

Received August 1, 2019, accepted August 27, 2019, date of publication September 2, 2019, date of current version September 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938986

Joint Entity and Relation Extraction Based on Reinforcement Learning

XIN ZHOU¹, LUPING LIU^{ID1,2}, XIAODONG LUO¹, HAIQIANG CHEN³,
LINBO QING^{ID1,4}, (Member, IEEE), AND
XIAOHAI HE^{ID1}, (Member, IEEE)

¹College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China

²Key Laboratory of Network Assessment Technology, CAS (Institute of Information Engineering, Chinese Academy of Science), Beijing 100093, China

³China Information Technology Security Evaluation Center, Beijing 100085, China

⁴Key Laboratory of Wireless Power Transmission of Ministry of Education, Chengdu 610065, China

Corresponding authors: Linbo Qing (qing_lb@scu.edu.cn) and Xiaohai He (hxh@scu.edu.cn)

This work was supported in part by the Joint Funds of the National Natural Science Foundation of China under Grant U1836118, and in part by the Open Research Fund of Key Laboratory of Network Assessment Technology, Institute of Information Engineering, Chinese Academy of Sciences under Grant NST-18-001.

ABSTRACT Information extraction is a crucial technology to construct a knowledge base. In this paper, a novel model was proposed to extract entities and relations from plain text. This model consists of two components: a joint network and a reinforcement learning agent. The joint network is designed end-to-end, which can extract entities and relations simultaneously. In the joint model, a new tagging scheme was adopted, then the entity and relation extraction can be modeled as a joint sequence tagging problem. To enhance the robustness of the model, we also introduced a reinforcement learning (RL) agent to remove the noisy data from the training dataset. The RL agent aims at determining whether a candidate instance should be removed from the training dataset or not. When the agent completes a selection process, the training dataset will be divided into two parts: clean data and noisy data. Then the joint network can be trained again on the clean dataset to generate a better model. To assess the validity of the model we proposed, extensive experiments were conducted on the New York Times dataset (NYT10 and NYT 11). The experimental results showed that the model we proposed is superior compared with the baselines, achieving the F1 value on NYT10 and NYT11 with 0.612 and 0.549, respectively.

INDEX TERMS Information extraction, joint extraction, tagging scheme, reinforcement learning.

I. INTRODUCTION

Information extraction is a fundamental task in natural language processing (NLP), which can facilitate many other tasks, including knowledge base construction, question answering, and automatic text summarization. The goal of this task is to extract triplets (e_1, R, e_2) from the unstructured texts, where e_1 is the source entity, e_2 represents the object entity, and R is the semantic relation between e_1 and e_2 .

Many existing studies on information extraction adopted both supervised [1], [2] and unsupervised methods [3], [4]. The supervised methods train statistical and neural models for entities and relations extraction, where these methods need to create a large number of human-annotated datasets to train models. However, it is costly and nearly impossible for human annotators to go through a large

The associate editor coordinating the review of this article and approving it for publication was Yonghong Tian.

corpus of sentences to generate abundant labeled training dataset. To address this issue, Mintz et al. [5] proposed distant supervision (DS) method to generate large-scale training dataset for relation extraction automatically. Distant supervision method assumes that any sentence mentioning two entities expresses the same relationship as entity-pair expressed in the existed Knowledge Base (KB); then they align existed KBs with plain sentences to produce extensive training sentences. Although distant supervision method is effective and cheap, it inevitably introduces a large amount of noisy data (false positive). That is because the assumption is too strict, and the same entity pair may not express the desired relation types. The introduced noisy data can be divided into two types: (1) the entity pair mentioned in the sentences does not express the same relation type corresponding to the entities expressed in the KBs; (2) The target entity pair does not describe any relation type in the sentences.

TABLE 1. The examples of two types of noisy data existed in dataset.

Dataset	Relation	Sentence	Label	Correct
NYT10	/people/person	The traditional Republican bastions of Long Island crumbled as Lyndon B. Johnson was elected president, [Robert F. Kennedy] _{E1-place_lived} was elected a United States senator from [New York] _{E2-place_lived} , and both houses of the State Legislature were controlled by Democrats	/people/pers on/place_lived	
	/place_lived	If Hillary Clinton is re-elected to the Senate next fall and runs for president in 2008, she will be the first [New York] _{E2-place_lived} Democrat to make a serious bid for the White House since [Robert F. Kennedy] _{E1-place_lived} , who used the same Senate seat as his springboard 40 years earlier.	/people/pers on/place_lived	YES
NYT11	N/A	When I grew up I thought I was going to be in [Amityville] _{E1} for the rest of my life, " [Fraser] _{E2} said. " Contributions to AMIT, 817 Broadway, NYC 10003; or [New York Medical College] _{E1} , [Valhalla] _{E2} , NY 10595.	N/A	/

Taking Table 1 as an example, there are four sentences coming from the distant supervision dataset. The first two sentences come from the NYT10 [6] are labeled as the relationship “/people/person/place_lived”. In the first sentence, “*The traditional Republican bastions of Long Island crumbled as Lyndon B. Johnson was elected president, [Robert F. Kennedy] was elected a United States senator from [New York], and both houses of the State Legislature were controlled by Democrats*”, it can be seen clearly that the relation type “*place_lived*” is expressed by the preposition “from”. While the second sentence is noisy data as it does not express the same relation type. The last two sentences come from the NYT11 [7] are marked as negative samples because there is no relationship between the two entities. In reality, there is a large amount of noisy data existed in the training dataset generated by DS method. Therefore, we argue that a model’s performance will be affected severely if it is trained on a noisy dataset.

In this paper, we focused on combatting the noisy data in the training dataset and exploring the possibility of using dynamic selection strategies to remove them. More specifically speaking, a robust model was introduced to extract entities and novel model consists of two parts: a joint network and an RL agent. The joint network is an end-to-end model which produces entities and relations simultaneously based on a new tagging scheme. What’s more, a named entity recognition layer was added to the joint network to improve the performance of the model. The RL agent is based on reinforcement learning, where a performance-driven and policy-based reinforcement learning agent was designed to recognize the noisy sentences and take appropriate actions to remove them. During the training, the RL agent interacts with the joint model continuously and divides the training dataset into two parts: clean dataset and noisy dataset. In each interaction, the RL agent updates its policy network based on the reward received from the joint model. The interaction process will come to an end when the joint model trained on the clean dataset achieves the best performance.

