Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks

Ningyu Zhang[1,2,3] **Shumin Deng**[1,3] **Zhanlin Sun**[1,3] **Guanying Wang**[1,3]
Xi Chen[4] **Wei Zhang**[2,3] **Huajun Chen**[1,3]*
1. College of Computer Science and Technology, Zhejiang University
2. Alibaba Group
3. AZFT†Joint Lab for Knowledge Engine
4. Tencent
{3150105645,231sm,guanying_wgy,huajunsir}@zju.edu.cn
jasonxchen@tencent.com,{ningyu.zny,lantu.zw}@alibaba-inc.com

Abstract

We propose a distance supervised relation extraction approach for long-tailed, imbalanced data which is prevalent in real-world settings. Here, the challenge is to learn accurate "few-shot" models for classes existing at the tail of the class distribution, for which little data is available. Inspired by the rich semantic correlations between classes at the long tail and those at the head, we take advantage of the knowledge from data-rich classes at the head of the distribution to boost the performance of the data-poor classes at the tail. First, we propose to leverage implicit relational knowledge among class labels from knowledge graph embeddings and learn explicit relational knowledge using graph convolution networks. Second, we integrate that relational knowledge into relation extraction model by coarse-to-fine knowledge-aware attention mechanism. We demonstrate our results for a large-scale benchmark dataset which show that our approach significantly outperforms other baselines, especially for long-tail relations.

1 Introduction

Relation extraction (RE) is an important task in information extraction, aiming to extract the relation between two given entities based on their related context. Due to the capability of extracting textual information and benefiting many NLP applications (e.g., information retrieval, dialog generation, and question answering), RE appeals to many researchers. Conventional supervised models have been widely explored in this task (Zelenko et al., 2003; Zeng et al., 2014); however, their performance heavily depends on the scale and quality of training data.

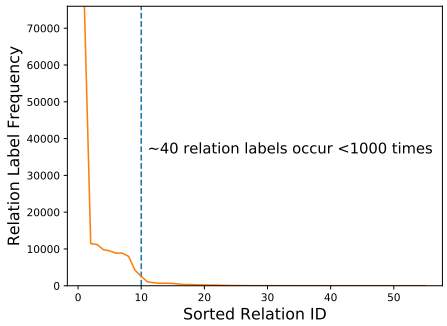


Figure 1: Label frequency distribution of classes without NA in NYT dataset.

To construct large-scale data, (Mintz et al., 2009) proposed a novel distant supervision (DS) mechanism to automatically label training instances by aligning existing knowledge graphs (KGs) with text. DS enables RE models to work on large-scale training corpora and has thus become a primary approach for RE recently (Wu et al., 2017; Feng et al., 2018). Although these DS models achieve promising results on common relations, their performance still degrades dramatically when there are only a few training instances for some relations. Empirically, DS can automatically annotate adequate amounts of training data; however, this data usually only covers a limited part of the relations. Many relations are long-tail and still suffer from data deficiency. Current DS models ignore the problem of long-tail relations, which makes it challenging to extract comprehensive information from plain text.

Long-tail relations are important and cannot be ignored. Nearly 70% of the relations are long-tail in the widely used New York Times (NYT) dataset[1] (Riedel et al., 2010; Lei et al., 2018) as shown in Figure 1. Therefore, it is crucial for mod-

* Corresponding author.
†Alibaba-Zhejiang University Frontier Technology Research Center

[1] http://iesl.cs.umass.edu/riedel/ecml/

els to be able to extract relations with limited numbers of training instances.

Dealing with long tails is very difficult as few training examples are available. Therefore, it is natural to transfer knowledge from rich and semantically similar head classes ta-poor tail classes (Wang et al., 2017). example, the long-tail relation /people/deceased_person/place_of_burial and head relation /people/deceased_person/place_of_death are in the same branch /people/deceased_person/* as shown in Figure 2. They are semantically similar, and it is beneficial to leverage head relational knowledge and transfer it to the long-tail relation, thus enhancing general performance. In other words, long-tail relations of one entity tuple can have class ties with head relations, which can be leveraged to enhance RE for narrowing potential search spaces and reducing uncertainties between relations when predicting unknown relations (Ye et al., 2017). If one pair of entities contains /people/deceased_person/place_of_death, there is a high probability that it will contain /people/deceased_person/place_of_burial. If we can incorporate the relational knowledge between two relations, extracting head relations will provide evidence for the prediction of long-tail relations.

