Hierarchical Relation Extraction with Coarse-to-Fine Grained Attention

Xu Han[1,2,3*], **Pengfei Yu**[2,3,4*], **Zhiyuan Liu**[1,2,3†], **Maosong Sun**[1,2,3], **Peng Li**[5]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Institute for Artificial Intelligence, Tsinghua University, Beijing, China
[3]State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
[4]Department of Electronic Engineering, Tsinghua University, Beijing, China
[5]Pattern Recognition Center, WeChat, Tencent, China

Abstract

Distantly supervised relation extraction employs existing knowledge graphs to automatically collect training data. While distant supervision is effective to scale relation extraction up to large-scale corpora, it inevitably suffers from the wrong labeling problem. Many efforts have been devoted to identifying valid instances from noisy data. However, most existing methods handle each relation in isolation, regardless of rich semantic correlations located in relation hierarchies. In this paper, we aim to incorporate the hierarchical information of relations for distantly supervised relation extraction and propose a novel hierarchical attention scheme. The multiple layers of our hierarchical attention scheme provide coarse-to-fine granularity to better identify valid instances, which is especially effective for extracting those long-tail relations. The experimental results on a large-scale benchmark dataset demonstrate that our models are capable of modeling the hierarchical information of relations and significantly outperform other baselines. The source code of this paper can be obtained from https://github.com/thunlp/HNRE.

1 Introduction

Relation extraction (RE) aims to predict relational facts from plain text. Conventional supervised RE models (Zelenko et al., 2003; Mooney and Bunescu, 2006) usually suffer from the lack of high-quality training data, because manual labeling of training data is time-consuming and human-intensive. Mintz et al. (2009) propose distant supervision to automatically label training instances by aligning existing knowledge graphs (KGs) and text: For an entity pair in KGs, those sentences containing both the entities will be labeled with the corresponding relation of the entity pair in KGs. RE relies on distant supervision to scale up to large-scale training corpora. However, this automatic mechanism is inevitably accompanied by the wrong labeling problem, because not all sentences containing two entities can exactly express their relations in KGs, e.g., we may mistakenly label "*Bill Gates* retired from *Microsoft*" with the relation `business/company/founders`.

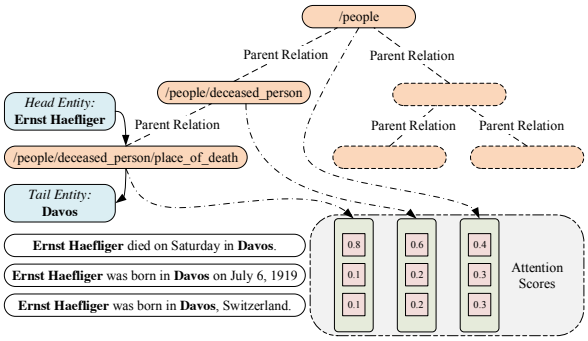

Figure 1: An example of hierarchical relation extraction.

To alleviate the wrong labeling problem, many efforts (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Zeng et al., 2015) have been devoted to identifying valid instances from noisy data, especially the recent state-of-the-art attention-based methods (Lin et al., 2016; Ji et al., 2017; Liu et al., 2017; Wu et al., 2017). Nevertheless, each relation is handled in isolation in most existing methods. For each relation, there is often a separate model (e.g. neural attention scheme) to select relation-related informative instances from noisy data, regardless of rich semantic correlations among relations, typically located in the form of relation hierarchies.

We take the KG Freebase (Bollacker et al., 2008) as an example, in which relations are labeled as hierarchical structures. For example, the

* indicates equal contribution
† Corresponding author: Z.Liu(liuzy@tsinghua.edu.cn)

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2236–2245
Brussels, Belgium, October 31 - November 4, 2018.

relation /location/province/capital in Freebase indicates the relation between a province and its capital. It is labeled under the location branch. Under this branch, there are some other relations /location/location/contains and /location/country/capital, which are closely correlated to each other. The rich correlations among relations are well revealed by these relation hierarchies. In fact, McCallum et al. (1998) take advantage of hierarchies of classes to improve classification models and inspire many later models (Rousu et al., 2005; Weinberger and Chapelle, 2009). Furthermore, the hierarchical information of entities in KGs has also been utilized and demonstrated to be effective for model enhancement (Hu et al., 2015; Xie et al., 2016).

To take advantage of the rich correlated information among relations, we propose a novel hierarchical attention scheme via utilizing the relation hierarchies, rather than directly utilizing hierarchical information as features for models. Similar to the conventional attention-based method, our method also computes an attention score for each instance according to its significance of expressing the corresponding relation. The key difference is that, as illustrated in Figure 1, our hierarchical attention scheme follows the relation hierarchies to compute scores for those instances containing the same entity pair on the each layer of the hierarchies.