Recently, with the advancement of reinforcement learning in the field of natural language processing, reinforcement learning has been adopted by many previous studies for relation extraction. Feng et al. [8] used a reinforcement learning agent to remove noisy data from the training dataset, and they used the likelihood of the sentence as a reward to guide the agent’s training. However, the reward in this study is derived from the performance change of the joint network for different epochs. Qin et al. [9] used reinforcement learning strategy to generate the false-positive indicator. They also used the performance change of the classifier as a reward. Unlike their research aiming to generate a positive-false indicator, this paper focuses on enhancing the robustness of the joint model. Feng et al. [10] modeled the joint extraction as a two-step decision process and employed the Q-Learning algorithm to get the control policy network in the two-step decision process. Also, Takanobu et al. [11] modeled relation extraction as two levels of the reinforcement learning process and use a hierarchical reinforcement learning framework to extract relations and entities. Different from the two studies mentioned above, this study models the removal of noisy data as a reinforcement learning process.

In summary, the contributions of this paper are as follows:

- 1) A novel end-to-end network was proposed to extract entities and relations jointly. A named entity recognition layer was added to the network, which enables the joint network to acquire better performance.
- 2) A reinforcement learning agent was proposed to obtain the ability to remove noisy data in the training dataset, making the joint model trained on the clean dataset more robust.
- 3) Extensive experiments were conducted on the public datasets (NYT10 and NYT11). The experimental results demonstrated that the proposed method could achieve much better performance than baselines.

The remainder of this paper is structured as follows. In section 2, related work associated with this field is described. Section 3 illustrates the detailed structure of the model proposed in this paper. In section 4, the related

experiments and results are presented. In section 5, the conclusion and limitation are made.

II. RELATED WORK

There are three main types of methods that are widely used in the joint extraction of entities and relations: pipeline method, joint entity and relation extraction method, and the end-to-end method.

The pipeline method extracts the token spans in the text to detect entity pair first and then predicts the relation type between them. Therefore, we describe some methods for named entity recognition (NER) and relation classification (RC). Many traditional NER models are linear sequence models, such as Maximum Entropy Markov Models (MEMM) [12], Conditional Random Fields (CRF) [13] and Hidden Markov Models (HMM) [14]. These models heavily depend on the appropriate features which often require much manual feature engineering efforts or external resources. Recently, deep learning methods have been proposed for NER, which model this task as a sequence tagging problem. Some researches apply Recurrent Neural Network (RNN) (e.g., Long Short-Term Memory (LSTM) or Bidirectional Long Short-Term Memory (Bi-LSTM)) based models with CRF to NER task [15], [16], while some others use Convolutional Neural Networks (CNN) based models with CRF [17] to do this task. The methods based on the neural network are more robust as they are less dependent on manual features or domain resources. The relation extraction is often modeled as a classification task, where the model is designed to predict the relation type between the entity pairs. The methods for relation classification can also be divided into two categories: feature-based methods [18], [19], and neural-based methods [20]–[23]. The feature-based methods first capture semantic features from the sentences and then apply a machine learning classifier to determine the relation types of the sentences. The neural-based methods employ neural networks to extract features from the sentences automatically and represent the sentences with feature vectors. Then the feature vectors are fed to a classifier to predict the relation types. Though the pipeline systems are flexible to design, they often suffer from error propagation [22].

To solve this problem, the joint entity and relation extraction methods were proposed, which can extract entities and relations simultaneously. The joint extracting method embodies the information of entities and relations; then it can achieve better performance than many previous pipeline methods. Many traditional joint extracting methods are feature-based systems, where they depend on heavy feature engineering and require many manual efforts and domain expertise [24]–[28]. Recently, joint learning methods were investigated to reduce manual work. The joint learning methods share parameters in the encoding layer, and the final decision is made by two networks separately for detecting entities and relation types. Miwa and Bansal [29] proposed an end-to-end LSTM based sequence and tree-structured

model that shares parameters for entity extraction and relation classification. Zheng et al. [22] designed a novel bidirectional encoder-decoder model which encodes the input sentence through Bi-LSTM, and two decoders were designed for entity extraction and relation classification. Bekoulis et al. [30] proposed a joint neural model which models the entity recognition task using a CRF (Conditional Random Fields) layer and the relation extraction task as a multi-head selection problem.

The end-to-end method is a new way to extract entities and relations in the same decoding network. Zheng et al. [31] designed a tagging scheme where each word is assigned a unique tag containing both entity and relation types. With this tagging scheme, the extraction of entities and relations can be done in the same network. To deal with overlapping relations, Wang et al. [3], [2] converted the joint task into a directed graph by designing a novel graph scheme and solved this issue by using a transition-based parsing framework.

III. METHOD

The overall architecture of the framework proposed in this paper is illustrated in Figure 1. The framework consists of two components: a joint network and an RL model. The Joint network is an end-to-end network used to extract entities and relations jointly. It is independent of the RL model. The role of the RL model is to remove noisy sentences from the original dataset and construct a clean dataset. Each sentence from the original dataset is checked by the agent's policy network. The agent takes actions to decide whether the candidate should be removed or not. After this, the agent will redistribute the training dataset into two parts: clean dataset and noisy dataset. Then the clean dataset is used to train and update the joint network, and the difference between the current and previous epoch on F_1 score is used to calculate the reward. Finally, the RL agent can update its policy network with the received reward and start a new selection process.

A. JOINT NETWORK

The joint network is composed of four layers: embedding layer, Bi-LSTM layer, named entity recognition layer, and joint decoding layer.

1) EMBEDDING LAYER

Given an instance s with T tokens, a common input representation of the joint layer is $[x_1, x_2, \dots, x_T]$, where $x_i \in R^d$ denotes the embedding of the token i , and d is the token embedding size. Particularly, In this study, we adopt the pre-trained word embedding Glove [33] and character embedding network to embed each token of the sentence.

The character representation is commonly applied to neural NER [16], [17]. It has been proved that character representation enables the model to achieve better performance on F_1 score in NER [30]. In this work, a neural network was applied to embed the word from the character level. The structure of the network is shown in Figure 2. Firstly,