However, there exist two problems: (1) *Learning relational knowledge:* Semantically similar classes may contain more relational information that will boost transfer, whereas irrelevant classes (e.g., /location/location/contains and /people/family/country) usually contain less relational information that may result in negative transfer. (2) *Leveraging relational knowledge:* Integrating relational knowledge to existing RE models is challenging.

To address the problem of learning relational knowledge, as shown in (Lin et al., 2016; Ye et al., 2017), we use class embeddings to represent relation classes and utilize KG embeddings and graph convolution networks (GCNs) to extract implicit and explicit relational knowledge. Specifically, previous studies (Yang et al., 2015) have shown that the embeddings of semantically similar relations are located near each other in the latent space. For instance, the relation /people/person/place_lived and /people/person/nationality are more relevant, whereas the relation /people/person/profession has less correlation with the former two relations. Thus, it

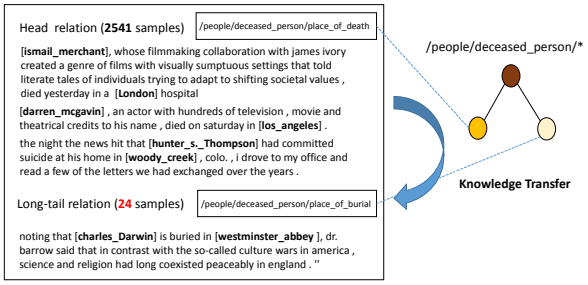


Figure 2: Head and long-tail relations.

is natural to leverage this knowledge from KGs. However, because there are many one-to-multiple relations in KGs, the relevant information for each class may be scattered. In other words, there may not be enough relational signal between classes. Therefore, we utilize GCNs to learn explicit relational knowledge.

To address the problem of leveraging relational knowledge, we first use convolution neural networks (Zeng et al., 2014, 2015) to encode sentences; then introduce coarse-to-fine knowledge-aware attention mechanism for combining relational knowledge with encoded sentences into bag representation vectors. The relational knowledge not only provides more information for relation prediction but also provides a better reference message for the attention module to raise the performance of long-tail classes.

Our experimental results on the NYT dataset show that: (1) our model is effective compared to baselines especially for long-tail relations; (2) leveraging relational knowledge enhances RE and our model is efficient in learning relational knowledge via GCNs.

2 Related Work

Relation Extraction. Supervised RE models (Zelenko et al., 2003; GuoDong et al., 2005; Mooney and Bunescu, 2006) require adequate amounts of annotated data for training which is time-consuming. Hence, (Mintz et al., 2009) proposd DS to automatically label data. DS inevitably accompanies with the wrong labeling problem. To alleviate the noise issue, (Riedel et al., 2010; Hoffmann et al., 2011) proposed multi-instance learning (MIL) mechanisms. Recently, neural models have been widely used for RE; those models can accurately capture textual relations without explicit linguistic analysis (Zeng et al., 2015;

Lin et al., 2016; Zhang et al., 2018a). To further improve the performance, some studies incorporate external information (Zeng et al., 2017; Ji et al., 2017; Han et al., 2018a) and advanced training strategies (Ye et al., 2017; Liu et al., 2017; Huang and Wang, 2017; Feng et al., 2018; Zeng et al., 2018; Wu et al., 2017; Qin et al., 2018). These works mainly adopt DS to make large-scale datasets and reduce the noise caused by DS, regardless of the effect of long-tail relations.

There are only a few studies on long-tail for RE (Gui et al., 2016; Lei et al., 2018; Han et al., 2018b). Of these, (Gui et al., 2016) proposed an explanation-based approach, whereas (Lei et al., 2018) utilized external knowledge (logic rules). These studies treat each relation in isolation, regardless of the rich semantic correlations between the relations. (Han et al., 2018b) proposed a hierarchical attention scheme for RE, especially for long-tail relations. Different from those approaches, we leverage implicit and explicit relational knowledge from KGs and GCNs rather than data-driven learned parameter spaces where similar relations may have distinct parameters, hindering the generalization of long-tail classes.

Knowledge Graph Embedding. Recently, several KG embedding models have been proposed. These methods learn low-dimensional vector representations for entities and relations (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015). TransE (Bordes et al., 2013) is one of the most widely used models, which views relations as translations from a head entity to a tail entity on the same low-dimensional hyperplane. Inspired by the rich knowledge in KGs, recent works (Han et al., 2018a; Wang et al., 2018; Lei et al., 2018) extend DS models under the guidance of KGs. However, these works neglect rich correlations between relations. Relation structure (relational knowledge) has been studied and is quite effective for KG completion (Zhang et al., 2018b). To the best of our knowledge, this is the first effort to consider the relational knowledge of classes (relations) using KGs for RE.