The hierarchical attention scheme provides coarse-to-fine granularity for identifying valid instances. The attention on the bottom layer can capture more specific features of the relation, which has a comparable ability of fine-grained instance selection like conventional attention-based methods. The attention on the top layer can capture the common features shared by several related sub-relations, which provides coarse-grained instance selection. Since there are more sufficient data for training the top-layer attention, the whole hierarchical attention scheme can enhance RE models for solving those long-tail relations.

We conduct experiments on a large-scale benchmark dataset for RE in this paper. The experimental results show that the proposed coarse-to-fine grained attention scheme based on relation hierarchies significantly outperforms other baseline methods, even as compared to the recent state-of-the-art attention-based models, especially for those long-tail relations.

2 Related Works

Supervised models (Zelenko et al., 2003; Zhou et al., 2005; Mooney and Bunescu, 2006) for RE require adequate amounts of annotated data for their training. It is time-consuming to manually label large-scale training data. Hence, Mintz et al. (2009) propose distant supervision to automatically label data. Distant supervision inevitably accompanies with the wrong labeling problem. To alleviate the noise issue caused by distant supervision, Riedel et al. (2010) and Hoffmann et al. (2011) propose multi-instance learning (MIL) mechanisms. Riedel et al. (2013) propose universal schema to transmit information between relations of KGs and textual patterns to enhance extraction performance.

These early RE methods mainly extract semantic features using NLP tools to build relation classifiers. Recently, neural models have been widely used for RE. These neural models can accurately capture textual relations without explicit linguistic analysis (Zeng et al., 2014; Xu et al., 2015; Santos et al., 2015; Zhang and Wang, 2015; Verga et al., 2016; Verga and McCallum, 2016). Zeng et al. (2015) employ the MIL scheme by selecting one most valid instance for distantly supervised neural relation extraction (NRE), whose denoising capability is far from satisfactory because most informative instances are neglected. Lin et al. (2016) and Zhang et al. (2017) propose neural attention schemes to select those informative instances. To further improve the attention performance, some works incorporate knowledge information (Zeng et al., 2017; Ji et al., 2017; Han et al., 2018) and advanced training strategies (Liu et al., 2017; Huang and Wang, 2017). More sophisticated mechanisms, such as reinforcement learning (Feng et al., 2018; Zeng et al., 2018) and adversarial training (Wu et al., 2017), have also been adapted for RE recently.

However, most existing works model each relation in isolation to identify informative instances, neglecting rich correlations among relations, especially the hierarchical information of those relations. Hierarchical information is widely applied for model enhancement, especially for classification models (McCallum et al., 1998; Rousu et al., 2005; Weinberger and Chapelle, 2009; Zhao et al., 2011; Bi and Kwok, 2011; Zhou et al., 2011; Verma et al., 2012). Many efforts are also devoted to utilizing hierarchical information in KGs.

Leacock and Chodorow (1998) and Ponzetto and Strube (2007) adopt hierarchical information derived from KGs to construct concept relatedness. Morin and Bengio (2005) propose a neural language model by utilizing hierarchical information in WordNet. Further, Hu et al. (2015) learn entity representations by considering the whole entity hierarchies of Wikipedia and inspire many works (Krompaß et al., 2015; Xie et al., 2016) to utilize hierarchical type structures to help the representation learning of KGs.

Different from the recent hierarchical models that mainly focus on entity hierarchies and directly utilize hierarchical information as simple features, we incorporate relation hierarchies to build a hierarchical attention scheme with coarse-to-fine granularity to enhance RE performance. As compared with the existing models for RE, our models could take advantage of relation correlations to better identify informative instances, especially for those long-tail relations, by transferring the knowledge from their related relations of high-frequency.

3 Methodology

In this section, we introduce the overall framework of our hierarchical attention for RE, starting with notations and definitions.

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 the entire instances into multiple entity-pair bags $\{\mathcal{S}_{h_1,t_1}, \mathcal{S}_{h_2,t_2}, \ldots\}$. Each bag $\mathcal{S}_{h_i,t_i}$ contains multiple instances $\{s_1, s_2, \ldots\}$ mentioning both the entities h_i and t_i. The distant supervision mechanism will label the bag with the corresponding relation of the mentioned entity pair. Each instance s in these bags is denoted as a word sequence $s = \{w_1, w_2, \ldots\}$.