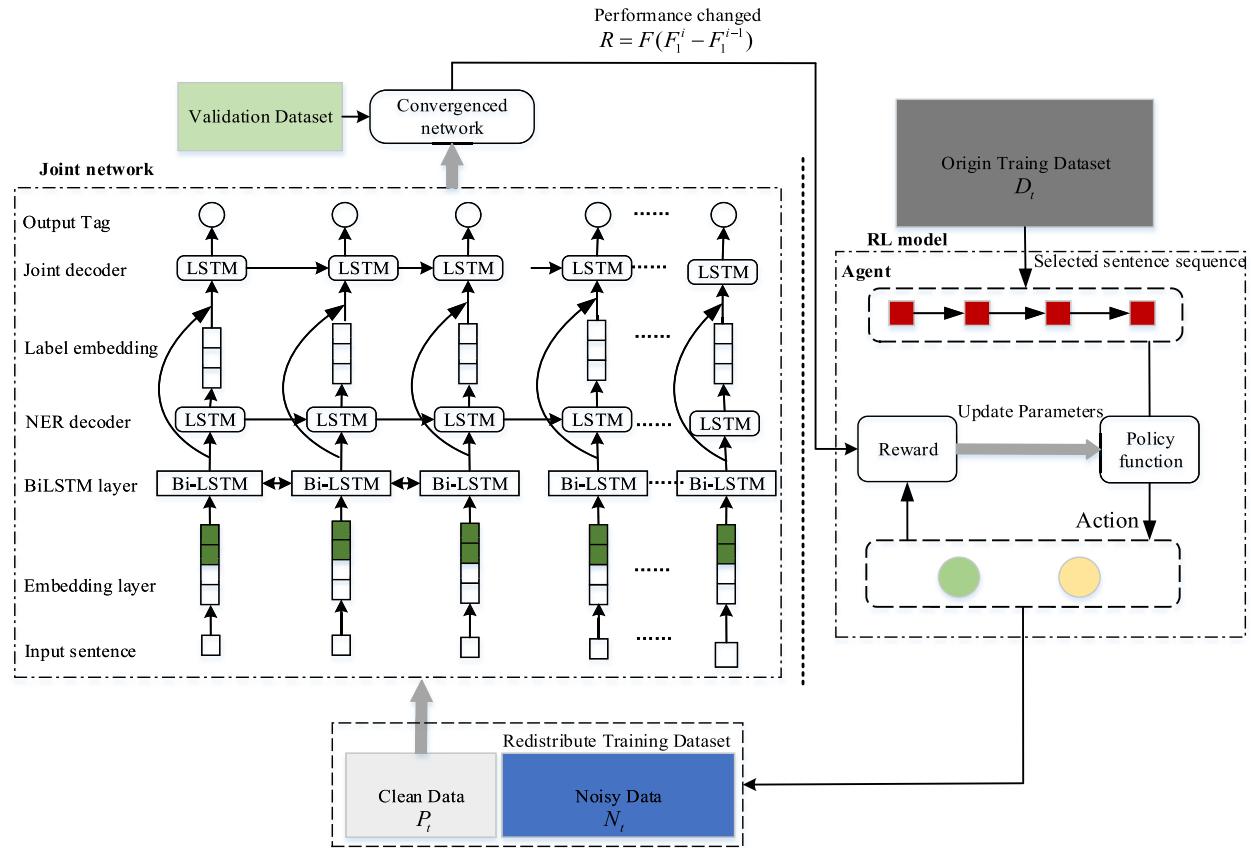


FIGURE 1. The main framework of our model. It is composed of two parts: A joint network and RL model. The joint network is used to extract entities and relations from the sentence, and it is trained on the clean training dataset. We use the F1 score as the criterion to evaluate the performance of the network. The RL model selects clean sentences from the training dataset and constructs clean dataset. Then the clean dataset will be feed to the joint network. The performance change of the joint network is treated as a reward to update the RL model's policy network.

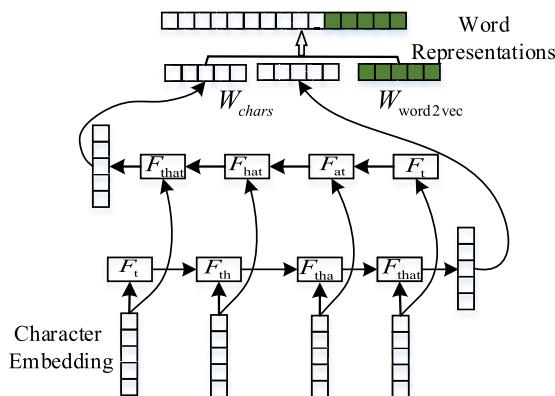


FIGURE 2. The architecture of the embedding layer. For a given word (i.e., *that*). Every character of the word is represented to a vector by embedding. All the vectors are feed to the Bi-LSTM layer. The state of the two directions (forward and backward) is concatenated to represent the character embedding. Finally, the character embedding and word embedding (Glove) are concatenated to represent the final embedding.

each word from the sentence is split into characters, and every character is represented by a vector (i.e., embedding). Then the vectors of all characters are feed into a Bi-LSTM network. The Bi-LSTM network captures the context information

of the words from two directions and generates forward and backward states. The two states are concatenated and used to represent the character embedding. Finally, the character embedding and word embedding are concatenated to represent the final word embedding. The detailed operation of the Bi-LSTM will be illustrated in the next section.

2) ENCODING LAYER

The encoding layer is used to capture the semantic information of the input sentence and generate hidden states. In this research, we employ a Bi-LSTM network to encode the input sequence. Once we have obtained the sentence's words representation $W = [w_1, w_2, \dots, w_n]$, where $w_t \in R^d$ is the d -dimensional word representation of the t -th word in the sentence and n is the length of the sentence. The sentence's words representation is fed to a Bi-LSTM network. The Bi-LSTM network consists two layers: forward LSTM layer and backward LSTM layer, and it can exploit the past (from the previous words) and future (from the next words) contextual information of the input sentence. For a given word embedding w_t in the sentence, two separate hidden states \overrightarrow{h}_t and \overleftarrow{h}_t are computed to represent the forward

and backward sequences respectively. The forward encoding \vec{h}_t captures the past information from w_0 to w_t , and the backward encoding \overleftarrow{h}_t captures the future information from w_n to w_t . The LSTM is a variant of the Recurrent Neural Network (RNN) which can avoid the gradient problem. The primary constituent unit of LSTM is a memory block that is composed of a cell (the memory part of the LSTM unit) and three “gates”. For the forward LSTM, at each time-step, a memory block computes the hidden vector \vec{h}_t based on three parts of information: the current word embedding w_t , the previous hidden vector h_{t-1} , and the previous cell vector C_{t-1} . The details of the operation can be formalized as follows.

$$i_t = \delta(W_i \cdot [w_t, h_{t-1}, C_{t-1}] + b_i) \quad (1)$$

$$f_t = \delta(W_f \cdot [w_t, h_{t-1}, C_{t-1}] + b_f) \quad (2)$$

$$z_t = \tanh(W_c \cdot [w_t, h_{t-1}, C_{t-1}] + b_c) \quad (3)$$

$$c_t = f_t * c_{t-1} + i_t * z_t \quad (4)$$

$$o_t = \delta(W_o \cdot [w_t, h_{t-1}, C_{t-1}] + b_o) \quad (5)$$

$$\vec{h}_t = o_t * \tanh(c_t) \quad (6)$$

where i, f, o represents the three gate separately: input gate, forget gate and output gate, $W_{(.)}$ is the parameter, b is the bias term, c is the cell memory, \cdot is the element-wise multiplication.

