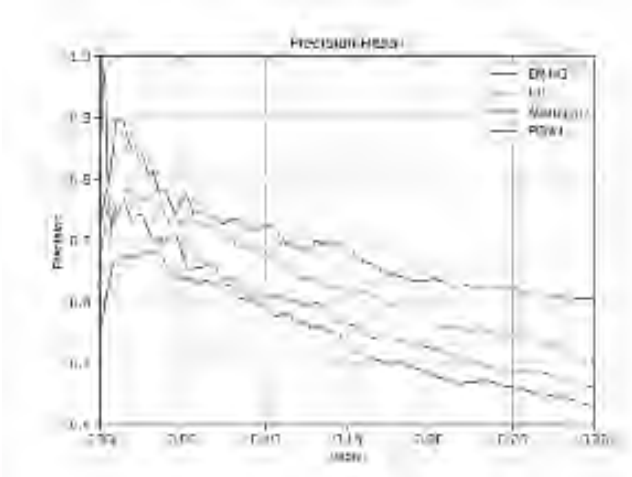
(1) On both datasets, our method (ER-HG) outperforms over all other methods. When the recall is the same, ER-HG achieves higher precision on both datasets. Especially on NYT-Wikidata, ER-HG achieves the best performance with 35% improvement in precision to PCNN, more than 15% improvement to GCN_PDT and RA _BAG_ATT and 5% improvement to SOTA methods ToHRE and PCNN+KATT when recall equals 0.5. This observation indicates that incorporating external background information with ER-GCN improves the performance of RE.

(2) The higher performance of GCN_PDT and RA_BAG _ATT over PCNN shows that fine-grained features help improve the performance of the model. RA_BAG_ATT performs much better than GCN_PDT, shows that the noise reduction on DS data is required. PCNN+KATT performs better than GCN_PDT and RA_BAG _ATT indicates that external knowledge is helpful for DSRE. The higher performance of ToHRE over PCNN+KATT shows that introducing prior knowledge at the relation label level is more effective than using entity-level knowledge directly. These methods still have a particular gap compared with our model (ER-HG), it shows that the methods using only single external information have drawbacks, and introducing more additional information can make up for this.

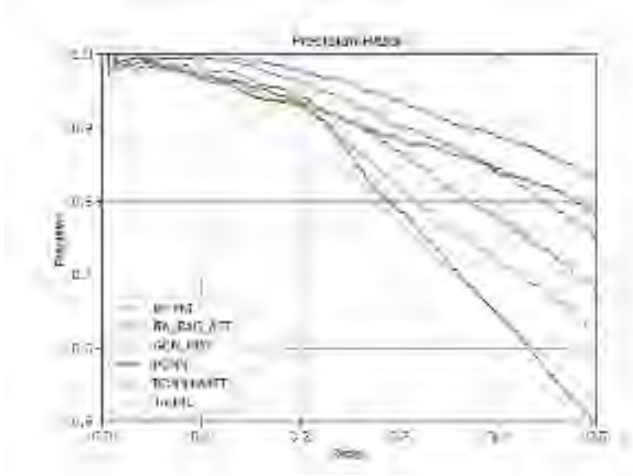
(3) The improvement of ER-HG on NYT-Wikidata is more evident than it is on NYT-Freebase, which shows that our method performs better with more background information. Although the background information on NYT-Freebase is not sufficient, ER-HG still outperforms other methods, reflecting that our method is also beneficial to RE in the case of low-quality background information.



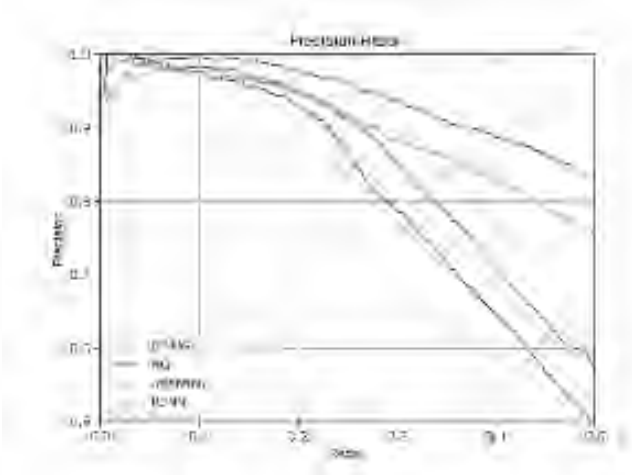
(a) Overall results on NYT-Freebase.



