Reinforcement Learning for Relation Classification from Noisy Data

Jun Feng[§], Minlie Huang[§*], Li Zhao[‡], Yang Yang[†], and Xiaoyan Zhu[§]
[§] State Key Lab. of Intelligent Technology and Systems, National Lab. for Information Science and Technology
Dept. of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China
[‡] Microsoft Research Asia
[†] College of Computer Science and Technology, Zhejiang University
feng-j13@mails.tsinghua.edu.cn, aihuang@tsinghua.edu.cn, lizo@microsoft.com
yangya@zju.edu.cn, zxy-dcs@tsinghua.edu.cn

Abstract

Existing relation classification methods that rely on distant supervision assume that a bag of sentences mentioning an entity pair are all describing a relation for the entity pair. Such methods, performing classification at the bag level, cannot identify the mapping between a relation and a sentence, and largely suffers from the noisy labeling problem. In this paper, we propose a novel model for relation classification at the sentence level from noisy data. The model has two modules: an instance selector and a relation classifier. The instance selector chooses high-quality sentences with reinforcement learning and feeds the selected sentences into the relation classifier, and the relation classifier makes sentence-level prediction and provides rewards to the instance selector. The two modules are trained jointly to optimize the instance selection and relation classification processes. Experiment results show that our model can deal with the noise of data effectively and obtains better performance for relation classification at the sentence level.

Introduction

Relation classification, aiming to categorize semantic relations between two entities given a plain text, is an important problem in natural language processing, particularly for knowledge graph completion and question answering. Most existing works for relation classification adopt supervised learning approaches, either based on traditional handcrafted features (Mooney and Bunescu 2005; Zhou et al. 2005) or based on the features automatically generated by deep neural networks (Zeng et al. 2014; dos Santos, Xiang, and Zhou 2015), but all require high-quality annotated data.

In order to obtain large-scale training data, distant supervision (Mintz et al. 2009) was proposed by assuming that if two entities have a relation in a given knowledge base, all sentences that contain the two entities will mention that relation. Although distant supervision is effective to label data automatically, it suffers from the *noisy labeling problem*. Taking the triple (Barack_Obama, *BornIn*, United_States) as an example, the noisy sentence *"Barack Obamba is the 44th president of the United State"* will be regarded as a positive instance by distant supervision and a *BornIn* relation is assigned to this sentence, although the sentence does not describe the relation *BornIn* at all.

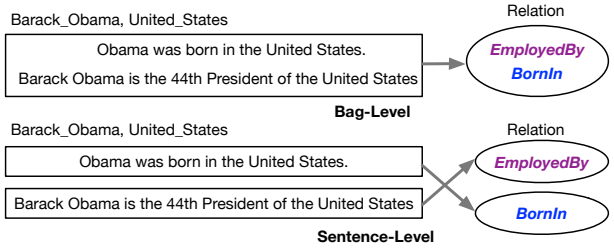

Figure 1: Bag-level: Relations are mapped to a bag of sentences, each of which contains the same entity pair; Sentence-level: Each sentence is mapped to a specific relation.

To address the issue of noisy labeling, previous studies adopt multi-instance learning to consider the noises of instances (Riedel, Yao, and McCallum 2010; Hoffmann et al. 2011; Surdeanu et al. 2012; Zeng et al. 2015; Lin et al. 2016; Ji et al. 2017). In these studies, the training and test process is proceeded at the bag level, where a bag contains noisy sentences mentioning the same entity pair but possibly not describing the same relation. As a result, previous studies suffer from two limitations: 1) Unable to handle the sentence-level prediction; 2) Sensitive to the bags with all noisy sentences which do not describe a relation at all.

To better explain the first limitation, we show an example in Figure 1. Bag-level prediction can find the two relations "*EmployedBy*" and "*BornIn*" between the entity pair "*Barack_Obama*" and "*United_States*". However, sentence-level prediction is able to further map each relation to the corresponding sentences. As for the second limitation, for each bag, previous bag-level methods retain at least one sentence, even if all the sentences in a given bag are noisy (not describing the relation). Such bags, produced by distant supervision, are quite common. For instance, our investigation on a widely used dataset[1] shows that 53% out of 100 sample bags have no sentences that describe the relation. Such noisy bags will definitely decrease the performance of relation classification.

*Corresponding author: Minlie Huang, aihuang@tsinghua.edu.cn

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

In this paper, to handle the above two limitations, we propose a novel relation classification model consisting of two modules: instance selector and relation classifier. By having an explicit instance selector[2], we are able to first select high-quality sentences from a sentence bag, and then predict a relation at the sentence level by the relation classifier. To handle the second limitation, our instance selector will filter the entire bag if all sentences are labeled incorrectly. The major challenge here is how to train the two modules jointly, particularly when the instance selector has no explicit knowledge about which sentences are labeled incorrectly.