Graph Convolutional Networks. GCNs generalize CNNs beyond two-dimensional and one-dimensional spaces. (Defferrard et al., 2016) developed spectral methods to perform efficient graph convolutions. (Kipf and Welling, 2016) assumed the graph structure is known over input instances and apply GCNs for semi-supervised learning. GCNs were applied to relational data (e.g., link prediction) by (Schlichtkrull et al., 2018). GCNs have also had success in other NLP tasks such as semantic role labeling (Marcheggiani and Titov, 2017), dependency parsing (Strubell and McCallum, 2017), and machine translation (Bastings et al., 2017).

Two GCNs studies share similarities with our work. (1) (Chen et al., 2017) used GCNs on structured label spaces. However, their experiments do not handle long-tail labels and do not incorporate attention but use an average of word vectors to represent each document. (2) (Rios and Kavuluru, 2018) proposed a few-shot and zero-shot text classification method by exploiting structured label spaces with GCNs. However, they used GCNs in the label graph whereas we utilize GCNs in the hierarchy graph of labels.

3 Methodology

In this section, we introduce the overall framework of our approach for RE, starting with the notations.

3.1 Notations

We denote a KG as $\mathcal{G} = \mathcal{E}, \mathcal{R}, \mathcal{F}$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{F}$ indicate the sets of entities, relations and facts respectively. $(h, r, t) \in \mathcal{F}$ indicates that there is a relation $r \in \mathcal{R}$ between $h \in \mathcal{E}$ and $t \in \mathcal{E}$. We follow the MIL setting and split all instances into multiple entity-pair bags $\{\mathcal{S}_{h_1,t_1}, \mathcal{S}_{h_2,t_2}, ...\}$. Each bag $\mathcal{S}_{h_i,t_i}$ contains multiple instances $\{s_1, s_2, ...\}$ mentioning both entities h_i and t_i. Each instance s in these bags is denoted as a word sequence $s = \{w_1, w_2, ...\}$.

3.2 Framework

Our model consists of three parts as shown in Figure 3:

Instance Encoder. Given an instance and its mentioned entity pair, we employ neural networks to encode the instance semantics into a vector. In this study, we implement the instance encoder with convolutional neural networks (CNNs) given both model performance and time efficiency.

Relational Knowledge Learning. Given pre-trained KG embeddings (e.g., TransE (Bordes et al., 2013)) as implicit relational knowledge, we employ GCNs to learn explicit relational knowledge. By assimilating generic message-passing inference algorithms with neural-network counterpart, we can learn better embeddings for Knowl-

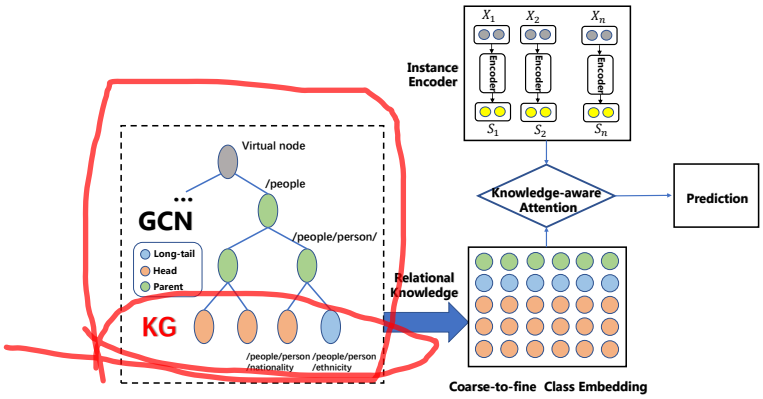


Figure 3: Architecture of our proposed model.

edge Relation. We concatenate the outputs of GCNs and the pretrained KG embeddings to form the final class embeddings.

Knowledge-aware Attention. Under the guidance of final class embeddings, knowledge-aware attention is aimed to select the most informative instance exactly matching relevant relation.

3.3 Instance Encoder

Given an instance $s = \{w_1, ..., w_n\}$ mentioning two entities, we encode the raw instance into a continuous low-dimensional vector x, which consists of an embedding layer and an encoding layer.

Embedding Layer. The embedding layer is used to map discrete words in the instance into continuous input embeddings. Given an instance s, we map each word w_i in the instance to a real-valued pretrained Skip-Gram (Mikolov et al., 2013) embedding $w_i \in \mathbb{R}^{d_w}$. We adopt position embeddings following (Zeng et al., 2014). For each word w_i, we embed its relative distances to the two entities into two d_p-dimensional vectors. We then concatenate the word embeddings and position embeddings to achieve the final input embeddings for each word and gather all the input embeddings in the instance. We thus obtain an embedding sequence ready for the encoding layer.