(b) Results of ablation experiments on NYT-Freebase.



(c) Overall results on NYT-Wikidata.



(d) Results of ablation experiments on NYT-Wikidata.

Figure 4: Aggregate PR curves upon the held-out evaluation. (a) and (c) shows the comparison between our methods and other baselines respectively. (b) and (d) demonstrates the effect of each module in our proposed method on DSRE.

4.3.2. Ablation Experiment

While previous experiments demonstrate the effectiveness of incorporating multiple background information, there still lacks sufficient evidence to prove that our method can better encode a variety of additional information and reduce the impact of noisy information. Thus, we designed an ablation experiment to evaluate the effect of each part of the model on the improvement of DSRE. So we compared ER-HG to the following models.

- HG is our method, which uses additional information and pure GCN model mentioned in this paper.
- Attention is the method that uses an attention mechanism to aggregate all external information without the graph model.

Furthermore, we add PCNN as a baseline for comparison. Fig. 4 (b) and (d) present the precision/recall curves of these methods.

(1) We obverse that HG and ER-HG outperform over other methods with the same background information on both datasets. The results indicate that the graph-based model can encode background information better than previous methods, and the attention layer after the graph can effectively reduces the impact of noise in background information.

(2) The result of PCNN with background information performs worse than Attention, which can further prove the effectiveness of introducing additional information. On NYT-Freebase, the improvement of Attention to PCNN is not apparent, which also shows that in the case of lack of knowledge, directly using the attention mechanism to aggregate additional information cannot dig out effective information well.

(3) On NYT-wikidata, when the recall is low (0-0.25), the performances of HG and Attention are almost equal, and when the recall is high(0.25-0.5), HG performs better than Attention. The results show that using the graph model can effectively improve the model effect on the relatively tricky problems of relation extraction (with low confidence). The experimental results illustrate the necessity of using the graph model to encode additional information.

(Noise)	75%				85%				95%			
P@N (%)	10%	20%	50%	F1	10%	20%	50%	F1	10%	20%	50%	F1
PCNN	89.0	71.5	41.2	58.4	88.4	72.9	41.0	57.8	88.0	71.5	40.7	57.1
GCN_PDT	89.6	72.9	42.6	59.8	89.1	73.3	42.1	58.9	88.4	73.1	40.9	58.1
RA_BAG_ATT	89.5	74.1	42.2	60.5	88.8	74.1	42.0	60.0	88.7	74.1	41.9	59.6
PCNN+KATT	90.1	75.3	42.9	60.9	90.0	75.1	42.3	60.3	89.6	74.9	42.1	59.9
ToHRE	91.3	75.8	42.7	61.3	91.0	75.4	42.6	60.8	90.5	75.0	42.2	60.2
HG	91.9	76.1	43.0	61.7	91.3	75.5	43.0	61.0	91.2	75.3	42.7	60.8
ER-HG	**92.5**	**77.2**	**43.9**	**62.1**	**92.3**	**77.3**	**43.5**	**61.3**	**92.1**	**77.1**	**43.4**	**61.0**

Table 4: P@N and F1 with different percentage of no-relation facts on NYT-Wikidata with same background information.

(4) ER-HG achieves higher precision than HG over the entire recall range on both NYT-Wikidata and NYT-Freebase. The results indicate that ER can help the graph model to extract more profound features from additional information. It also shows that the ER-HG has a more vital ability in relation prediction than a simple graph model.

4.4. Model Robustness

In DSRE, the generated dataset consists of plenty of noisy instances. More specifically, the label "NA" means there is no relation between two entities that are viewed as a kind of noise, and the RE models might not distinguish these noisy data very well. Following the settings proposed in [15], we evaluated those models with the same relational facts and different percentages of "NA" sentences to verify the robustness. There are three groups of experiments with different noise percentages: 75%, 85%, and 95%. We compared our model with other baseline models with the same background information on NYT-Wikidata. We extracted the top 20,000 predicting relational facts according to the predicting probabilities and reported the P@N for @top 10%, @top 20%, @top 50% and F1 score in Table. 4. In order to prevent the influence of different NA sampling on the evaluation results, we conducted five random "NA" sampling for each group of experiments, and the average value of the five experiments was taken as the final result. The evaluations results is shown in Table. 4. We observe that:

(1) Our model achieves the best performance among different noise percentages, which indicates that our model makes more reasonable use of background information. The graph model adds interactions between different information and updates each node and edge feature at each step using the gate unit to control the interaction information.