3.2 Framework

Given an entity pair (h, t) and its entity-pair bag $\mathcal{S}_{h,t}$, we adopt our models to measure the probability of each relation $r \in \mathcal{R}$ holding between the pair. As shown in Figure 2, the overall framework of our models includes a sentence encoder and a coarse-to-fine grained hierarchical attention. The sentence encoder adopts several convolutional neural networks to represent sentence semantics with embeddings, and the hierarchical attention is used to select the most informative instances to exactly express their relations.

For each instance $s_i \in \mathcal{S}_{h,t}$, we use the sentence encoder to represent its semantic information as an embedding $\mathbf{s}_i$. The details of the sentence encoder will be introduced in Section 3.3. Since not all instances in the bag $\mathcal{S}_{h,t}$ are positive to express the relation between h and t, we apply the hierarchical attention to compute an instance weight α_i for each instance s_i. The details of the hierarchical attention will be introduced in Section 3.4. We build the global textual relation representation $\mathbf{r}_{h,t}$ with the weighted sum of instance output embeddings,

$$\mathbf{r}_{h,t} = \sum_{i=1}^{m} \alpha_i \mathbf{s}_i, \quad s_1, \ldots, s_m \in \mathcal{S}_{h,t}. \tag{1}$$

Here α_i is the instance weight for the ith instance output embedding $\mathbf{s}_i$. By taking $\mathbf{r}_{h,t}$ as the textual relation representation of the entity pair (h, t), we estimate its probability over each relation $r \in \mathcal{R}$, i.e., whether there is a specific relation r between h and t. We define the conditional probability $P(r|h, t, \mathcal{S}_{h,t})$,

$$P(r|h, t, \mathcal{S}_{h,t}) = \frac{\exp(\mathbf{o}_r)}{\sum_{\tilde{r} \in \mathcal{R}} \exp(\mathbf{o}_{\tilde{r}})}, \tag{2}$$

where $\mathbf{o}$ is the scores of all relations, which is defined as follows,

$$\mathbf{o} = \mathbf{M}\mathbf{r}_{h,t}, \tag{3}$$

where $\mathbf{M}$ is the representation matrix to calculate the relation scores.

3.3 Sentence Encoder

Given an instance s containing two entities, we apply several neural architectures to encode the instance into its corresponding embeddings $\mathbf{s}$.

Input Layer

The input layer of the sentence encoder aims to embed both semantic information and positional information of words into their input embeddings.

Word Embedding is proposed by Hinton (1986), which aims to transform words into distributed representations to capture syntactic and semantic meanings of words. Given a sentence s consisting of multiple words $s = \{w_1, \ldots, w_n\}$, we adopt Skip-Gram (Mikolov et al., 2013) to

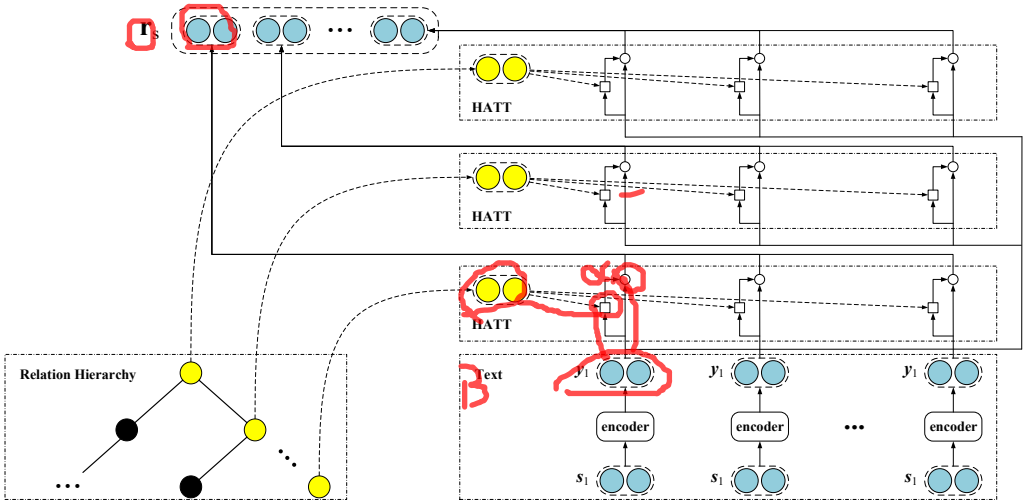

Figure 2: The architecture of hierarchical attention model.

compute all k_w-dimensional word embeddings $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$.

Position Embedding is proposed by Zeng et al. (2014). Position embedding is used to embed the relative distances of each word to the two entities into two k_p-dimensional vectors. By concatenating the distance embeddings for the current word w_i to the both head and tail entities, we get a unified position embedding $\mathbf{p}_i \in \mathbb{R}^{k_p \times 2}$.