Encoding Layer. The encoding layer aims to compose the input embeddings of a given instance into its corresponding instance embedding. In this study, we choose two convolutional neural architectures, CNN (Zeng et al., 2014) and PCNN (Zeng et al., 2015) to encode input embeddings into instance embeddings. Other neural architectures such as recurrent neural networks (Zhang and Wang, 2015) can also be used as sentence encoders. Because previous works show that both convolutional and recurrent architectures can achieve comparable state-of-the-art performance, we select convolutional architectures in this study. Note that, our model is independent of the encoder choices, and can, therefore, be easily adapted to fit other encoder architectures.

3.4 Relational Knowledge Learning through KG Embeddings and GCNs.

Given pretrained KG embeddings and predefined class (relation) hierarchies[2], we first leverage the implicit relational knowledge from KGs and initialize the hierarchy label graph; then we apply two layer GCNs to learn explicit fine-grained relational knowledge from the label space.

Hierarchy Label Graph Construction. Given a relation set $\mathcal{R}$ of a KG $\mathcal{G}$ (e.g., Freebase), which consists of base-level relations (e.g., /people/person/ethnicity), we can generate the corresponding higher-level relation set $\mathcal{R}^H$. The relations in a high-level set (e.g., people) are more general and common; they usually contain several sub-relations in the base-level set. The relation hierarchies are tree-structured, and the generation process can be done recursively. We use a virtual father node to construct the highest level associations between relations as shown in Figure 3. In practice, we start from $\mathcal{R}^0 = R$ which is the set of all relations we focus on for RE, and the generation process is performed $L - 1$ times to get the hierarchical relation sets $\{\mathcal{R}^0, \mathcal{R}^1, ..., \mathcal{R}^L\}$, where $\mathcal{R}^L$ is the virtual father node. Each node has a specific type $t \in \{0, 1, ..., L\}$ to identify its layer hierarchies. For example, as shown in Fig-

[2]For datasets without predefined relation hierarchies, hierarchy clustering (Johnson, 1967) or K-means can construct relation hierarchies (Zhang et al., 2018b); details can be found in supplementary materials.

ure 3, node /people/person/ethnicity has a specific type 0 to indicate it is in the bottom layer of the graph. The vectors of each node in the bottom layer are initialized through pretrained TransE (Bordes et al., 2013) KG embeddings. Other KG embeddings such as TransR (Lin et al., 2015) can also be adopted. Their parent nodes are initialized by averaging all children vectors. For example, the node vector of /people/person/ is initialized by averaging all the nodes under the branch of /people/person/* (all child nodes).

GCN Output Layer. Due to one-to-multiple relations and incompleteness in KGs, the implicit relevant information obtained by KG embeddings for each label is not enough. Therefore, we apply GCNs to learn explicit relational knowledge among labels. We take advantage of the structured knowledge over our label space using a two-layer GCNs. Starting with the pretrained relation embedding $v_i^{implicit} \in \mathbb{R}^d$ from KGs, we combine the label vectors of the children and parents for the i-th label to form,

$$v_i^1 = f(W^1 v_i + \sum_{j \in \mathcal{N}_p} \frac{W_p^1 v_j}{|\mathcal{N}_p|} + \sum_{j \in \mathcal{N}_c} \frac{W_c^1 v_j}{|\mathcal{N}_c|} + b_g^1) \quad (1)$$

where $W^1 \in \mathbb{R}^{q*d}$, $W_p^1 \in \mathbb{R}^{q*d}$, $W_c^1 \in \mathbb{R}^{q*d}$, $b_g^1 \in \mathbb{R}^q$, f is the rectified linear unit (Nair and Hinton, 2010) function, and $\mathcal{N}_c$ ($\mathcal{N}_p$) is the index set of the i-th labels children (parents). We use different parameters to distinguish each edge type where parent edges represent all edges from high level labels and child edges represent all edges from low level labels. The second layer follows the same formulation as the first layer and outputs $v_i^{explicit}$. Finally, we concatenate both pretrained $v_i^{implicit}$ with GCNs node vector $v_i^{explicit}$ to form hierarchy class embeddings,

$$q_r = v_i^{implicit} || v_i^{explicit} \quad (2)$$

where $q_r \in \mathbb{R}^{d+q}$.