(2) The performance of our methods has a prolonged decay with the noise percentage increase. It indicates that the attention layer over the graph improves the robustness of the model with different noise percentages. There is no coping mechanism for this kind of noise in the traditional method in the traditional methods. Therefore, when the noise percentage increase, the performance of traditional methods decay rapidly. In our ER-HG and HG models, the information generated through independent channels is further encoded by graph convolution, and the attention mechanism filters the final output, so the model's ability to deal with noise is improved.

Tracks	Models	P(%)	R(%)	F1(%)
Bag2bag	PCNN	48.3	28.4	35.7
	RA_BAG_ATT	52.2	33.5	40.8
	GCN_PDT	53.7	34.3	41.8
	PCNN+KATT	55.4	36.9	44.2
	ToHRE	56.7	38.4	45.7
	ER-HG	**58.9**	**41.1**	**48.4**
Bag2sent	PCNN	46.5	21.2	29.1
	RA_BAG_ATT	50.1	29.7	37.2
	GCN_PDT	49.6	31.8	38.7
	PCNN+KATT	51.8	31.3	39.0
	ToHRE	52.2	**34.2**	41.3
	ER-HG	**55.1**	34.0	**42.1**

Table 5: Experiment results on NYT-H.

(3) The performance of ER-HG in precision and F1 value is better than HG in any proportion of noise, which shows that edge reasoning can obtain features that are more conducive to relation extraction, and the effectiveness of these features is resistant to noise.

4.5. Experiment on Human Annotation Dataset

Since the DS test set contains noise, we used the artificially labeled number test set in NYT-H to test the performance of our model. This part of the experiment includes two evaluation strategies, bag2bag track, and bag2sent track. The bag2bag track tests the model's actual performance in DSRE through the human annotation high-precision sentence bag. The bag2sent track uses the DSRE model to extract relations from a single instance, which can test the ability of the model to recognize relationships in a single instance. We compared our ER-HG with other SOTA DSRE methods on both bag2bag track and bag2sent track of NYT-H and calculate precision(P), recall(R), and F1 value of each model. Table. 5 shows the experiment results on NYT-H.

From the results in Table. 5, we can observe the following facts:

(1) In bag2bag track, ER-HG achieves the highest F1 value, which shows that ER-HG can still achieve the best performance on manually labeled data. Compared with other models, ER-HG can overcome the noise problem in training data better.

(2) The performances of ToHRE and PCNN+KATT better than RA_BAG_ATT and GCN_PDT can further illus-

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

Table 6: Representative cases in testing dataset.

trate the effectiveness of introducing external knowledge, which is consistent with the previous experimental results in Section 4.3.

(3) ER-HG is better than ToHRE and PCNN+KATT shows that it is necessary to use more additional information in DSRE, and graph-based edge reasoning on additional information can extract relational features more accurately.

(4) There is a gap between the experimental results of all models on the bag2sent track and bag2bag track, indicating that the distant supervision method still cannot effectively solve the problem of relation extraction from a single instance.

(5) ER-HG has the best F1 score on bag2sent track, which shows that ER-HG can better extract the relation features in a single sentence after distant supervision training, which should benefit from the rich additional information provided by the graph model to assist secondary encoding.

(6) ToHRE has the best recall, which may be because the top-down hierarchical relationship representation can better encode the relation features in distant supervision training. However, the precision of ToHRE is not as good as that of ER-HG, indicating that using more additional information in relation extraction can improve the accuracy of relation classification.

In conclusion, experiments verify that introducing rich background information is helpful for DSRE. Besides, ER-HG can extract deeper relational features through interactive coding between different information and reduce the side effects caused by noise in the background information. Simultaneously, compared with traditional GCN, the edge reasoning method can further improve the accuracy of relation prediction, but its effect is affected by the information coverage.

4.6. Case Study

In some exceptional cases, the model performs worse after adding more additional information. It shows that the impact of the introduction of wrong knowledge on the model is often irreversible. At present, our model's treatment of noise is still at a macro level, and it cannot deal with the impact of all types of noise. This phenomenon indicates that our model can not eliminate the effects of noise, and we leave this issue for future work.

Table. 6 shows two representative examples of the test 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 the prediction, incorporating various background information may significantly enhance performance. For the second case, the sentence between *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 robust representation even with some noisy information.