We address this challenge by casting the instance selection task as a reinforcement learning problem (Sutton and Barto 1998). Intuitively, although we do not have an explicit supervision for the instance selector, we can measure the utility of the selected sentences as a whole. Thus, the instance selection process has the following two properties: first, *trial-and-error-search*, meaning that the instance selector attempts to choose some sentences and obtain feedback (or *reward*) on the quality of the selected sentences from the relation classifier; second, the feedback from the relation classifier can be obtained only when we finish the instance selection process, which is typically *delayed*. These two properties naturally inspire us to utilize reinforcement learning techniques.

Our contributions in this work include:

- We propose a new model for relation classification, which consists of an instance selector and a relation classifier. This formalization enables our model to extract relations at the sentence level on the cleansed data.
- We formulate instance selection as a reinforcement learning problem, which enables the model to perform instance selection without explicit sentence-level annotations but just with a weak supervision signal from the relation classifier.

Related Work

Relation classification is a common task in natural language processing. Many approaches have been developed, particularly with supervised methods (Mooney and Bunescu 2005; Zhou et al. 2005; Zelenko, Aone, and Richardella 2003). However, such supervised methods heavily rely on high-quality labeled data.

Recently, neural models have been widely applied to relation classification (Zeng et al. 2014; dos Santos, Xiang, and Zhou 2015; Mooney and Bunescu 2005; Yang et al. 2016) including convolutional neural networks, recursive neural network (Ebrahimi and Dou 2015; Liu et al. 2015), and long short-term memory network (Miwa and Bansal 2016; Xu et al. 2015; Miwa and Bansal 2016). In (Wang et al. 2016), two levels of attention is proposed in order to better discern patterns in heterogeneous contexts for relation classification.

In general, a large amount of labeled data are required to train neural models, which is quite expensive. To address this issue, distant supervision was proposed (Mintz et al. 2009) by assuming that all sentences that mention two entities of a fact triple describe the relation in the triple. In spite of the success of distance supervision, such methods suffer from the noisy labeling issue. To alleviate this issue, many studies formulated relation classification as a multi-instance learning problem (Riedel, Yao, and McCallum 2010; Hoffmann et al. 2011; Surdeanu et al. 2012; Zeng et al. 2015). In (Lin et al. 2016; Ji et al. 2017; Tianyu Liu and Sui 2017), a sentence-level attention mechanism over multiple instances was proposed and incorrect sentences can be down-weighted. However, such multi-instance learning models all predict relations at the bag level but not at the sentence level, and they can not deal with the bags in which all sentences are not describing a relation at all. There are other approaches to reduce the noise of distant supervision using active learning (Sterckx et al. 2014) and negative patterns (Takamatsu, Sato, and Nakagawa 2012).

Previous methods are all at the bag level but not at the sentence level and as such, they cannot find the exact mapping between a relation and a sentence. Furthermore, these methods are unable to handle the bags in which all the sentences are not describing the relation. To address these issues, we propose a new framework which first selects correct sentences in the framework of reinforcement learning (Sutton and Barto 1998; Narasimhan, Yala, and Barzilay 2016) and then predicts relations from each sentence in the cleansed data.

Methodology

We propose a new relation classification framework, which is able to select correct sentences from noisy data for better relation classification. The proposed framework can predict relations at the sentence level from the cleansed data, rather than at the bag level. Sentence-level prediction is more friendly to the tasks that need to comprehend sentences such as question answering and semantic parsing.

Our framework consists of two key modules: the instance selector which selects correct sentences from noisy data, and the relation classifier which predicts relation and updates its parameters with cleaned data. The two modules interacts with each other during the training process.

Problem Definition

Formally, we decompose the task of relation classification into two sub-problems in this paper: instance selection and relation classification.

We formulate the instance selection problem as follows: given a set of <sentence, relation label> pairs as $X = \{(x_1, r_1), (x_2, r_2), \ldots, (x_n, r_n)\}$, where x_i is a sentence associated with two entities (h_i, t_i) and r_i is a noisy relation label produced by distant supervision. The goal is to determine which sentence truly describes the relation and should be selected as a training instance.

The relation classification problem is formulated as follows: given a sentence x_i and the mentioned entity pair (h_i, t_i), the goal is to predict the semantic relation r_i in x_i. Essentially, the model estimates the probability: $p_\Phi(r_i|x_i, h_i, t_i)$.

[2]Instance is referred to a sentence in this paper.