3.5 Knowledge-aware Attention

Traditionally, the output layer of PCNN/CNN would learn label specific parameters optimized by a cross-entropy loss. However, the label specific parameters spaces are unique to each relation, matrices associated with the long-tails can only be exposed to very few facts during training, resulting in poor generalization. Instead, our method attempts to match sentence vectors to their corresponding class embeddings rather than learning label specific attention parameters. In essence, this becomes a retrieval problem. Relevant information from class embeddings contains useful relational knowledge for long-tails among labels.

Practically, given the entity pair (h, t) and its bag of instances $S_{h,t} = \{s_1, s_2, ..., s_m\}$, we achieve the instance embeddings $\{s_1, s_2, ..., s_m\}$ using the sentence encoder. We group the class embeddings according to their type (i.e., according to their layers in the hierarchy label graph), e.g., $q_{r^i}, i \in \{0, 1, ..., L\}$. We adopt $q_{r^i}, i \neq L$ (layer L is the virtual father node) as layer-wise attention query vector. Then, we apply coarse-to-fine knowledge-aware attention to them to obtain the textual relation representation $r_{h,t}$. For a relation r, we construct its hierarchical chain of parent relations $(r^0, ..., r^{L-1})$ using the hierarchy label graph, where r^{i-1} is the sub-relation of r^i. We propose the following formulas to compute the attention weight (similarity or relatedness) between each instances feature vector s_k and q_{r^i},

$$e_k = W_s(tanh[s_k; q_{r^i}]) + b_s$$
$$\alpha_k^i = \frac{exp(e_k)}{\sum_{j=1}^m exp(e_j)} \quad (3)$$

where $[x_1; x_2]$ denotes the vertical concatenation of x_1 and x_2, W_s is the weight matrix, and b_s is the bias. We compute attention operations on each layer of hierarchy label graph to obtain corresponding textual relation representations,

$$r_{h,t}^i = ATT(q_{r^i}, \{s_1, s_2, ..., s_m\}) \quad (4)$$

Then we need to combine the relation representations on different layers. Direct concatenation of all the representations is a straightforward choice. However, different layers have different contributions for different tuples. For example, the relation /location/br_state/ has only one sub-relation /location/br_state/capital, which indicates that it is more important. In other words, if the sentence has high attention weights on relation /location/br_state/, it has a very high probability to have relation /location/br_state/capital. Hence, we use an attention mechanism to emphasize the layers,

$$g_i = W_g tanh(r_{h,t})$$
$$\beta_i = \frac{exp(g_i)}{\sum_{j=0}^{L-1} exp(g_j)} \quad (5)$$

where W_g is a weight matrix, $r_{h,t}$ is referred to as a query-based function that scores how well the input textual relation representations and the predict relation r match. The textual relation representations in each layer are computed as,

$$r^i_{h,t} = \beta_i r^i_{h,t} \tag{6}$$

We simply concatenate the textual relation representations on different layers as the final representation,

$$r_{h,r} = Concat(r^0_{h,t}, .., r^{L-1}_{h,t}) \tag{7}$$

The representation $r_{h,t}$ will be finally fed to compute the conditional probability $\mathcal{P}(r|h,t,\mathcal{S}_{h,t})$,

$$\mathcal{P}(r|h,t,\mathcal{S}_{h,t}) = \frac{exp(o_r)}{\sum_{\tilde{r}\in R} exp(o_{\tilde{r}})} \tag{8}$$

where o is the score of all relations defined as,

$$o = Mr_{h,t} \tag{9}$$

where M is the representation matrix to calculate the relation scores. Note that, attention weight q_{r^i} is obtained from the outputs of GCNs and pretrained KG embeddings, which can provide more informative parameters than data-driven learned parameters, especially for long-tails.

4 Experiments

4.1 Datasets and Evaluation

We evaluate our models on the NYT dataset developed by (Riedel et al., 2010), which has been widely used in recent studies (Lin et al., 2016; Liu et al., 2017; Wu et al., 2017; Feng et al., 2018). The dataset has 53 relations including the NA relation, which indicates that the relations of instances are not available. The training set has 522611 sentences, 281270 entity pairs, and 18252 relational facts. In the test set, there are 172448 sentences, 96678 entity pairs, and 1950 relational facts. In both training and test set, we truncate sentences with more than 120 words into 120 words.

We evaluate all models in the held-out evaluation. It evaluates models by comparing the relational facts discovered from the test articles with those in Freebase and provides an approximate measure of precision without human evaluation. For evaluation, we draw precision-recall curves for all models. To further verify the effect of our model for long-tails, we follow previous studies (Han et al., 2018b) to report the Precision@N results. The dataset and baseline code can be found on Github [3].