5. Related Work

5.1. Distant Supervison

Relation extraction (RE) is considered as a multi-class classification problem but suffer from lack of large labeled training data. To address this limitation, distant supervison (DS) was first proposed in [49] which focused on extracting binary relations by using a protein KG. With the development of DS, Mintz et al. [4] aligned Freebase [46] relations with New York Times (NYT) corpus to automatically generate a large training dataset. The work of Mintz et al. was based on the assumption of *expressed-at-least-once*, however the assumption does not always hold true. To alleviate this shortcoming, Riedel et al. [5] relaxed DSRE to a multi-instance learning (MIL) problems. Surdeanu et al. [50] and Hoffmann et al. [3] utilized probabilistic graphics models to improve DSRE in MIL framework. Zheng et al. [51] built the inter information of aggregated inter-sentence to enhance DSRE performance. Subsequently, reinforcement learning [20] and adversarial learning [24] were used for improving the quality of labeled data. Recently, Wang et al.[52] proposed a sampling-based method to select samples in the sentence bag to solve the noise problem in DSRE. However, incorrectly labeled data cannot be completely eliminated, there

is much noise in the DS generated data. The current methods reduced the influence of noise by filtering the characteristics of instances and constructing the characteristics of the bag[14, 26].

5.2. Neural Network Methods

An important part of DSRE is the NLP tools to construct the vector features of the instances and the bags. The current mainstream tools are based on neural networks. With the great breakthrough of deep neural networks (DNNs), some researchers applied it for DSRE and obtained a promising result. Zeng et al. [6] firstly proposed a convolutional neural network (CNN) for relation classification. Zhou et al. [8] and Socher et al. [53] proposed to utilize bidirectional long short-term memory (Bi-LSTM) networks to model the sentence with sequential information with all words. With the popularity of GCN, Zhang et al. [12] and Guo et al. [54] leveraged GCN to encode dependency trees of instance to enhance DSRE. Lin et al. [16] employed sentence-level attention to reducing the weight of noisy data and achieves state-of-the-art. Jung et al.[55] proposed a dual supervision framework which utilizes human annotation data to improve the performance of DSRE. Yu et al.[48] formulated DSRE as a hierarchical classification task and propose a novel hierarchical classification framework, which extracts the relation in a top-down manner. In addition, the representation learning method [29] was used to construct entity and relationship embeddings that are independent of the training data, improving the robustness of the model. However, theses methods cannot deal with the case that does not follow the *expressed-at-least-once* assumption. 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.

5.3. Background Information in DSRE

Background information is the information that is related to the entity pairs and sentence bags. It can complement missing semantics in sentence bags. 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 [26, 25]. Other works [28, 27] attempted to combine a human constructed KG into DSRE and propose a joint representation learning framework. In addition, Zeng et al. [15] built an inference chain between two target entities and proposed a path-based neural relation extraction model to encode the relational semantics from both direct sentences and inference chains. Rocktäschel et al. [30] encoded the logical relationships in the knowledge graph further enriches background information. These methods all achieved promising success but suffered from the side effect of introduced noise of background information.

6. Conclusion and Future Work

In this paper, we proposed a novel edge-reasoning graph model to fuse heterogeneous information for DSRE. Firstly, various types of information were converted into vector representations via corresponding encoders. Then, we used a hybrid graph to represent each type of information embedding and the correlations between them. In addition, we constructed an imaginary edge between two entities to represent the target relation, and used edge-reasoning GCN to encode nodes and edge. Moreover, an attention mechanism was used to alleviate the noisy information by incorporating structured triples in a KG. We evaluated our model on NYT-Freebase, NYT-Wikidata and a manually labeled DSRE dataset NYT-H, our model achieved considerable improvement over previous methods. The results demonstrated that incorporating heterogeneous background information is effective, and our ER-HG can reduce the side effect of noisy information. Simultaneously, the test on the bag2sent track showed that the introduction of additional information through ER-HG can also better help the distant supervision model extract the relation from a single instance.

In the future, we will extend our model to sizeable unlabeled text and try to learn more confident features in the unsupervised relation extraction task. For edge reasoning methods, more in-depth research can be carried out in the future, and strategies for the coordinated update of edges and nodes can be designed and measures for the stability of graph coding.

7. Acknowledgement

This work was supported by National Key R&D Program of China (2018YFC 0830200) and National Natural Science Foundation of China Key (U1736204).

References

[1] B. Shi, T. Weninger, Open-world knowledge graph completion, in: Proceedings of AAAI, 2018.

[2] K. Xu, S. Reddy, Y. Feng, S. Huang, D. Zhao, Question answering on freebase via relation extraction and textual evidence, arXiv preprint arXiv:1603.00957 (2016).