Finally, the forward encoding \vec{h}_t and backward encoding \overleftarrow{h}_t are concatenated together and marked as $h_t = [\vec{h}_t, \overleftarrow{h}_t]$, which can represent the encoding information of word embedding w_t .

3) NAMED ENTITY RECOGNITION LAYER

To make the joint network sensitive to all the entities, a named entity recognition layer was added to the joint model to detect all the entities of the input sentence. This layer does not produce the output directly but just feed the detected entities into the next layer.

We formulate the NER as a tagging scheme task and use BILOS (Beginning, Inside, Last, Outside, Single) tagging scheme to assign a tag for every token. By using the BILOS tagging scheme, a token can be identified by its positions and types (e.g., ORG). In this layer, we adopt an LSTM structure to decode the output of the Bi-LSTM layer. The process of the decoding is as follows.

$$h_d = LSTM(h_t) \quad (7)$$

where h_t is the output of Bi-LSTM. On the top of the LSTM layer, a full-connect layer is added to transform the hidden state h_d to the predicted vector score y_t :

$$y_t = W^{(e)} h_i + b^{(e)} \quad (8)$$

where $W^{(e)} \in R^{p \times 2d}$, $b^{(e)} \in R^p$ is the training parameters, and d is the hidden size of the LSTM. In the last layer, we use a

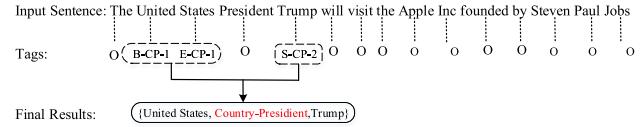


FIGURE 3. An example of the new tagging scheme.

softmax layer to compute the entity tag scores.

$$p_t^i = \text{soft max}(y_t) = \frac{e^{y_t^i}}{\sum_{j=1}^p e^{y_t^j}} \quad (9)$$

where p is the number of entities tags.

4) JOINT DECODING LAYER

In this study, a new tagging scheme was adopted to represent the output tag then we can model the relation extraction as a sequence tagging problem.

The tagging scheme was first proposed by Zheng et al. [31]. In this form of tagging scheme, each tag contains three parts of information, namely the position of the entity, the relation type, and the relation role. The entity position is represented in ‘BIEOS’ (Begin, Inside, End, Outside, Single) form. The relation type represents the relations between two entities, which is pre-defined. The relation role information represents that the entity belongs to the first entity (representing as number “1”) or the second entity (representing as number “2”). Figure 3 illustrates an example of the new tagging scheme. In this sentence, the first entity “United States” and the second entity “Trump” has the relation type “Country-President”. Therefore, the word “United” is tagged as “B-CP-1”, where “B” represents the entity position, “CP” represents the relation type, and number “1” means this word belongs to the first entity. Similarly, the word ‘States’ is tagged as “E-CP-1”. The second entity is tagged as “S-CP-2”, which means that the second entity consists of one word, and the relation type is “CP”. Finally, we can extract a triple (United States, Country-President, Trump) from this sentence. According to this tagging scheme, the total number of tags is $N_t = 2^4 * |R| + 1$, where $|R|$ the size of the predefined relation set.

Based on this tagging scheme, a LSTM architecture was adopted to predict the output tag as the NER layer does. The differences are the input and the output tag. The input consists of two parts: the output of the Bi-LSTM layer h_t and the entity tag embedding w_t . We use an embedding layer to encode the entity tag and concatenated the output with h_t . The input of joint decoding layer is formulated as $h_j = [h_t, w_t]$. During the training process, the entity tag is obtained directly from the input annotated labels. While for testing, the entity tag is obtained from the NER layer. For the output, a token will be assigned as a tag based on the new tagging scheme.

5) THE LOSS FUNCTION OF THE JOINT NETWORK

The loss of the joint model includes two parts: loss of the NER layer and loss of the joint decoding layer. We use

cross-entropy as the cost function for both of them, and the total loss can be formulated as follows:

$$J_T = J_{NEP} + J_{Joint} = -\frac{1}{|D|} \sum_{j=1}^{|D|} \log(p_t^j = y_t^{(j)} | x_j, \Theta_{ner}) \\ - \frac{1}{|D|} \sum_{j=1}^{|D|} \log(p_{jo}^j = y_{jo}^{(j)} | x_j, \Theta_{jo}) \quad (10)$$

During the training, the optimization method Adam proposed by Kingma et al. [34] was adopted to minimize the objective loss J_T .

B. RL MODEL

In this section, we first describe the definitions of RL model and then illustrate the pre-training and re-training strategies.

1) THE DEFINITIONS OF RL MODEL

In this study, we cast the noisy data removing as a reinforcement learning problem and aim at obtaining an agent that can take right actions to remove false positive sentences according to the state of the external environment. The agent obtained will interact with the external environment and updates its parameters based on the reward received from the joint network. The definitions of the RL model are elaborated as follows.

Agent: For the RL agent, it is formulated as a policy network $\pi_\theta(s, a) = p(a|s; \theta)$. The policy network is used to indicate whether a sentence should be removed or not according to the state. Thus, it is a binary classifier. In this studies, the policy network we adopted is a simple CNN model which was used as a text classifier in previous researches [35]. The agent can be written shortly as follows:

$$\pi(s; \theta) = CNN_{c_w, c_k}(s; \theta) \quad (11)$$

where c_w is the size of the windows and c_k is the size of the kernel. After the feature map was obtained, a max-over-time pooling operation was applied to capture the most import feature. Finally, these features are passed to a fully connected layer to generate the probability distribution over labels.

State: The state s_i includes the information of the current sentence and the sentences that have been removed in early states. To represent the state as a continuous vector, we utilize both word embedding and position embedding to convert the sentences into vectors, where the word embedding contains the Glove embedding and character embedding. With the sentence vectors, the current state is concatenated by the vector of the current input sentence and the averaged vector of the removed sentences.

Action: There are two actions for each agent: *retaining* (1) and *removing* (0). For each sentence of the training dataset, the agent takes an action a_i to determine whether it should be removed or not. The value of a_i is obtained from the policy network $\pi_\theta(s, a)$.