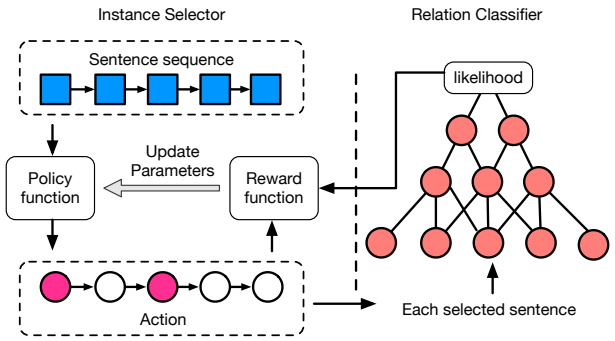

Figure 2: Overall process. The instance selector chooses sentences according to a policy function, and then the selected sentences are used to train a better relation classifier. The instance selector updates its parameters, with a reward computed from the relation classifier.

Overview

The proposed model is based on a reinforcement learning framework and consists of two components: the *instance selector* and the *relation classifier*. In the instance selector, each sentence x_i has a corresponding *action* a_i to indicate whether or not x_i will be selected as a training instance for relation classification. The state s_i is represented by the current sentence x_i, the already chosen sentences among $\{x_1, \cdots, x_{i-1}\}$, and the entity pair h_i and t_i in sentence x_i. The instance selector samples an action given the current state according to a stochastic policy. For the relation classifier, it adopts a convolutional architecture to automatically determine the semantic relation for an entity pair in a given sentence. The instance selector distills the training data to the relation classifier to train the convolutional neural network. Meanwhile, the relation classifier gives feedback to the instance selector to refine its policy function. Figure 2 gives an illustration of how the proposed framework works.

With the help of the instance selector, our method directly filters out noisy sentences. Unlike reducing the weights of noisy sentences (Lin et al. 2016) or retaining one sentence in a bag (Zeng et al. 2014), our method is better at dealing with noisy data. The relation classifier is trained and tested at the sentence level on the cleansed data, whereas previous models treat the sentence bag as a whole and predict relation at the bag level.

Instance Selector

We cast instance selection as a reinforcement learning problem. The instance selector is the agent, who interacts with the environment that consists of data and the relation classifier. The agent follows a policy to decide which action (choosing the current sentence or not) at each state (consisting of the current sentence, the chosen sentence set, and the entity pair), and then receive a reward from the relation classifier at the terminal state when all the selections are made.

As aforementioned, we can obtain a delayed reward from the relation classifier only when the selection on all the training instances are finished. Thus, we can only update the policy function once for each scan of the entire training data, which is obviously inefficient. To obtain more feedbacks and to make the training process more efficiently, we split the training sentence instances $X = \{x_1, \ldots, x_n\}$ into N bags $\mathbf{B} = \{B^1, B^2, \ldots, B^N\}$ and compute a reward when we finish data selection in a bag. Each bag corresponds to a distinct entity pair, and each bag B^k is a sequence of sentences $\{x_1^k, x_2^k, \ldots, x_{|B^k|}^k\}$ with the same relation label r^k, however, the relation label is noisy. We define the action as selecting a sentence or not according to a policy function. The reward is computed once the selection decisions are completed on one bag. When the training process of the instance selector is completed, we merge all the selected sentences in each bag to obtain a cleansed dataset $\hat{X}$. Then, the cleansed data will be used to train the relation classifier at the sentence level.

We will introduce (i.e., *state*, *action*, and *reward*) as follows. To be clear, we will omit the superscript k which denotes the bag index. Thus, the formulation hereafter is based on only one bag.

State. The state s_i represents the current sentence, the already selected sentences, and the entity pair when making decision on the i-th sentence of the bag B. We represent the state as a continuous real-valued vector $\boldsymbol{F}(s_i)$, which encodes the following information: 1) The vector representation of the current sentence, which is obtained from the nonlinear layer of the CNN for relation classification; 2) The representation of the chosen sentence set, which are the average of the vector representations of all chosen sentences; 3) The vector representations of the two entities in a sentence, obtained from a pre-trained knowledge graph embedding table.

Action. We define an action $a_i \in \{0, 1\}$ to indicate whether the instance selector will select the i-th sentence of the bag B or not. We sample the value of a_i by its policy function $\pi_\Theta(s_i, a_i)$, where Θ is the parameters to be learned. In this work, we adopt a logistic function as the policy function:

$$\begin{aligned}\pi_\Theta(s_i, a_i) &= P_\Theta(a_i|s_i) \\ &= a_i\sigma(\boldsymbol{W} * \boldsymbol{F}(s_i) + \boldsymbol{b}) \\ &+ (1 - a_i)(1 - \sigma(\boldsymbol{W} * \boldsymbol{F}(s_i) + \boldsymbol{b}))\end{aligned} \tag{1}$$

where $\boldsymbol{F}(s_i)$ is the state feature vector, and $\sigma(.)$ is the sigmoid function with the parameter $\Theta = \{\boldsymbol{W}, \boldsymbol{b}\}$.