For each word w_i, we concatenate its word embedding $\mathbf{w}_i$ and position embedding $\mathbf{p}_i$ to build its input embedding $\mathbf{x}_i \in \mathbb{R}^{k_i} (k_i = k_w + k_p \times 2)$.

Encoding Layer

The encoding layer aims to compose the input embeddings of the given instance into its corresponding instance embedding. In this paper, 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 architectures (Zhang and Wang, 2015) can also be used as sentence encoders. Since previous works show that both convolutional and recurrent architectures can achieve comparable state-of-the-art performance, we simply select convolutional architectures in this paper. Note that, our hierarchical attention scheme is designed independently to the encoder choices, hence it can be easily adapted to fit other encoder architectures.

CNN slides a convolution kernel with the window size m over the input sequence $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ to get the k_h-dimensional hidden embeddings.

$$\mathbf{h}_i = \text{CNN}\big(\mathbf{x}_{i-\frac{m-1}{2}}, \ldots, \mathbf{x}_{i+\frac{m-1}{2}}\big). \quad (4)$$

A max-pooling is then applied over these hidden embeddings to output the final instance embedding $\mathbf{s}$ as follows,

$$[\mathbf{s}]_j = \max_{1 \le i \le n} \{[\mathbf{h}_i]_j\}, \quad (5)$$

where $[\cdot]_j$ is the j-th value of a vector.

PCNN is an extension to CNN, which also adopts a convolution kernel to obtain hidden embeddings. Then, a piecewise max-pooling is applied over the hidden embeddings,

$$\begin{aligned} [\mathbf{s}^{(1)}]_j &= \max_{1 \le i \le i_1} \{[\mathbf{h}_i]_j\}, \\ [\mathbf{s}^{(2)}]_j &= \max_{i_1+1 \le i \le i_2} \{[\mathbf{h}_i]_j\}, \\ [\mathbf{s}^{(3)}]_j &= \max_{i_2+1 \le i \le n} \{[\mathbf{h}_i]_j\}, \end{aligned} \quad (6)$$

where $[\cdot]_j$ is the j-th value of a vector, i_1 and i_2 are entity positions. The final instance embedding $\mathbf{s}$ is achieved by concatenating these three pooling results as follows,

$$\mathbf{s} = [\mathbf{s}^{(1)}; \mathbf{s}^{(2)}; \mathbf{s}^{(3)}]. \quad (7)$$

3.4 Hierarchical Selective Attention

Given the entity pair (h, t) and its bag of instances $\mathcal{S}_{h,t} = \{s_1, s_2, \ldots, s_m\}$, we achieve the instance embeddings $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}$ using the sentence encoder. Afterwards, we apply a hierarchical selective attention over them to get the textual relation representation $\mathbf{r}_{h,t}$ for extracting relations. In

this part, we will first introduce a plain selective attention, and then introduce our hierarchical attention.

Plain Selective Attention

The plain selective attention scheme computes the attention score α_i for each instance s_i to indicate how well the instance can express the relation between the two entities. We assign a query vector $\mathbf{q}_r$ to each relation $r \in \mathcal{R}$ and the attention for each sentence in $\mathcal{S}_{h,t} = \{s_1, s_2, \ldots, s_m\}$ is defined as follows,

$$\begin{aligned} e_i &= \mathbf{q}_r^\top \mathbf{W}_s \mathbf{s}_i, \\ \alpha_i &= \frac{\exp(e_i)}{\sum_{j=1}^{m} \exp(e_j)}, \end{aligned} \tag{8}$$

where $\mathbf{W}_s$ is the weight matrix. The attention scores can be used in Eq. 1 to compute textual relation representations. For simplicity, we denote such a plain selective attention operation as the following equation,

$$\mathbf{r}_{h,t} = \mathrm{ATT}(\mathbf{q}_r, \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}). \tag{9}$$

Hierarchical Selective Attention

The inherent hierarchical structure of relations lead us to modeling hierarchical attention. Generally, given a relation set $\mathcal{R}$ of a KG $\mathcal{G}$ (e.g. Freebase), which consists of base-level relations (e.g. `/location/province/capital`), we can generate the corresponding higher-level relation set $\mathcal{R}^H$. The relations in the high-level set (e.g. `location`) are more general and common, which usually contain several sub-relations in the base-level set. We assume that the sub-relations of different relations are disjoint, in other words, the relation hierarchies are tree-structured. The generation process can be done recursively. In practice, we start from $\mathcal{R}^0 = \mathcal{R}$ which is the set of all relations we focus for RE, and generate $k-1$ times to get a total of k-level hierarchical relation sets $\{\mathcal{R}^0, \mathcal{R}^1, \ldots, \mathcal{R}^{k-1}\}$.