Reward: The goal of the RL model is to filter out the noisy data from the training dataset, resulting in better performance

of the joint network trained on the clean dataset. Therefore, the reward is calculated through the performances change of the joint network. We select the F_1 score as the evaluation criterion to reflect the comprehensive performance of the joint model. In this research, the reward is formulated as the difference of the F_1 score between the adjacent epochs:

$$R_i = F(F_1^i - F_1^{i-1}) = \begin{cases} 1, & F_1^i > F_1^{i-1} \\ 0, & F_1^i \leq F_1^{i-1} \end{cases} \quad (12)$$

In formula 12, if the F_1 score improves in step i , the agent will receive a reward with the value of 1. Otherwise, the agent will be fed a reward with a value of 0. In this paper, we use a simple set $\{0, 1\}$ as a reward because the reward is used to label the removing dataset. If the reward is 1, then the removed dataset is labeled as removing; otherwise, the removed dataset is labeled as re-training. To eliminate the randomness, the average F_1 value of the last five epochs were used to calculate the reward.

2) PRE-TRAINING STRATEGY

The pre-training procedure has been adopted by many RL models. It uses the annotated data to do supervised learning for the agent. In this research, the policy network is modeled as a CNN classifier. It takes the state vector as input and outputs a score, which can be used to determine whether the sentence should be removed or retained. During the pre-training procedure, a pre-trained dataset was constructed based on the original dataset to train the binary classifier first. We randomly select 80% proportional data from the original dataset and then label them with two parts: positive dataset (contains at least one relation) and negative dataset (does not express any relation in the sentence). During the pre-training, the cross-entropy was employed as the cost function to train the classifier, in which the negative label represents that the agent should take a removing action, and the positive labels represents that the agent should perform a re-training action.

$$J(\theta) = \sum_i \frac{(y_i \log[\pi(a = y_i | s_i; \theta)] + (1 - y_i) \log[1 - \pi(a = y_i | s_i; \theta)])}{(1 - y_i)} \quad (13)$$

3) RE-TRAINING AGENT WITH REWARDS

After the pre-training, the RL agent can remove the obvious negative samples from the original dataset. While it is difficult for the agent to identify some indistinguishable negative samples. Therefore, a re-training process is necessary to update the agent's policy network in order to enhance its capabilities. The re-training process is a continuous interaction with the joint network, which receives the reward from the joint network to guide the agent's training. Practically, the detail of the re-training process is depicted in Algorithm 1.

As manifested in Figure 1, the original training dataset D_t is composed of two parts: positive samples and negative samples. Before the re-training procedure, we assume that all the samples in the original training dataset are positive. Therefore, the positive dataset P_t^{ori} is equal to the original dataset D_t , and the negative dataset N_t^{ori} is initialized as zero.

Algorithm 1 Re-Training of the Agent

INPUT: Training dataset D_t , validation dataset D_d , the fixed number of removal m

- 1: Loading parameters θ from pre-trained policy network
- 2: Initialize s^* as zero vectors and P_{ori} as the training dataset D_t
- 3: **for** epoch $i = 1 \rightarrow N$ **do**
- 4: **for** $s_j \in P_t^{ori}$ **do**
- 5: $\bar{s}_j \in \text{concatenation}(s_j, s^*)$
- 6: compute action $a_j \sim \pi(a|\bar{s}_j; \theta)$; compute $p_j = \pi(0|\bar{s}_j; \theta)$;
- 7: **if** $a_j == 0$ **then**
- 8: Save tuple $t_j \sim (s_j, p_j)$ in T and compute the average vector of the removed sentence s^*
- 9: **end if**
- 10: **end for**
- 11: Rank T base on p_j from high to low, obtain T_{rank}
- 12: **for** t_i in $T_{rank}[: m] [:]$ **then**
- 13: Add $t_i [0]$ to Ω_t
- 14: **end for**
- 15: $P_t^{ori} = P_t^{ori} - \Omega_t$, $N_t^{ori} = N_t^{ori} + \Omega_t$
- 16: Train the joint model on P_t^{ori}
- 17: Calculate F_1^i on the validation set D_d
- 18: $R_i = F(F_1^i - F_1^{i-1})$
- 19: Update the θ : $g \propto \nabla_\theta \sum^{\Omega_i} (1 - R_i) \log[\pi(a|s; \theta)] + \sum^{\Omega_i} R_i \log[1 - \pi(a|s; \theta)]$
- 20: **end for**

The agent's job is to filter out the negative samples in the original dataset. During the re-training procedure, in each epoch, the agent removes a noisy set Ω_i from the original training dataset according to the policy $\pi(a|s)$. Because the set Ω_i is regarded as a negative dataset, then we obtain a new positive set and a negative sample dataset, equal to $P_t^{ori} = P_t^{ori} - \Omega_i$, $N_t^{ori} = N_t^{ori} + \Omega_i$ respectively. Then we utilize positive data P_t^{ori} to train the joint network. When the joint network is trained to converge, we use the validation set D_d to measure the performance of the joint network. The F_1 score of the joint network is calculated from the joint network. The difference between current and previous epoch on the F_1 score is used to calculate the reward, which will be used to train a more robust agent.

During the re-training, a fixed number of sentences with the lowest scores will be filtered out from the training dataset P_t^{ori} in every epoch based on the scores predicted by the RL agent. The quality of the RL agent is reflected by the performance change of the joint network. The remained clean parts P_t^{ori} in different epochs are the determinant of the change of F_1 score. If the F_1 score increases in epoch i , it means that the agent takes reasonable actions in epoch i to remove the negative samples. In other words, the removed dataset should be labeled as “0” which means to be removed. Conversely, if the F_1 score drops in epoch i , it reflects the agent takes unreasonable actions to removed data and the removed data should be labeled as “1”. Therefore, we use the removed dataset Ω_i to retrain the policy network in epoch i where the label of the dataset Ω_i comes from the reward received by the agent. If the agent receives a reward with a value of “1”, the dataset is labeled as “0”, and if the agent received a

reward with a value of “0”, the dataset is labeled as “1”. The loss function of the re-training process can be formulated as follows:

$$J(\theta) = - \sum^{\Omega_i} (1 - R) \log[\pi(a|s; \theta)] + R \log[1 - \pi(a|s; \theta)] \quad (14)$$

It is important to note that if the agent receives a reward with the value of 0, it states that the model has taken unreasonable actions. Therefore, we first retrain the agent with the removed data and then return the removed data to the original data.

IV. EXPERIMENTS**A. DATASETS AND METRICS**

Datasets: We evaluated the model on the New York Times (NYT) corpus, which is developed by distant supervision and contains noisy data. The dataset includes two versions: NYT10 [6] and NYT11 [7]. The NYT10 is the original version, which is generated by aligning the raw data with Freebase relation. NYT11 is a smaller version of which the testing set was manually annotated. The original NYT10 version contains more than 700,000 training sentences, but only a few of them contain relation types. If the original version is used to train the model, the cost will be intolerable. Therefore, we split some of the training data from the NYT10 to construct a new-sub dataset.

Individually, we randomly filter some noisy sentences, which does not contain any relation or the relation type does not exist in the testing dataset, from the original dataset. Finally, the remained dataset contains

TABLE 2. The statistics of the dataset.