Reward. The reward function is an indicator of the utility of the chosen sentences. For certain bag $B = \{x_1, \ldots, x_{|B|}\}$, we sample an action for each sentence, to determine whether the current sentence should be selected or not. We assume that the model has a terminal reward when it finishes all the selection. Therefore we only receive a delayed reward at the terminal state $s_{|B|+1}$. The reward is zero at other states. Therefore, the reward is defined as follows:

$$r(s_i|B) = \begin{cases} 0 & i < |B| + 1 \\ \frac{1}{|\hat{B}|} \sum\limits_{x_j \in \hat{B}} \log p(r|x_j) & i = |B| + 1 \end{cases} \tag{2}$$

where $\hat{B}$ is the set of selected sentences, which is a subset of B, and r is the relation label of bag B. As shown in Figure

2, $p(r|x_j)$ is calculated by the relation classifier which is given by a CNN model. For the special case $\hat{B} = \emptyset$, we set the reward as the average likelihood of all sentences in the training data, which enables our instance selector to exclude noisy bag effectively.

Note that the relation classifier is at the sentence-level since it computes $p(r|x)$ for each sentence. The reward is computed on a new bag of sentences selected by the instance selector. Essentially, the above reward evaluates the overall utility of all the actions made by the policy. It supervises the instance selector to maximize the average likelihood of the chosen instances, which makes the objective function of the instance selector consistent with the relation classifier.

In the selection process, not only the final action contributes to this reward, but also all the previous actions do. Therefore, this reward is delayed, and can be handled very well by reinforcement learning techniques (Sutton and Barto 1998).

Optimization. For a bag B, we aim to maximize the expected total reward. More formally, our objective function is defined as

$$\begin{aligned} J(\Theta) &= V_\Theta(s_0|B) \\ &= E_{s_0,a_0,s_1,...,s_i,a_i,s_{i+1}...}[\sum_{i=0}^{|B|+1} r(s_i|B)] \end{aligned} \tag{3}$$

where $a_i \sim \pi_\Theta(s_i, a_i)$, $s_{i+1} \sim P(s_{i+1}|s_i, a_i)$. The transition functions $P(s_{i+1}|s_i, a_i)$ are equal to 1, since the state s_{i+1} is fully determined by the state s_i and a_i. V_Θ is the value function, and $V_\Theta(s_0|B)$ represents the expected future total reward that we can obtain by starting at certain state s_0 following policy $\pi_\Theta(s_i, a_i)$.

According to the policy gradient theorem (Sutton et al. 1999) and the REINFORCE algorithm (Williams 1992), we compute the gradient in the following way. For each bag B, we sample an action for each state sequentially according to the current policy. We then get a sampled trajectory $\{s_1, a_1, s_2, a_2, ..., s_{|B|}, a_{|B|}, s_{|B|+1}\}$ and a corresponding terminal reward $r(s_{|B|+1}|B)$. Since we only have a non-zero terminal reward, the value function is the same for all states from s_1 to $s_{|B|}$, namely $v_i = V(s_i|B) = r(s_{|B|+1}|B)$, for $i = 1, 2, ..., |B|$. We update the current policy using the following gradient:

$$\Theta \leftarrow \Theta + \alpha \sum_{i=1}^{|B|} v_i \nabla_\Theta \log \pi_\Theta(s_i, a_i) \tag{4}$$

Relation classifier

In the relation classifier, we adopt a CNN architecture to predict relations. The CNN network has an input layer, a convolution layer, a max pooling layer and a non-linear layer from which the representation is used for relation classification.

Input layer. For each sentence x, we represent it as a list of vectors $\mathbf{x} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m)$. Each representation vector consists of two parts: one is the word embedding; the other is the position embedding. Word embeddings are obtained from word2vec[3], and the dimension is d^w. Similar to (Zeng et al. 2014), we use d^p-dimensional position embeddings, which are vector representations of the relative distances from the current word respectively to the head or tail entities in this sentence. We concatenate the word and position embeddings of each word to form a new vector $\mathbf{w}_i$ ($\mathbf{w}_i \in \mathbb{R}^d$, and $d = d^w + 2 \times d^p$), and then input these vectors to the CNN model.