[3] R. Hoffmann, C. Zhang, X. Ling, L. S. Zettlemoyer, D. S. Weld, Knowledge-based weak supervision for information extraction of overlapping relations, in: Proceedings of ACL, 2011, pp. 541–550.

[4] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of ACL, 2009, pp. 1003–1011.

[5] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Proceedings of ECML PKDD, 2010, pp. 148–163.

[6] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: Proceedings of COLING, 2014, pp. 2335–2344.

[7] R. Cai, X. Zhang, H. Wang, Bidirectional recurrent convolutional neural network for relation classification, in: Proceedings of ACL, 2016, pp. 756–765.

[8] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: Proceedings of ACL, 2016, pp. 207–212.

[9] Y. Y. Huang, W. Y. Wang, Deep residual learning for weakly-supervised relation extraction, arXiv preprint arXiv:1707.08866 (2017).

[10] S. Vashishth, R. Joshi, S. S. Prayaga, C. Bhattacharyya, P. Talukdar, Reside: Improving distantly-supervised neural relation extraction using side information, arXiv preprint arXiv:1812.04361 (2018).

[11] D. Zeng, K. Liu, Y. Chen, J. Zhao, Distant supervision for relation extraction via piecewise convolutional neural networks, in: Proceedings of EMNLP, 2015, pp. 1753–1762.
[12] Y. Zhang, P. Qi, C. D. Manning, Graph convolution over pruned dependency trees improves relation extraction, arXiv preprint arXiv:1809.10185 (2018).
[13] M. Miwa, M. Bansal, End-to-end relation extraction using lstms on sequences and tree structures, in: Proceedings of ACL, 2016.
[14] D. Zeng, K. Liu, Y. Chen, J. Zhao, Distant supervision for relation extraction via piecewise convolutional neural networks, in: Proceedings of EMNLP, 2015, pp. 1753–1762.
[15] W. Zeng, Y. Lin, Z. Liu, M. Sun, Incorporating relation paths in neural relation extraction, in: Proceedings of EMNLP, 2017, pp. 1768–1777.
[16] Y. Lin, S. Shen, Z. Liu, H. Luan, M. Sun, Neural relation extraction with selective attention over instances, in: Proceedings of ACL, 2016, pp. 2124–2133.
[17] G. Ji, K. Liu, S. He, J. Zhao, Distant supervision for relation extraction with sentence-level attention and entity descriptions, in: Proceedings of AAAI, 2017.
[18] Y. Yuan, L. Liu, S. Tang, Z. Zhang, Y. Zhuang, S. Pu, F. Wu, X. Ren, Cross-relation cross-bag attention for distantly-supervised relation extraction, in: Proceedings of AAAI, 2019, pp. 419–426.
[19] Y. Zhixiu, L. Zhenhua, Distant supervision relation extraction with intra-bag and inter-bag attentions, in: Proceedings of NAACL, 2019.
[20] P. Qin, W. Xu, W. Y. Wang, Robust distant supervision relation extraction via deep reinforcement learning, arXiv preprint arXiv:1805.09927 (2018).
[21] B. Luo, Y. Feng, Z. Wang, Z. Zhu, S. Huang, R. Yan, D. Zhao, Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix, arXiv preprint arXiv:1705.03995 (2017).
[22] B. Liu, H. Gao, G. Qi, S. Duan, T. Wu, M. Wang, Adversarial discriminative denoising for distant supervision relation extraction, in: Proceedings of International Conference on Database Systems for Advanced Applications, 2019, pp. 282–286.
[23] D. Zeng, Y. Dai, F. Li, R. S. Sherratt, J. Wang, Adversarial learning for distant supervised relation extraction, Computers, Materials & Continua 55 (1) (2018) 121–136.
[24] Y. Wu, D. Bamman, S. Russell, Adversarial training for relation extraction, in: Proceedings of EMNLP, 2017, pp. 1778–1783.
[25] Y. Liu, K. Liu, L. Xu, J. Zhao, Exploring fine-grained entity type constraints for distantly supervised relation extraction, in: Proceedings of COLING, ACL, 2014, pp. 2107–2116.
[26] G. Ji, K. Liu, S. He, J. Zhao, Distant supervision for relation extraction with sentence-level attention and entity descriptions, in: Proceedings of AAAI, 2017, pp. 3060–3066.
[27] J. Weston, A. Bordes, O. Yakhnenko, N. Usunier, Connecting language and knowledge bases with embedding models for relation extraction, in: Proceedings of EMNLP, 2013, pp. 1366–1371.
[28] X. Han, Z. Liu, M. Sun, Neural knowledge acquisition via mutual attention between knowledge graph and text, in: Proceedings of AAAI, 2018, pp. 4832–4839.