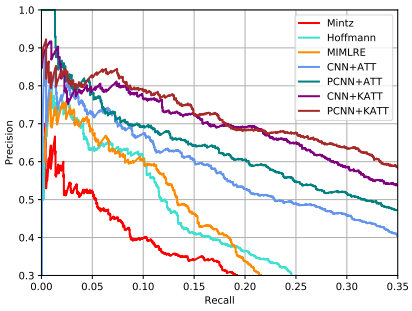


Figure 4: Precision-recall curves for the proposed model and various baseline models.

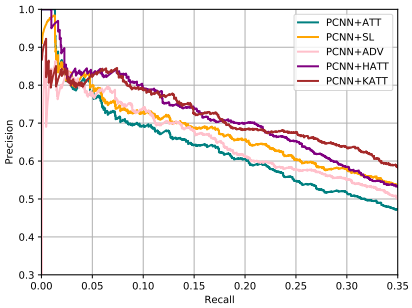


Figure 5: Precision-recall curves for the proposed model and various attention-based neural models.

4.2 Parameter Settings[4]

To fairly compare the results of our models with those baselines, we also set most of the experimental parameters by following (Lin et al., 2016). We apply dropout on the output layers of our models to prevent overfitting. We also pretrain the sentence encoder of PCNN before training our model.

4.3 Overall Evaluation Results

To evaluate the performance of our proposed model, we compare the precision-recall curves of our model with various previous RE models. The evaluation results are shown in Figure 4 and Figure 5. We report the results of neural architectures including CNN and PCNN with various attention based methods: **+KATT** denotes our approach, **+HATT** is the hierarchical atten-

[3] https://github.com/thunlp/OpenNRE

[4] Details of hyper-parameters settings and evaluation of different instances can be found in supplementary materials

Training Instances		<100			<200		
Hits@K (Macro)		10	15	20	10	15	20
CNN	+ATT	<5.0	<5.0	18.5	<5.0	16.2	33.3
	+HATT	5.6	31.5	57.4	22.7	43.9	65.1
	+KATT	**9.1**	**41.3**	**58.5**	**23.3**	**44.1**	**65.4**
PCNN	+ATT	<5.0	7.4	40.7	17.2	24.2	51.5
	+HATT	29.6	51.9	61.1	41.4	60.6	68.2
	+KATT	**35.3**	**62.4**	**65.1**	**43.2**	**61.3**	**69.2**

Table 1: Accuracy (%) of Hits@K on relations with training instances fewer than 100/200.

Training Instances	<100			<200		
Hits@K (Macro)	10	15	20	10	15	20
+KATT	**35.3**	**62.4**	**65.1**	**43.2**	**61.3**	**69.2**
w/o hier	34.2	62.1	65.1	42.5	60.2	68.1
w/o GCNs	30.5	61.9	63.1	39.5	58.4	66.1
Word2vec	30.2	62.0	62.5	39.6	57.5	65.8
w/o KG	30.0	61.0	61.3	39.5	56.5	62.5

Table 2: Results of ablation study with PCNN.

tion method (Han et al., 2018b), **+ATT** is the plain selective attention method over instances (Lin et al., 2016), **+ATT+ADV** is the denoising attention method by adding a small adversarial perturbation to instance embeddings (Wu et al., 2017), and **+ATT+SL** is the attention-based model using soft-labeling method to mitigate the side effect of the wrong labeling problem at entity-pair level (Liu et al., 2017). We also compare our method with feature-based models, including **Mintz** (Mintz et al., 2009), **MultiR** (Hoffmann et al., 2011) and **MIML** (Surdeanu et al., 2012).

As shown in both figures, our approach achieves the best results among all attention-based models. Even when compared with PCNN+HATT, PCNN+ATT+ADV, and PCNN+ATT+SL, which adopt sophisticated denoising schemes and extra information, our model is still more advantageous. This indicates that our method can take advantage of the rich correlations between relations through KGs and GCNs, which improve the performance. We believe the performance of our model can be further improved by adopting additional mechanisms like adversarial training, and reinforcement learning, which will be part of our future work.