CNN. In order to obtain high-level and abstractive representation of the raw input of a sentence, we apply a CNN structure for relation classification. This can be briefly described as below:

$$\mathbf{L} = \text{CNN}(\mathbf{x}) \tag{5}$$

where $\mathbf{x}$ is the input vectors as described in the input layer

[3]https://code.google.com/p/word2vec/

ALGORITHM 1: Overall Training Procedure

1. Initialize the parameters of the CNN model of relation classifier and the policy network of instance selector with random weights respectively
2. Pre-train the CNN model to predict relation r_i given the sentence x_i by maximizing $\log p(r_i|x_i)$
3. Pre-train the policy network by running Algorithm 2 with the CNN model fixed.
4. Run Algorithm 2 to jointly train the CNN model and the policy network until convergence

ALGORITHM 2: Reinforcement Learning Algorithm for the Instance Selector

Input: Episode number L. Training data $\mathbf{B} = \{B^1, B^2, \ldots, B^N\}$. A CNN and a policy network model parameterized by Φ and Θ, respectively

Initialize the target networks as: $\Phi' = \Phi, \Theta' = \Theta$

for *episode* $l = 1$ *to* L **do**

  Shuffle $\mathbf{B}$ to obtain the bag sequence $\mathbf{B} = \{B^1, B^2, \ldots, B^N\}$

  foreach $B^k \in \mathbf{B}$ **do**

    Sample instance selection actions for each data instance in B^k with Θ':

    (To be clear, we omit the superscript k below)

    $A = \{a_1, \ldots, a_{|B|}\}, a_i \sim \pi_{\Theta'}(s_i, a_i)$

    Compute delayed reward $r(s_{|B|+1}|B)$

    Update the parameter Θ of instance selector:

    $\Theta \leftarrow \Theta + \alpha \sum_i v_i \nabla_\Theta \log \pi_\Theta(s_i, a_i)$, where $v_i = r(s_{|B|+1}|B)$

  end

  Update Φ in the CNN model

  Update the weights of the target networks:

  $\Theta' = \tau\Theta + (1-\tau)\Theta'$

  $\Phi' = \tau\Phi + (1-\tau)\Phi'$

end

and $\mathbf{L} \in \mathbb{R}^{d^s}$ is the output of the max pooling layer. In this structure, there is a convolution layer, and a max pooling layer. The convolution operation is performed on 3 consecutive words, and the number of feature maps d^s is set to 230, the same as the setting of (Lin et al. 2016). Hence, the convolution parameters are $\mathbf{W}_f \in \mathbb{R}^{d^s \times (3d)}$ and $\mathbf{b}_f \in \mathbb{R}^{d^s}$.

Then, the probability for relation prediction $p(r|x;\mathbf{\Phi})$ is given as follows:

$$p(r|x;\mathbf{\Phi}) = softmax(\mathbf{W}_r * tanh(\boldsymbol{L}) + \mathbf{b}_r) \quad (6)$$

where $\mathbf{W}_r \in \mathbb{R}^{n_r \times d^s}$ and $\mathbf{b}_r \in \mathbb{R}^{n_r}$ are parameters in the fully-connected layer, n_r is the total number of relation types, and $\mathbf{\Phi} = \{\mathbf{W}_f, \mathbf{b}_f, \mathbf{W}_r, \mathbf{b}_r\}$.

The key difference between our relation classifier and other studies lies in that our classifier performs relation classification at the sentence level. The input to the relation classifier in other studies is a bag of sentences. Instead, the input to ours is just one sentence, since we already filter out noisy sentences with the instance selector.

Loss function. Given the selected training set $\{\hat{X}\}$ provided by the instance selector, we define the objective function of the relation classifier using cross-entropy as follows:

$$\mathcal{J}(\Phi) = -\frac{1}{|\hat{X}|}\sum_{i=1}^{|\hat{X}|} \log p(r_i|x_i;\Phi) \quad (7)$$

Model Training

As the instance selector and the relation classifier are correlated mutually, we train them jointly.The complete joint training process is described in Algorithm 1. To optimize the policy network in the instance selector, we use a Monto-Carlo based policy gradient method (Williams 1992), which favors actions with high sampled reward. To optimize the CNN component, we use a gradient descent method to minimize the objective function (i.e., Eq. 7). We pre-train the model before the joint training process starts. We first pre-train the CNN in the relation classifier, and then pre-train the policy function by computing the reward with the pre-trained CNN, while the parameters of the CNN model are frozen. At last, we jointly train the instance selector and the relation classifier. We found such a pre-training strategy is quite crucial for our method, which is also widely recommended by many other reinforcement learning studies(Bahdanau et al. 2016).

Algorithm 2 presents the details of the joint training process. The relation classifier provides a mechanism of computing the rewards of the selected sentences to refine the instance selector. The instance selector chooses high-quality data by excluding wrongly labeled sentences to better train the relation classifier. In order to have a stable update, we take advantage of a target policy network and a target CNN with parameter sets Θ' and Φ' respectively, similar to (Lillicrap et al. 2015). The parameters in the target networks are updated much more slowly than the original ones. We update Θ' and Φ' by linear interpolation: $\Theta' \leftarrow (1-\tau)\Theta' + \tau\Theta$ and $\Phi' \leftarrow (1-\tau)\Phi' + \tau\Phi$, where $\tau \ll 1$ is a hyper-parameter.