Dataset	NYT10	NYT11
Relation types	29	23
Training sentences	206,989	235,645
Training relations	136,546	65,673
Testing sentences	4,006	369
Testing relations	4,006	369

TABLE 3. The hyper-parameters of the two networks.

Parameters	Parameters description	Parameters value
d_w	The dimension of the word embedding	300
d_c	The dimension of the char embedding	100
d_l	The dimension of the label embedding	300
h_c	The hidden units number of the Bi-LSTM layer	300
h_d	The hidden units number of the LSTM layer	300
c_w	The filter number of the windows for CNN model	5
c_k	The filter size of the kernel for CNN model	100
b_p	The batch size for training the joint model	64
l_r	The Learning rate	1e-3
$d_{dropjoint}$	The dropout for the joint model	0.5
$d_{dropenn}$	The dropout for the RL agent	0.5

206,989 sentences. Also, in this study, we focus on handling the single-relationship extraction task, where only one triplet will be extracted for every sentence. Therefore, we retain one relation type for each sentence of the training and testing dataset. The statistics of the dataset is illustrated in Table 2.

Evaluation Metrics: To evaluate the effectiveness of the model, the widely used metrics Precision (P), Recall (R), and F1-scores (F1) were adopted to measure the performance of the model. An extracted triplet is regarded as correct when the relation type and two entities are both correct.

B. EXPERIMENTAL SETTING

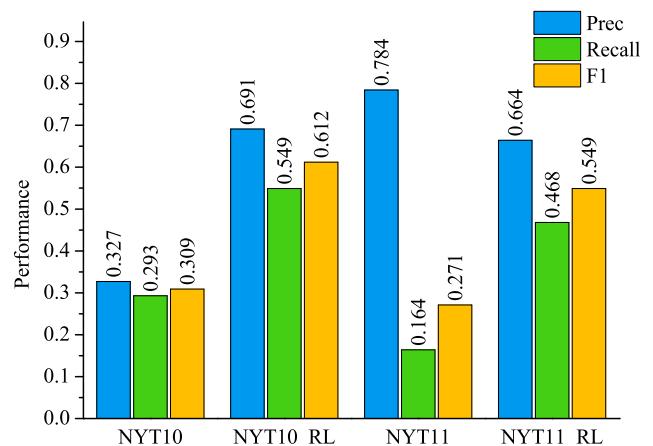
The proposed model consists of two parts: the joint network and RL model. The hyper-parameters of the two parts are illustrated in Table 3.

During the re-training of the RL agent, for each epoch, we select 480 sentences as input and remove a fixed number of the sentence in each epoch. For every epoch, if more than 240 sentences are judged to be removed, the agent filters out 240 sentences with the lowest score. While in another status, if less than 240 sentences are judged to be removed, all of them will be filtered out.

C. EXPERIMENTAL RESULTS

1) THE EFFECTIVENESS OF REINFORCEMENT LEARNING

The first experiment we conducted is to evaluate the effectiveness of the RL agent. Firstly, two models were trained on the two public original datasets (NYT10 and NYT11) and then were evaluated on the testing dataset. Secondly, we added an RL agent to remove the noisy data from the training dataset during the interaction with the joint model. Then two new models were trained on the remaining clean dataset and evaluated on the same testing dataset. During the

**FIGURE 4.** Performance comparison on the original and clean dataset (NYT10 and NYT11).

training, for each epoch, we need to train the joint model on the new dataset and wait for it to converge. At the beginning of the iteration, it takes about an hour for the joint model to complete a training process. Therefore, the whole training process is time-consuming. To speed up the training process, the experiments were conducted on a 4GPU (GeForce GTX 1080 Ti) server. On the dataset NYT10, the whole training process takes about 13 days. On the dataset NYT11, we spend about 15 days to remove the noisy data. Finally, more than 70,000 noise sentences were removed from NYT10 and more than 100,000 from NYT11.

The final experimental results are presented in Figure 4.

From the results, it can be seen that the performance of the models trained on the clean dataset acquired a significant improvement than that of the original dataset. On the NYT10 dataset, the model trained on the original dataset acquired the precision, recall, and F1 with the value of 0.327, 0.293, and 0.309, respectively. While for the model trained the clean dataset (NYT10_RL), significant improvement can be seen with its precision value 0.691, recall value 0.594, and F1 value 0.612. On the NYT11 dataset, although the precision value of the model trained on the clean data dropped, the recall and F1 value increase dramatically. The F1 value of the model trained on the clean dataset is almost twice over that of the model trained on the original dataset.

In table 4, two noisy sentences removed by the agent were given as examples to illustrate the effectiveness of the agent. The first sentence is incorrectly labeled by the DS method, and it does not express the relationship “/people/person/place_lived”. The second was labeled “N/A” as there is no relationship between the two entities. In reality, the agent can remove both of them correctly.

2) COMPARISON WITH BASELINES

The baselines in this research included four categories: pipeline methods, joint learning methods, tagging scheme methods, and hierarchical reinforcement learning-based method.

TABLE 4. Two example of noisy sentences removed by the agent.

Sentence	Label	Remove d
After the death of [Jackie McLean] <small>E1-place lived</small> , jazz saxophonist and educator, on March32, the jazz world turned its attention to [Hartford] <small>E2-place lived</small> .	/people/person/place_lived	YES
Over the last year [Chile] <small>E1</small> , Liberia, Germany and Jamaica have all sworn in their first female heads of government, and the [United States] <small>E2</small> is poised to do the same.	N/A	YES

TABLE 5. comparison with previous state-of-the-art methods. the first part (from row 1 to row 3) is the pipelined methods, the second part (from line3 to line7) is joint extraction methods, The third part is the end-to-end methods, the fourth part is the hierarchical reinforcement learning method and the last is the method in this study.