As shown in Figure 2, for a relation $r = r^0 \in \mathcal{R} = \mathcal{R}^0$, which is the focus for RE, we construct its hierarchical chain of parent relations by backtracking the relation hierarchy as follows,

$$(r^0, \ldots, r^{k-1}) \in \mathcal{R}^0 \times \ldots \times \mathcal{R}^{k-1}, \tag{10}$$

where r^{i-1} is the sub-relation of r^i.

As with the plain attention, we assign a query vector $\mathbf{q}_r$ to each relation $r \in \bigcup_{i=0}^{k-1} \mathcal{R}^i$. With the hierarchical chain, we compute attention operations on the each layer of the relation hierarchies to obtain corresponding textual relation representations,

$$\mathbf{r}^i_{h,t} = \mathrm{ATT}(\mathbf{q}_{r^i}, \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}). \tag{11}$$

During the training process, those relation query vectors of high-level relations (i.e., $\mathbf{q}_{r^i}$ with larger i) have more instances for training than those query vectors of base-level relations. Hence, the high-level query vectors are more robust for instance selection but with coarse-grained capability. In contrast, the base-level query vectors (i.e., $\mathbf{q}_{r^i}$ with smaller i) always suffer from data sparsity, especially for those long-tail base relations. Hence, the base-level query vectors can perform fine-grained instance selection but the performance is not stable.

Based on the hierarchical selective attention, we can simply concatenate the textual relation representations on different layers as the final representation,

$$\mathbf{r}_{h,t} = [\mathbf{r}^0_{h,t}; \ldots; \mathbf{r}^{k-1}_{h,t}]. \tag{12}$$

The representation $\mathbf{r}_{h,t}$ will be finally fed to compute the conditional probability $P(r|h,t,\mathcal{S}_{h,t})$ in Eq. 2. Note that, those high-level representations (i.e., $\mathbf{r}^i_{h,t}$ with larger i) are coarse-grained, and those base-level representations (i.e., $\mathbf{r}^i_{h,t}$ with smaller i) are fine-grained. These hierarchical representations can provide more informative information than single-layered attention for relation prediction, especially for those long-tail relations.

3.5 Initialization and Implementation Details

Here we introduce the learning and optimization details for our hierarchical attention model. During the training process, we minimize the cross entropy loss function. Given the collection of entity-pair bags $\pi = \{\mathcal{S}_{h_1,t_1}, \mathcal{S}_{h_2,t_2}, \ldots\}$ and corresponding labeled relations $\{r_1, r_2, \ldots\}$, we define the loss function as follows,

$$J(\theta) = -\frac{1}{|\pi|} \sum_{i=1}^{|\pi|} \log P(r_i|h_i, t_i, \mathcal{S}_{h_i,t_i}) + \lambda \|\theta\|_2^2, \tag{13}$$

where λ is a harmonic factor, and $\|\theta\|_2^2$ is the regularizer defined as L_2 normalization. All models are optimized using stochastic gradient descent (SGD).

4 Experiments

4.1 Datasets and Evaluation

We evaluate our models on the New York Times (NYT) dataset developed by Riedel et al. (2010), which is widely used in recent works (Lin et al., 2016; Zeng et al., 2017; Ji et al., 2017; Han et al., 2018; Liu et al., 2017; Wu et al., 2017; Huang and Wang, 2017; Feng et al., 2018; Zeng et al., 2018). The dataset has 53 relations including the NA relation which indicates relations of instances are not available. The training set has 522,611 sentences, 281,270 entity pairs and 18,252 relational facts. In the test set, there are 172,448 sentences, 96,678 entity pairs and 1,950 relational facts. In both the training and test set, we truncate the sentences which have 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. Besides precision-recall curves, we also show the precision values at the specific recall rate to conduct a more direct comparison, and calculate the micro and macro average precision scores to show the overall effect of different models. To further verify the effect of our hierarchical attention for few-shot entity pairs, we follow the previous works to report the Precision@N results. The dataset and baseline code can be found from Github [1] (Lin et al., 2016; Wu et al., 2017; Liu et al., 2017).

4.2 Parameter Settings

To fairly compare the results of our hierarchical attention models with those baselines, we also set most of the experimental parameters following Lin et al. (2016). Table 1 shows all experimental parameters used in the experiments. We apply dropout on the output layers of our models to prevent overfitting. For CNN, we set the dropout rate to 0.5. For PCNN, we observe that this model tends to overfit on the training set very quickly, and hence we set the dropout rate to 0.9 to alleviate the overfitting problem. We also pre-train the sentence encoder of PCNN before training our hierarchical attention.