Experiment

Experiment Setup

Dataset. To evaluate our model, we adopted a widely used dataset[4] generated by the sentences in NYT[5] and developed by (Riedel, Yao, and McCallum 2010). There are 522,611 sentences, 281,270 entity pairs, and 18,252 relational facts in the training data; and 172,448 sentences, 96,678 entity pairs and 1,950 relational facts in the test data. Among the data, there are 39,528 unique entities and 53 unique relations from Freebase including a special relation *NA* that signifies no relation between two entities in a sentence.

Word and entity embedding. We adopted word2vec to train the word embeddings on the NYT corpus. For entity embedding, we implemented the TransE model (Bordes et al. 2013) and trained it on a set of Freebase fact triples whose entities have been mentioned in the training and test data.

Model pre-training. As described in Algorithm 2, we pre-trained the relation classifier and instance selector before the joint training process. As the reward is calculated based on the CNN model in the relation classifier, we first pre-trained the CNN model on the entire training data. Then, we fixed the parameters of the CNN model and pre-trained the policy function in the instance selector where the reward is obtained from the fixed CNN model.

Parameter setting. Similar to previous studies, we tuned our model using three-fold cross validation. For the parameters of the instance selector, we set the dimension of entity embedding as 50, the learning rate as 0.02/0.01 at the pre-training stage and joint training stage respectively. The delay coefficient τ is 0.001.

For the parameters of the relation classifier, the word embedding dimension $d^w = 50$ and the position embedding dimension $d^p = 5$. The window size of the convolution layer l is 3. The learning rate of the instance selector is $\alpha = 0.02$ both at the pre-training and joint training stage. The batch size is fixed to 160. The training episode number $L = 25$. We employed a dropout strategy with a probability of 0.5 during the training of the CNN component.

Sentence-Level Relation Classification

As discussed previously, the key difference between our method and other models lies in that our method can perform sentence-level relation classification. We conducted manual evaluation on relation classification in this section.

Evaluation settings. We predicted a relation label for each sentence, instead of for each bag. For example, the task in Figure 1 needs to map the first sentence to relation "*BornIn*" and the second sentence to "*EmployedBy*".

Since the data obtained from distant supervision are noisy, we randomly chose 300 sentences and manually labeled the relation type for each sentence to evaluate the classification

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

[5] New York Times, a widely used text corpus.

Method	Macro F_1	Accuracy
CNN	0.40	0.60
CNN+Max	0.06	0.34
CNN+ATT	0.29	0.56
CNN+RL(ours)	**0.42**	**0.64**

Table 1: Performance on sentence-level relation classification.

performance. We adopted accuracy and macro-averaged F_1 as the evaluation metric.

Baselines. We adopted three state-of-the-art baselines:

- **CNN** (Zeng et al. 2014) is a sentence-level classification model. It does not consider the noisy labeling problem.
- **CNN+Max** (Zeng et al. 2015) is a bag-level classification model. It assumes that there is one sentence describing the relation in a bag. It chooses the most correct sentence in each bag.
- **CNN+ATT** (Lin et al. 2016) is also a bag-level model, similar to CNN+Max. It adopts a sentence-level attention over the sentences in a bag and thus can down weight noisy sentences in a bag.

CNN is a sentence-level model that is trained directly on noisy data. For bag-level models (CNN+Max and CNN+ATT), the training process is the same as the referenced papers. During test, each sentence is treated as a bag and a relation is predicted for each bag. In this scenario, the bag-level relation prediction is exactly the same as the sentence-level prediction. All the baselines were implemented with the source codes released by (Li et al. 2016).

Results. Results in Table 1 reveal the following observations.

- CNN+RL obtains superior performance than CNN, indicating that filtering noisy data by instance selection benefits the task.
- CNN+RL outperforms CNN+Max and CNN+ATT remarkably. It shows the effectiveness of instance selection with reinforcement learning.
- The sentence-level models (CNN and CNN+RL) perform much better than the bag-level models (CNN+Max and CNN+ATT), indicating that bag-level models do not perform well for sentence-level prediction.

Instance Selection

We then evaluated the effectiveness of our instance selector from several aspects. First, we evaluated whether the selected data by our instance selector are better for relation classification. Second, we justified the accuracy of selection decision in the selector by manually checking the decisions on sentences. Third, we compared the proposed RL selection strategy in our selector with greedy selection. Last, we assessed whether the selector has the ability of filtering those bags that contain all noisy sentences.

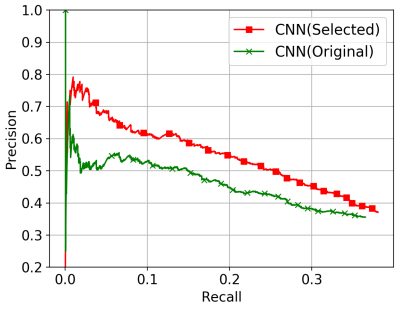

Figure 3: Comparison between the CNN model trained on the original and selected data.