Method	NYT10			NYT11		
	Prec	Recall	F1	Prec	Recall	F1
FCM [Gormley et al., 2015]	-	-	-	0.553	0.154	0.240
LINE [Tang et al., 2015]				0.335	0.329	0.332
MultiR [Hoffmann et al., 2011]	-	-	-	0.338	0.327	0.333
DS-Joint [Li and Ji et al., 2014]	-	-	-	0.574	0.256	0.354
CoType [Ren et al., 2017]	-	-	-	0.495	0.376	0.427
SPTree [Li et al., 2014]	0.492	0.557	0.522	0.522	0.541	0.531
CopyR [Ren et al., 2017]	0.569	0.452	0.504	0.347	0.534	0.421
Bi-LSTM-Bias (Zheng et al., 2017)	0.593	0.381	0.464	0.615	0.414	0.495
Tagging-Graph[Wang et al., 2018]	-	-	-	0.643	0.421	0.509
HRL [Ryuichi et al., 2019]	0.714	0.586	0.644	0.538	0.538	0.538
JOINT_RL(NO_NER)	0.676	0.532	0.595	0.650	0.447	0.529
JOINT_RL	0.691	0.549	0.612	0.664	0.468	0.549

- 1) The chosen pipeline methods are FCM [36] and LINE [37]. FCM is a compositional embedding model by combining hand-crafted features with learned word embedding for relation extraction. LINE is a network embedding method which can embed very large information networks into low-dimensional vectors. Both of them obtain the NER results by CoType [38], and then the results are fed into the two models to predict the relation type.
- 2) The joint learning methods used in this research include feature-based methods (DS-Joint [24], MultiR [7], CoType [38], and neural-based methods (SPTree [27], CopyR [39]). DS-Joint is an incremental joint framework which extracts entities and relations based on structured perceptron and beam-search. MultiR is a joint extracting approach for multi-instance learning with overlapping relations. CoType extracts entities and relations by jointly embedding entity mentions, relation mentions, text features, and type labels into two meaningful representations. SPTree is a joint learning model that represents both word sequence and dependency tree structures using bidirectional sequential and tree-structured LSTM-RNNs. CopyR is a Seq2Seq learning framework with a copy mechanism for joint extraction.
- 3) When comparing with the tagging scheme method, we choose Bi-LSTM-Bias [31] and Tagging-Graph [32] as baselines. The Bi-LSTM-Bias gets the context representation of the input sentences through a Bi-LSTM network and uses an LSTM network to decode the tag sequences. The Tagging-Graph converts the joint

task into a directed graph by designing a novel graph scheme.

- 4) HRL is a hierarchical reinforcement learning framework which decomposes the whole extraction process into a hierarchy of two-level RL policies for relation extraction and entity extraction, respectively [11].

To illustrate that the NER layer is helpful to improve the performance of the model. We also conduct experiments to compare the performance of the JOINT_RL model with that of the JOINT_RL (NO_NER) which does not contain a NER layer.

The experimental results are presented in Table 5. Noticeably, the model's performance is quite different on the dataset NYT10 and NYT11. On the NYT10 dataset, the HRL and JOINT_RL achieved excellent results, and the performance is greatly improved compared with other baselines. The HRL achieved the best results, with an F1 value of 0.644, while our model came into second place with an F1 value of 0.612. The reason for this is that the testing dataset of NYT10 contains a large number of sentences which contain overlapping relationships (4,006 sentences contain 5859 relations, and we remain 4006 relations). HRL designed a hierarchical structure to address this issue, primarily. While the model proposed in this study focuses on the extraction of a single relationship for one sentence.

On the NYT11 dataset, there is almost no overlapping relation in the sentences (369 sentences contain 370 relations, and we just remain 369 relations). The JOINT_RL achieved the best performance and performed remarkably better than other baselines. The JOINT_RL model gains a F1 value

TABLE 6. The Output from the joint and joint_rl model. the first row for each example is the gold standard. the second row for each example is the result generated by the joint model. the third row for each example is the result generated by the joint_rl model.

Stand S1	For as long as [stephen harper] E1-nationality is prime minister of [Canada] E2-nationality , I vow to send him every two weeks, mailed on a Monday, a book that has been known to expand stillness.
JOINT	For as long as [stephen harper] E1-place_lived is prime minister of [Canada] E2-place_lived , I vow to send him every two weeks, mailed on a Monday, a book that has been known to expand stillness.
JOINT_RL	For as long as [stephen harper] E1-nationality is prime minister of [Canada] E2-nationality , I vow to send him every two weeks, mailed on a Monday, a book that has been known to expand stillness.
Stand S2	French critic and provocateur jean [baudrillard] E1-place_of_death , whose theories about consumer culture and the manufactured nature of reality were intensely discussed both in rarefied philosophical circles and in blockbuster movies like "the matrix," died yesterday in [Paris] E2-place_of_death .
JOINT	French critic and provocateur jean [baudrillard] E1-place_of_death , whose theories about consumer culture and the manufactured nature of reality were intensely discussed both in rarefied philosophical circles and in blockbuster movies like "the matrix," died yesterday in Paris.
JOINT_RL	French critic and provocateur jean [baudrillard] E1-place_of_death , whose theories about consumer culture and the manufactured nature of reality were intensely discussed both in rarefied philosophical circles and in blockbuster movies like "the matrix," died yesterday in [Paris] E2-place_of_death .
Stand S3	The proposal clearly rankled a co-founder of the [blackstone group] E1-founders , [stephen a. schwartzman] E2-founders , who was speaking at an awards presentation during a \$UNK\$ conference at the New York stock exchange yesterday.
JOINT	The proposal clearly rankled a co-founder of the [blackstone group] E1-founders , [Stephen] E2-founders a. Schwartzman, who was speaking at an awards presentation during a \$UNK\$ conference at the New York stock exchange yesterday.
JOINT_RL	The proposal clearly rankled a co-founder of the [blackstone group] E1-founders , [stephen a. schwartzman] E2-founders , who was speaking at an awards presentation during a \$UNK\$ conference at the New York stock exchange yesterday.

of 0.549 and has more than 1 percent improvement compared with the HRL. Results on NYT11 also show that neural-based models (SPTree, CopyR) and End-to-End model (Tagging) are more effective than traditional pipeline method (FCM) and feature-based method (MultiR, DS-joint, and CoType).

Comparing the results between the JORINT_RL (NO_NER) and JOINT_RL model, the performance of the JORINT_RL is better than the JORINT_RL (NO_NER). It means that the named entity recognition layer is helpful to improve the performance of the model.

3) LSTM VS CRF

In the previous joint extraction experiments based on the new tagging scheme [31], the LSTM decoder has been demonstrated to can achieve slightly better performance than CRF as LSTM is capable of learning long-term dependencies while CRF is good at capturing the joint probability of the entire tags. To analyze the effectiveness of the LSTM and CRF decoder, a comparative experiment has also been done. In this experiment, we replace the LSTM decoder with a CRF decoder and then train it on the same remain dataset. The results are manifested in Figure 5. As can be seen from the experimental results, the performance gap between the two decoders is small. The performance of the LSTM decoder is only a small improvement over the CRF decoder and can be ignored. Then we can use any of them to decode the output tags in this research.