4.4 Evaluation Results for Long-tail Relations

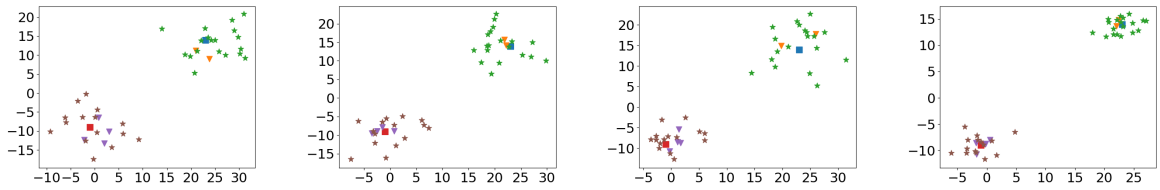
To further demonstrate the improvements in performance for long-tail relations, following the study by (Han et al., 2018b) we extract a subset of the test dataset in which all the relations have fewer than 100/200 training instances. We employ the Hits@K metric for evaluation. For each entity pair, the evaluation requires its corresponding golden relation in the first K candidate relations recommended by the models. Because it is difficult for the existing models to extract long-tail relations, we select K from $\{10,15,20\}$. We report the macro average Hits@K accuracies for these subsets because the micro-average score generally overlooks the influences of long-tails. From the results shown in Table 1, we observe that for both CNN and PCNN models, our model outperforms the plain attention model and the HATT model. Although our KATT method has achieved better results for long-tail relations as compared to both plain ATT method and HATT method, the results of all these methods are still far from satisfactory. This indicates that distantly supervised RE models still suffer from the long-tail relation problem, which may require additional schemes and extra information to solve this problem in the future.

4.5 Ablation Study

To analyze the contributions and effects of different technologies in our approach, we perform ablation tests. **+KATT** is our method; **w/o hier** is the method without coarse-to-fine attention (only utilizes bottom node embeddings of the hierarchy label graph), which implies no knowledge transfer from its higher level classes; **w/o GCN** is the method without GCNs, which implies no explicit relational knowledge; **Word2vec** is the method in which the node is initialized with pretrained Skip-Gram (Mikolov et al., 2013) embeddings; and **w/o KG** is the method in which the node is initialized with random embeddings, which implies no prior relational knowledge from KGs. From the evaluation results in Table 2, we observe that the performance slightly degraded without coarse-to-fine attention, which proves that knowledge transfer from the higher node is useful. We also noticed that the performance slightly degraded without KG or using word embeddings, and the performance significantly degraded when we removed GCNs. This is reasonable because GCNs can learn more explicit correlations between relation labels, which boost the performance for long-tail relations.

4.6 Case Study

We give some examples to show how our method affects the selection of sentences. In Table 3, we display the sentence's attention



(a) HATT (b) +KG (c) +GCNs (d) KATT

Figure 6: T-SNE visualizations of class embeddings. Cluster in the upper right is the relation /location/*/* and cluster in the bottom left is the relation /people/*/*). The square, triangle, and star refer to the high (/location) middle (/location/location/) and base (/location/location/contains) level relations, respectively.

/people/deceased_person/place_of_burial	**HATT**	**KATT**
richard_wagner had his **bayreuth**, with its festspielhaus specially designed to accommodate his music dramas.	0.21	0.07
wotan and alberich are **richard_wagner**; and the rheingold and valhalla are wagner's real-life grail, the opera house in **bayreuth**.	0.15	0.13
/location/br_state/capital	**HATT**	**KATT**
there's violence everywhere, said ms. mesquita, who, like her friend, lives in **belo_horizonte**, the capital of **minas_gerais** state	0.47	0.51
all the research indicates that we are certain to find more gas in th amazon, eduardo braga, the governor of **amazonas**, said in an interview in **manaus**	0.46	0.45

Table 3: Example sentences for case study.

score in the lowest level[5]. Both the relation /people/deceased_person/place_of_burial (24 instances) and /location/br_state/capital (4 instances) are long-tail relations. On one hand, relation /people/deceased_person/place_of_burial has semantically similar data-rich relation such as /people/deceased_person/place_of_death. We observe that HATT erroneously assigns high attention to the incorrect sentence whereas KATT successfully assigns the right attention weights, which demonstrates the efficacy of knowledge transfer from semantically similar relations (Both HATT and KATT methods can take advantage of knowledge transfer of high-level relations.). On the other hand, the relation /location/br_state/capital does not have semantically similar relations. However, we notice that KATT still successfully assigns the right attention weights, which demonstrates the efficacy of knowledge transfer from high-level relations using coarse-to-fine knowledge-aware attention.

[5]Both HATT and KATT methods can successfully select the correct sentence at the higher-level; details can be found in supplementary materials.