[1] NRE, AtNRE and soft-label-RE

Batch Size B	160
Learning Rate α	0.2
Hidden Layer Dimension k_c for CNNs	230
Word Dimension k_w	50
Position Dimension k_p	5
Convolution Kernel Size m	3

Table 1: Parameter settings.

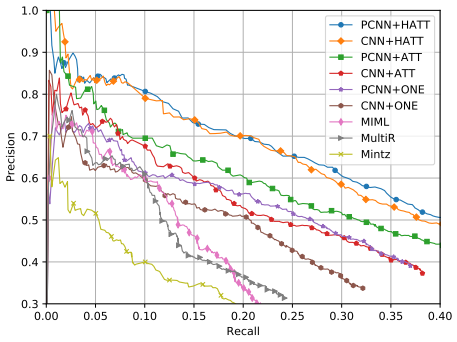

Figure 3: Precision-recall curves for the proposed model and various baseline models.

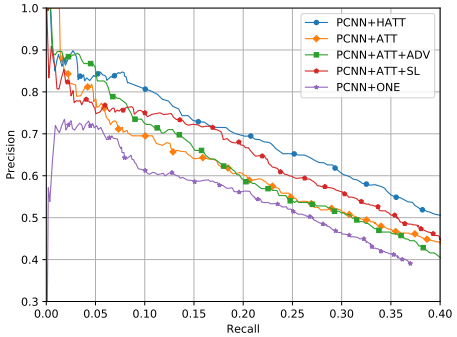

Figure 4: Precision-recall curves for the proposed model and various attention-based neural models.

4.3 Overall Evaluation Results

To evaluate the performance of our proposed hierarchical models, we compare the precision-recall curves of our models with various previous relation extraction models. The evaluation results are shown in Figure 3 and Figure 4. We report the results of the neural architectures including CNN and PCNN with various attention-based methods: **+HATT** is our hierarchical attention method; **+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); **+ATT+SL** is the attention-based model using soft-labeling method to mitigate the side effect of the wrong label-

ing problem at entity-pair level (Liu et al., 2017); **+ONE** is a vanilla MIL neural model without attention schemes (Zeng et al., 2015). 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).

From the results, we observe that:

(1) All methods have reasonable precision when recall is smaller than 0.05. When the recall gradually grows, the performance of the feature-based methods drops much more faster than those neural models. It shows that human-designed features are very limited as compared to neural models, especially in a noisy data environment. Hence, for simplicity, we mainly show the results of our models and other attention-based neural models in the following experiments.

(2) Both for CNNs and PCNNs, the models with attention schemes outperform the vanilla models without attention schemes. Though vanilla neural models are powerful for relation classification, it is still difficult to address data noise. The attention-based methods apply attentions over multiple instances and dynamically reduce the influence of noisy instances, which can effectively improve the performance of RE and achieve the state-of-the-art results.

(3) As shown in both of the figures, the models using hierarchical attention (HATT) achieve the best results among all the attention-based models. Even when compared with PCNN+ATT+ADV and PCNN+ATT+SL which adopt sophisticated denoising schemes and extra information, our models still keep significant advantages. This indicates that, as compared to the conventional plain attention schemes which handle each relation in isolation, our method can better take advantage of the rich correlations among relations. We believe the performance of our hierarchical attention scheme can be further improved by adopting extra mechanisms like adversarial training, reinforcement learning and soft-labeling at entity-pair level, which will be left as our future work.

4.4 Effect of Hierarchical Attention for Different Relations

To further verify the effectiveness of our hierarchical attention method for different relations, we evaluate the RE performance of our method and conventional attention methods. Since we focus more on the performance of those top-ranked results, we report the precision scores when the recall is 0.1, 0.2, 0.3 and their mean. We also report micro average scores and macro average scores in this experiment. As an approximation of the area under the precision-recall curve, the micro average score gives a more complete view of the model performance. Since the micro average score generally overlooks the influences of those long-tail relations, we use the macro average score to give more emphasis on long-tail relations in test sets, which is often neglected by the previous works.

Method		0.1	0.2	0.3
CNN	+ATT	67.5	52.8	58.5
	+HATT	**78.9**	**69.9**	**58.5**
PCNN	+ATT	69.4	60.6	51.6
	+HATT	**80.6**	**69.5**	**60.7**
Method		Mean	Micro	Macro
CNN	+ATT	55.4	31.8	8.2
	+HATT	**69.1**	**41.7**	**16.5**
PCNN	+ATT	60.5	38.0	15.1
	+HATT	**70.3**	**42.3**	**17.0**

Table 2: Precision (%) of attention-based models for different recalls.