[29] N. Zhang, S. Deng, Z. Sun, G. Wang, X. Chen, W. Zhang, H. Chen, Long-tail relation extraction via knowledge graph embeddings and graph convolution networks, arXiv preprint arXiv:1903.01306 (2019).
[30] T. Rocktäschel, S. Singh, S. Riedel, Injecting logical background knowledge into embeddings for relation extraction, in: Proceedings of ACL, 2015, pp. 1119–1129.
[31] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, K. Sima'an, Graph convolutional encoders for syntax-aware neural machine translation, arXiv preprint arXiv:1704.04675 (2017).
[32] D. Marcheggiani, I. Titov, Encoding sentences with graph convolutional networks for semantic role labeling, arXiv preprint arXiv:1703.04826 (2017).
[33] H. Zhu, Y. Lin, Z. Liu, J. Fu, T. Chua, M. Sun, Graph neural networks with generated parameters for relation extraction, arXiv preprint arXiv:1902.00756 (2019).
[34] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of SIGMOD, 2008, pp. 1247–1250.
[35] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of AAAI, 2014.
[36] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of NIPS, 2013, pp. 2787–2795.
[37] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of AAAI, 2015.
[38] K. Guu, J. Miller, P. Liang, Traversing knowledge graphs in vector space, in: Proceedings of EMNLP, 2015, pp. 318–327.
[39] A. McCallum, A. Neelakantan, R. Das, D. Belanger, Chains of reasoning over entities, relations, and text using recurrent neural networks, in: Proceedings of EACL, 2017, pp. 132–141.
[40] A. Neelakantan, B. Roth, A. McCallum, Compositional vector space models for knowledge base completion, in: Proceedings of ACL, 2015, pp. 156–166.
[41] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, M. Gamon, Representing text for joint embedding of text and knowledge bases, in: Proceedings of EMNLP, 2015, pp. 1499–1509.
[42] K. Toutanova, V. Lin, W. Yih, H. Poon, C. Quirk, Compositional learning of embeddings for relation paths in knowledge base and text, in: Proceedings of ACL, 2016, pp. 1434–1444.
[43] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of ICLR, 2016.
[44] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: Proceedings of ESWC, 2018, pp. 593–607.
[45] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, S. Liu, Modeling relation paths for representation learning of knowledge bases, in: Proceedings of EMNLP, 2015, pp. 705–714.
[46] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of SIGMOD, 2008, pp. 1247–1250.
[47] T. Zhu, H. Wang, J. Yu, X. Zhou, W. Chen, W. Zhang, M. Zhang, Towards accurate and consistent evaluation: A dataset for distantly-supervised relation extraction, in: COLING, 2020.
[48] E. Yu, W. Han, Y. Tian, Y. Chang, Tohre: A top-down classification strategy with hierarchical bag representation for distantly supervised relation extraction, in: COLING, 2020.
[49] M. Craven, J. Kumlien, Constructing biological knowledge bases by extracting information from text sources, in: Proceedings of ISMB, 1999, pp. 77–86.
[50] M. Surdeanu, J. Tibshirani, R. Nallapati, C. D. Manning, Multi-instance multi-label learning for relation extraction, in: Proceedings of EMNLP-CoNLL, 2012, pp. 455–465.
[51] H. Zheng, Z. Li, S. Wang, Z. Yan, J. Zhou, Aggregating inter-sentence information to enhance relation extraction, in: Proceedings of AAAI, 2016, pp. 3108–3115.
[52] Z. Wang, R. Wen, X. Chen, S.-L. Huang, N. Zhang, Y. Zheng, Finding influential instances for distantly supervised relation extraction, ArXiv abs/2009.09841 (2020).
[53] R. Socher, B. Huval, C. D. Manning, A. Y. Ng, Semantic compositionality through recursive matrix-vector spaces, in: Proceedings of EMNLP-CoNLL, 2012, pp. 1201–1211.
[54] Z. Guo, Y. Zhang, W. Lu, Attention guided graph convolutional networks for relation extraction, arXiv preprint arXiv:1906.07510 (2019).
[55] W. Jung, K. Shim, Dual supervision framework for relation extraction with distant supervision and human annotation, ArXiv abs/2011.11851 (2020).