4.7 Visualizations of Class Embeddings

We visualize the class embeddings via t-SNE (Maaten and Hinton, 2008) to further show how GCNs and KG embeddings can help RE for long-tail relations. We observe that (1) Figure 6(a) and 6(d) show that semantically similar class embeddings are closer with GCNs and pretrained KG embeddings, which help select long-tail instances. (2) Figure 6(b) and 6(c) show that KG embeddings and GCNs have different contributions for different relations to learn relational knowledge between classes. For example, /location/location/contain has a sparse hierarchy structure, which leads to inefficient learning for GCNs; therefore, the relative distance changes only slightly, which reveals the necessity of implicit relational knowledge from KGs. (3) Figure 6(d) shows that there are still some semantically similar class embeddings located far away, which may degrade the performance for long-tails. This may be caused by either sparsity in the hierarchy label graph or equal treatment for nodes with the same parent in GCNs, which is not a reasonable hypothesis. We will address this by integrating more information such as relation descriptions or combing logic reasoning as a part of future work.

5 Conclusion and Future Work

In this paper, we take advantage of the knowledge from data-rich classes at the head of distribution to boost the performance of the data-poor classes at the tail. As compared to previous methods, our approach provides fine-grained relational knowledge among classes using KG and GCNs, which is quite effective and encoder-agnostic.

In the future, we plan to explore the following directions: (1) We may combine our method

with recent denoising methods to further improve performance. (2) We may combine rule mining and reasoning technologies to learn better class embeddings to boost performance. (3) It will be promising to apply our method to zero-shot RE and further adapt to other NLP scenarios.

Acknowledgments

We want to express gratitude to the anonymous reviewers for their hard work and kind comments, which will further improve our work in the future. This work is funded by NSFC91846204/61473260, national key research program YS2018YFB140004, Alibaba CangJingGe (Knowledge Engine) Research Plan and Natural Science Foundation of Zhejiang Province of China (LQ19F030001).

References

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. *arXiv preprint arXiv:1704.04675*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pages 2787–2795.

Meihao Chen, Zhuoru Lin, and Kyunghyun Cho. 2017. Graph convolutional networks for classification with a structured label space. *arXiv preprint arXiv:1710.04908*.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*.

Yaocheng Gui, Qian Liu, Man Zhu, and Zhiqiang Gao. 2016. Exploring long tail data in distantly supervised relation extraction. In *Natural Language Understanding and Intelligent Applications*, pages 514–522. Springer.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018a. Neural knowledge acquisition via mutual attention between knowledge graph and text.

Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. 2018b. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2245.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL*, pages 541–550. Association for Computational Linguistics.

Yi Yao Huang and William Yang Wang. 2017. Deep residual learning for weakly-supervised relation extraction. *arXiv preprint arXiv:1707.08866*.

Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao, et al. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of AAAI*, pages 3060–3066.

Stephen C Johnson. 1967. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kai Lei, Daoyuan Chen, Yaliang Li, Nan Du, Min Yang, Wei Fan, and Ying Shen. 2018. Cooperative denoising for distantly supervised relation extraction. In *Proceedings of Coling*, pages 426–436.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, pages 2181–2187.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, volume 1, pages 2124–2133.

Tianyu Liu, Kexiang Wang, Baobao Chang, and Zhifang Sui. 2017. A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1790–1795.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Raymond J Mooney and Razvan C Bunescu. 2006. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of ICML*, pages 807–814.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Dsgan: Generative adversarial training for distant supervision relation extraction. In *Proceedings of ACL*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of EMNLP*, pages 3132–3142.

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

Emma Strubell and Andrew McCallum. 2017. Dependency parsing with dilated iterated graph cnns. *arXiv preprint arXiv:1705.00403*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*, pages 455–465. Association for Computational Linguistics.

Guanying Wang, Wen Zhang, Ruoxu Wang, Yalin Zhou, Xi Chen, Wei Zhang, Hai Zhu, and Huajun Chen. 2018. Label-free distant supervision for relation extraction via knowledge graph embedding. In *Proceedings of EMNLP*, pages 2246–2255.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. In *Proceedings of NIPS*, pages 7029–7039.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, volume 14, pages 1112–1119.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of EMNLP*, pages 1778–1783.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. *Proceedings of ICLR*.

Hai Ye, Wenhan Chao, Zhunchen Luo, and Zhoujun Li. 2017. Jointly extracting relations with class ties via effective deep ranking. In *Proceedings of ACL*, volume 1, pages 1810–1820.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating relation paths in neural relation extraction. In *Proceddings of EMNLP*.

Xiangrong Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Large scaled relation extraction with reinforcement learning. In *Processings of AAAI*, volume 2, page 3.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Ningyu Zhang, Shumin Deng, Zhanling Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018a. Attention-based capsule networks with dynamic routing for relation extraction. In *Proceedings of EMNLP*.

Zhao Zhang, Fuzhen Zhuang, Meng Qu, Fen Lin, and Qing He. 2018b. Knowledge graph embedding with hierarchical relation structure. In *Proceedings of EMNLP*, pages 3198–3207.