Training Instances		<100			<200		
Hits@K (Micro)		10	15	20	10	15	20
CNN	+ATT	<5.0	<5.0	21.1	<5.0	30.0	50.0
	+HATT	**5.3**	**36.8**	**52.6**	**40.0**	**60.0**	**70.0**
PCNN	+ATT	<5.0	10.5	47.4	33.3	43.3	66.7
	+HATT	**31.6**	**52.6**	**63.2**	**53.3**	**70.0**	**76.7**
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**
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**

Table 3: Accuracy (%) of Hits@K on relations with training instances fewer than 100/200.

The evaluation results are shown in Table 2, and from the results we observe that: Our HATT method achieves consistent and significant improvements as compared to the plain ATT method. From the micro and macro average precision scores, we find that our HATT method effectively improves RE performance especially for those long-tail relations. As compared to the plain ATT method, our method can take advantage of correlations among relations to achieve the improvement, especially on the long-tail relations.

To further demonstrate the improvements in performance on long-tail relations after introduc-

Test Mode	ONE				TWO				ALL			
P@N	100	200	300	Mean	100	200	300	Mean	100	200	300	Mean
CNN+ONE	68.3	60.7	53.8	60.9	70.3	62.7	55.8	62.9	67.3	64.7	58.1	63.4
CNN+ATT	76.2	65.2	60.8	67.4	76.2	65.7	62.1	68.0	76.2	68.6	59.8	68.2
CNN+HATT	**88.0**	**74.5**	**68.0**	**76.8**	**85.0**	**76.0**	**73.0**	**78.0**	**88.0**	**79.0**	**77.7**	**81.6**
PCNN+ONE	73.3	64.8	56.8	65.0	70.3	67.2	63.1	66.9	72.3	69.7	64.1	68.7
PCNN+ATT	73.3	69.2	60.8	67.8	77.2	71.6	66.1	71.6	76.2	73.1	67.4	72.2
PCNN+HATT	**84.0**	**76.0**	**69.7**	**76.6**	**85.0**	**76.0**	**72.7**	**77.9**	**88.0**	**79.5**	**75.3**	**80.9**

Table 4: Top-N precision (P@N) for RE on the entity pairs with different number of instances (%).

Relation: /people/person/children		
High	Good	**David** and Jody Smith and their son **Nathan** of Ankeny , Iowa , stayed at the hotel, . . .
	Bad	. . . doting grandfather of Amanda, Lindsay, **David**, Alexa, Reese, Paige and **Nathan**.
Base	Good	. . . cherished grandfather of **David**, Michael, Jason, Vicky, Andrew, Sam and **Nathan**
	Bad	**David** and Jody Smith and their son **Nathan** of Ankeny, Iowa, stayed at the hotel . . .

Table 5: Example sentences for case study.

ing relation hierarchies, we extract a subset of the test dataset in which all the relations has 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 recommanded 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 both micro and macro average Hits@K accuracies for these subsets. From the evaluation results in Table 3, we observe that:

(1) For both CNN and PCNN models, our hierarchical attention outperforms the plain attention model. By taking advantage of the relation hierarchy, our models can learn better about long-tail relations via correlation information among relations. We also observe that even our hierarchical CNN model presents a better performance than the plain PCNN model. This shows the power of relation hierarchies, which makes our simpler CNN model outperforms the PCNN model on those long-tail relations.

(2) Although our HATT method has achieved obvious progress on the long-tail relations as compared with the plain ATT method, the results of all these methods are still far from satisfactory. This indicates that distantly supervised RE models suffer from not only the wrong labeling problem, but also the long-tail relation problem. We will incorporate more schemes and extra information to solve this problem in the future.

4.5 Effect of Hierarchical Attention with Different Instances

Since our method mainly focuses on modifications over selective attention, we also conduct Precision@N tests on those entity pairs with few instances following (Lin et al., 2016). We use the three test settings for this experiment: the ONE test set where we randomly select one instance for each entity pair for evaluation; the TWO test set where we randomly select two instances for each entity pair; the ALL test set where we use all instances for each remaining entity pair for evaluation. For the ONE and TWO test set, we intend to show that taking correlation information among relations into consideration can lead to a better relation classifier. The ALL test set is designed to show the effect of our attention over multiple instances. We report the precision values of top N triples extracted, where $N \in \{100, 200, 300\}$.

The evaluation results are shown in Table 4, and from the results we observe that:

(1) The performance of all methods is generally improved as the instance number increases. This shows that the selective attention model can effectively take advantage of information from multiple noisy instances by combining useful instances while discarding useless ones.