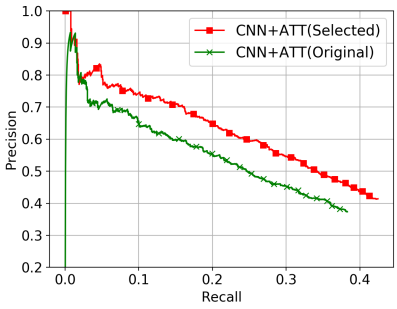

Figure 4: Comparison between the CNN+ATT model trained on the original and selected data.

Relation classification on selected data. To measure the quality of the selected data by our instance selector, we performed relation classification experiments on the selected data. We first used our instance selector to select the high-quality sentences from the original data. Then, we trained two state-of-the-art models, CNN and CNN+ATT with two settings. One setting is to train them on the original data, named as CNN(Original) and CNN+ATT(Original). The other setting is to train them on the selected data, which are named as CNN(Selected) and CNN+ATT(Seleted). We compared the performance of CNN(Original) (CNN+ATT(Original)) with CNN(Selected) (CNN+ATT(Selected)) on the relation classification task. The results are compared under the held-out evaluation configuration (Mintz et al. 2009) which provides an approximate measure of relation classification without expensive human annotations. The held-out evaluation compares the predicted relational fact from the test data with the facts in Freebase, but it does not consider the mapping between a relational fact and a sentence.

As shown in Figure 3 and Figure 4, the models trained on the selected data achieve much better performance than the counterparts trained on the original dataset. The results also indicate our instance selector has the ability of filtering out noisy sentences and distilling high-quality sentences, resulting better classification performance.

Accuracy of instance selection decision. To assess how

Bag I (**Entity Pair**: fabrice_santor, france; **Relation**:/people/person/nationality)	CNN+RL	CNN+ATT	CNN+Max
though not without some struggle, federer, the world 's top-ranked player, advanced to the fourth round with a thrilling, victory over the crafty **fabrice_santoro** of **france**, who is ranked 76th.	1	0.60	0
in his quarterfinal , nalbandian overwhelmed unseeded **fabrice_santoro** of **france**	1	0.39	1
fabrice_santoro, 33 , of **france** finally reached the quarterfinals in a major on his 54th attempt by defeating the 11th-seeded spaniard david ferrer	1	0.01	0
Bag II (**Entity Pair**: jonathan_littel, france; **Relation**:/people/person/nationality)			
jonathan_littell, a new york-born writer whose french-language novel about a murderous and degenerate officer has been the sensation of the french publishing season, on monday became the first american to win **france**'s most prestigious literary award, the prix goncourt	0	0.89	1
after a languid intercontinental auction that stretched for more than a week, the american rights to **jonathan_littell**'s novel les bienveillantes, which became a publishing sensation in **france**, have been sold to harpercollins, the publisher confirmed yesterday.	0	0.11	0

Table 2: Instance selection examples by different models. For CNN+RL and CNN+Max, 1 or 0 means the sentence is selected or not. For CNN+ATT, the value is the attention weight.

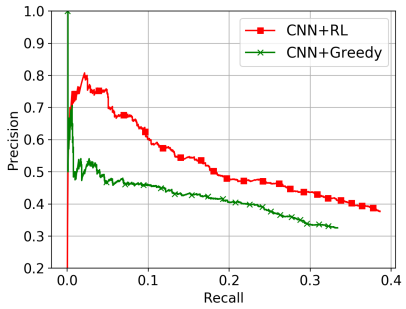

Figure 5: Comparison of instance selection with reinforcement learning against greedy selection.

accurate the decision is by the instance selector, we manually checked each sentence selected and rejected by the instance selector in a sampled dataset. For each sentence, the instance selector makes a correct decision if the sentence's label is correct and our instance selector selects it as a training instance, or, if its label is wrong and our instance selector rejects it. Otherwise, we judged that the instance selector makes a wrong decision.

Specifically, we sampled 300 sentences from the training data. Our instance selector chooses 64 sentences as the training instances, among which 45 sentences are correctly selected. The selector also rejects 236 instances, and 177 of them are noisy instances (not describing the relation). To summarize, the accuracy of our instance selector is $(45 + 177)/300 = 74\%$, which demonstrates the effectiveness of our instance selector.

Different instance selection strategies. To show the necessity for adopting the reinforcement learning framework for instance selection, we compared two instance selection strategies. Specifically, we performed relation classification on the selected data respectively with reinforcement learning (RL) selection and with greedy selection. The greedy selection selects the top N sentences with the largest likelihood which is estimated by a pre-trained CNN.