(2) Our HATT method has higher precision values in the ONE test set. This indicates that even in an insufficient information scenario, correlations among relations can be caught by our hierarchical attention.

4.6 Case Study

We give some examples of how our hierarchical selective attention takes effect in selecting the sentences. In Table 5, we display the sentences that are scored highest ("Good") or lowest ("Bad") by the attention of different hierarchical levels ("High" and "Base").

The relation /people/person/children

has fewer than 1000 training instances and it is a long-tail relation. For this relation, the instance recommended by the higher-level attention straightforwardly expresses the relational fact that *Nathan* is the child of *David* by telling that *Nathan* is *David*'s son, while the sentence with the low attention score actually gives the relationship of being at the same generation. On the contrary, the lower-level attention mistakenly assigns high attention to the incorrect sentence. This example shows that our hierarchical attention is beneficial for these long-tail relations.

5 Conclusion and Future Work

In this paper, we take advantage of relation hierarchies and propose a novel hierarchical instance-level attention for relation extraction. As compared with previous attention-based methods, our hierarchical attention provides coarse-to-fine granularity in instance selection and performs better extraction for long-tail relations. We conduct various experiments and the evaluation results show that incorporating the inherent hierarchical structure of relations into attention schemes can take advantage of correlations among relations and improve the performance significantly.

In the future, we plan to explore the following directions: (1) It will be promising to adopt extra information to help train more efficient models for solving the long-tail relation problem. (2) We may also combine our attention method with recent denoising methods to further improve model performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC No. 61572273, 61532010). This work is also funded by the Natural Science Foundation of China (NSFC) and the German Research Foundation (DFG) in Project Crossmodal Learning, NSFC 61621136008 / DFC TRR-169.

References

Wei Bi and James T Kwok. 2011. Multi-label classification on tree-and dag-structured hierarchies. In *Proceedings of ICML*, pages 17–24.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of KDD*, pages 1247–1250.

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

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Proceedings of AAAI*.

Geoffrey E Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of COGSCI*, volume 1, page 12.

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.

Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric P Xing. 2015. Entity hierarchy embedding. In *Proceedings of ACL*, volume 1, pages 1292–1300.

Yi Yao Huang and William Yang Wang. 2017. Deep residual learning for weakly-supervised relation extraction. In *Proceedings of EMNLP*, pages 1803–1807.

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

Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *Proceedings of ISWC*.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *Proceedings of WordNet: An electronic lexical database*, 49(2):265–283.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, 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 EMNLP*, pages 1790–1795.

Andrew McCallum, Ronald Rosenfeld, Tom M Mitchell, and Andrew Y Ng. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of ICML*, volume 98, pages 359–367.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, pages 1003–1011.

Raymond J Mooney and Razvan C Bunescu. 2006. Subsequence kernels for relation extraction. In *Proceedings of NIPS*, pages 171–178.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of AISTATS*, pages 246–252.

Simone Paolo Ponzetto and Michael Strube. 2007. Knowledge derived from wikipedia for computing semantic relatedness. *Proceedings of JAIR*, 30:181–212.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, pages 148–163.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL*, pages 74–84.

Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. 2005. Learning hierarchical multi-category text classification models. In *Proceedings of ICML*, pages 744–751.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL-IJCNLP*, pages 626–634.

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.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In *Proceedings of NAACL*, pages 886–896.

Patrick Verga and Andrew McCallum. 2016. Row-less universal schema. In *Proceedings of ACL*, pages 63–68.

Nakul Verma, Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. 2012. Learning hierarchical similarity metrics. In *Proceedings of CVPR*, pages 2280–2287.

Kilian Q Weinberger and Olivier Chapelle. 2009. Large margin taxonomy embedding for document categorization. In *Proceedings of NIPS*, pages 1737–1744.

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

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016. Representation learning of knowledge graphs with hierarchical types. In *Proceedings of IJCAI*, pages 2965–2971.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of EMNLP*, pages 1785–1794.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Proceedings of JMLR*, 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 *Proceedings of EMNLP*, pages 1768–1777.

Xiangrong Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Large scaled relation extraction with reinforcement learning. In *Proceedings of AAAI*.

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

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP*, pages 35–45.

Bin Zhao, Fei Li, and Eric P Xing. 2011. Large-scale category structure aware image categorization. In *Proceedings of NIPS*, pages 87–96.

Denny Zhou, Lin Xiao, and Mingrui Wu. 2011. Hierarchical classification via orthogonal transfer. In *Proceedings of ICML*.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL*, pages 427–434.