During the experiments, we kept the relation classifier untouched while replacing the RL selection by greedy selection. The number of selected instances N is the same as the RL strategy, where N is equal to $219,923$ (from $522,611$ original training sentences). As shown in Figure 5, the performance of our instance selector is much better than the greedy strategy on the held-out evaluation. The results show that our RL strategy is reasonable and effective.

Noisy bag filtering. As previous methods cannot filter the bags with all noisy sentences, we validated the ability of our model to filter bags with all noisy sentences. We randomly selected 100 deleted sentence bags and find that 86% of the bags consist of all noisy sentences. This indicates that our instance selector can exclude the noisy sentences effectively.

Case Study

Table 2 shows two bag examples for instance selection. The first bag has three correct sentences. The second bag has two noisy sentences. It is clearly show that our model can do better instance selection than both instance-weighting with CNN+ATT and maximum likelihood selection with CNN+Max. The second example indicates that our model is able to filter bags with all noisy sentences while other methods fail to do so.

Conclusion and Future Work

In this paper, we propose a novel model for sentence-level relation classification from noisy data using a reinforcement learning framework. The model consists of an instance selector and a relation classifier. The instance selector chooses high-quality data for the relation classifier. The relation classifier predicts relation at the sentence level and provides rewards to the selector as a weak signal to supervise the instance selection process. Extensive experiments demonstrate that our model can filter out the noisy sentences and perform sentence-level relation classification better than state-of-the-art baselines from noisy data.

Further, our solution for instance selection can be generalized to other tasks that employ noisy data or distant supervision. For instance, a possible attempt might be to perform sentiment classification on noisy data (Go, Bhayani, and Huang 2009). We leave this as our future work.

Acknowledgement

This work was partly supported by the National Science Foundation of China under grant No.61272227/61332007.

References

Bahdanau, D.; Brakel, P.; Xu, K.; Goyal, A.; Lowe, R.; Pineau, J.; Courville, A.; and Bengio, Y. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

dos Santos, C. N.; Xiang, B.; and Zhou, B. 2015. Classifying relations by ranking with convolutional neural networks. In *ACL*, 626–634.

Ebrahimi, J., and Dou, D. 2015. Chain based RNN for relation classification. In *NAACL*, 1244–1249.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter sentiment classification using distant supervision. *Cs224n Project Report*.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, 541–550. Association for Computational Linguistics.

Ji, G.; Liu, K.; He, S.; and Zhao, J. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *AAAI*, 3060–3066.

Li, J.; Monroe, W.; Ritter, A.; Jurafsky, D.; Galley, M.; and Gao, J. 2016. Deep reinforcement learning for dialogue generation. In *EMNLP*, 1192–1202.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *ACL*, volume 1, 2124–2133.

Liu, Y.; Wei, F.; Li, S.; Ji, H.; Zhou, M.; and Wang, H. 2015. A dependency-based neural network for relation classification. In *ACL*, 285–290.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *ACL*, 1003–1011. Association for Computational Linguistics.

Miwa, M., and Bansal, M. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*.

Mooney, R. J., and Bunescu, R. C. 2005. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, 171–178.

Narasimhan, K.; Yala, A.; and Barzilay, R. 2016. Improving information extraction by acquiring external evidence with reinforcement learning. *arXiv preprint arXiv:1603.07954*.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 148–163. Springer.

Sterckx, L.; Demeester, T.; Deleu, J.; and Develder, C. 2014. Using active learning and semantic clustering for noise reduction in distant supervision. In *4e Workshop on Automated Base Construction at NIPS2014 (AKBC-2014)*, 1–6.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP*, 455–465. Association for Computational Linguistics.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

Takamatsu, S.; Sato, I.; and Nakagawa, H. 2012. Reducing wrong labels in distant supervision for relation extraction. In *ACL*, 721–729. Association for Computational Linguistics.

Tianyu Liu, Kexiang Wang, B. C., and Sui, Z. 2017. A soft-label method for noise-tolerant distantly supervised relation extraction. In *EMNLP*, 17911796.

Wang, L.; Cao, Z.; de Melo, G.; and Liu, Z. 2016. Relation classification via multi-level attention cnns. In *ACL*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Xu, Y.; Mou, L.; Li, G.; Chen, Y.; Peng, H.; and Jin, Z. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*, 1785–1794.

Yang, Y.; Tong, Y.; Ma, S.; and Deng, Z. 2016. A position encoding convolutional neural network based on dependency tree for relation classification. In *EMNLP*, 65–74.

Zelenko, D.; Aone, C.; and Richardella, A. 2003. Kernel methods for relation extraction. *Journal of machine learning research* 3(Feb):1083–1106.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J.; et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, 2335–2344.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, 17–21.

Zhou, G.; Su, J.; Zhang, J.; and Zhang, M. 2005. Exploring various knowledge in relation extraction. In *ACL*, 427–434. Association for Computational Linguistics.