4) CASE STUDY

In this subsection, some representative examples are given to illustrate the effectiveness of the RL model. The cases are shown in Table 6. Each case contains three rows, the gold results (Stand Si), the results produced by the joint model, and the results generated from the addition of the RL model. In each sentence, the blue part is the correct result, and the red

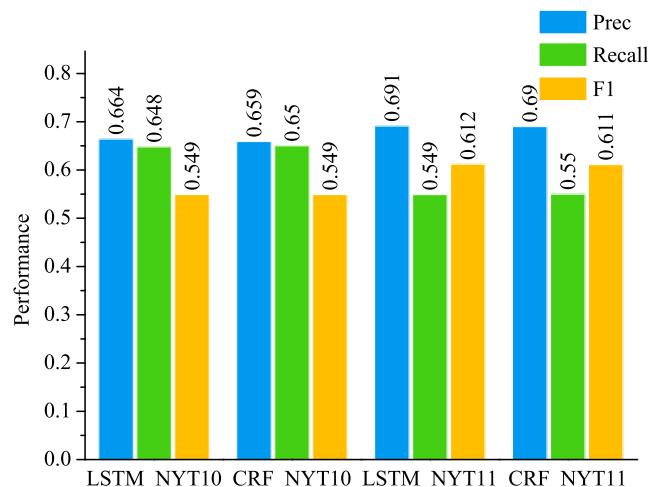


FIGURE 5. The performance of two decoders: LSTM and CRF. The pillars labeled with LSTM represents the results produced by the LSTM decoder. The pillars labeled with CRF correspond to the results generated by the CRF decoder.

part is the wrong one. In the sentence S1, the correct relationship between the two entities is "/people/person/nationality". The joint model trained on the original dataset can identify the two entities correctly, but the relationship between them is mispredicted while the joint model with the assistance of reinforcement learning can generate the triplet correctly. In the sentence S2, there is a long distance between the two entities. The model with RL can extract the two entities and their relation types correctly. While the joint model can predict the first entity and cannot find the second entity. In the sentence S3, the joint model can identify the first entity and the relation type of the sentence expressed, while the position of the second entity is not accurately identified. The model with RL can identify the position of the second entity accurately.

TABLE 7. The performance of the model for different elements.

Dataset	NYT10						NYT11					
Elements	E1			E2			E1			E2		
PRF	Prec	Recall	F1									
JOINT_RL	0.681	0.669	0.675	0.654	0.633	0.643	0.617	0.581	0.598	0.624	0.573	0.597
Elements	(E1,E2)			(E1,R,E2)			(E1,E2)			(E1,R,E2)		
PRF	Prec	Recall	F1									
JOINT_RL	0.795	0.629	0.703	0.693	0.548	0.612	0.664	0.468	0.549	0.650	0.469	0.545

D. ERROR ANALYSIS

In this subsection, we discuss the main factors which affect the performance of the end-to-end model. We analyzed the performance of the model to extract single elements, two entities, and triplets. The results are shown in Table 7. E1, E2 represents the model to extract the first entity and the second entity correctly. (E1, E2) represents that the model predicts the correct position of the first and second entity. (E1, R, E2) represents the model extract the triplets correctly where the extraction of the two entities and the relation are both correct.

From the results, we know that the precision of (E1, E2) is higher than both E1 and E2 individually, while the recall rate is lower than E1 and E2. It means that some of the entities identified by the model do not form a triplet. In some sentences, the model finds the first entity but does not catch the second entity or the model acquires the second entity but does not boast the first one. By comparing the result of (E1, E2) and (E1, R, E2), we find that the precision, recall rate and F1-score of the former is higher than the latter which means that the model can recognize both of the two entities but do not predict the relation or role correctly for some sentences.

V. CONCLUSION AND LIMITATION

In this paper, we aimed at handing the noisy data of the training dataset and proposed a novel model used to extract entities and relations jointly. The novel model is composed of two parts: the joint network and reinforce learning agent. A joint network was designed used to convert the joint extraction as a tagging scheme task based on a new tagging scheme. To make the joint decoder are more sensitive to all the entities and has higher performance, a name entity recognition layer was also added to the model to make the joint decoder are more sensitive to all the entities. To train a more robust joint network, a performance-driven agent was designed used to remove noisy data from the origin training dataset. Finally, the proposed model was evaluated on public datasets. The results showed that the proposed model can achieve better performance than others. The limitation of this work is that the model just extracts a single retaliation for one sentence and cannot handle overlapping relations. In the future, we would like design new network combined with RL model to solve this issue.

REFERENCES

- [1] R. J. Mooney and R. C. Bunescu, "Subsequence kernels for relation extraction," *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 171–178.
- [2] Z. GuoDong, S. Jian, Z. Jie, and Z. Min, "Exploring various knowledge in relation extraction," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, Jun. 2005, pp. 427–434.
- [3] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. Int. Conf. Comput. Linguistics*, 2014, pp. 2335–2344.
- [4] C. N. dos Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2015, pp. 626–634.
- [5] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. 4th Int. Joint Conf. Natural Lang. Process.*, Aug. 2009, pp. 1003–1011.
- [6] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Proc. Eur. Conf. Mach. Learn.*, 2010, pp. 148–163.
- [7] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jun. 2011, pp. 541–550.
- [8] J. Feng, M. Huang, L. Zhao, Y. Yang, and X. Zhu, "Reinforcement learning for relation classification from noisy data," in *Proc. Nat. Conf. Artif. Intell.*, Apr. 2018, pp. 5779–5786.
- [9] P. Qin, W. Xu, and W. Y. Wang, "Robust distant supervision relation extraction via deep reinforcement learning," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2018, pp. 2137–2147.
- [10] Y. Feng, H. Zhang, W. Hao, and G. Chen, "Joint extraction of entities and relations using reinforcement learning and deep learning," *Comput. Intell. Neurosci.*, vol. 2017, Aug. 2017, Art. no. 7643065.
- [11] R. Takanobu, T. Zhang, J. Liu, and M. Huang, "A hierarchical framework for relation extraction with reinforcement learning," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 7072–7079.
- [12] O. Bender, F. J. Och, and H. Ney, "Maximum entropy models for named entity recognition," in *Proc. 7th Conf. Natural Lang. Learn.*, Edmonton, AB, Canada, vol. 4, 2003, pp. 148–151.
- [13] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie, "Joint named entity recognition and disambiguation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 879–888.
- [14] A. Passos, V. Kumar, and A. McCallum, "Lexicon infused phrase embeddings for named entity resolution," in *Proc. 18th Conf. Comput. Natural Lang. Learn.*, Jun. 2014, pp. 78–86.
- [15] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," Aug. 2015, *arXiv:1508.01991*. [Online]. Available: <https://arxiv.org/abs/1508.01991>
- [16] G. Lample, M. Ballesteros, and S. Subramanian, "Neural architectures for named entity recognition," in *Proc. NAACL-HLT*, Jun. 2016, pp. 260–270.
- [17] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2016, pp. 1064–1074.
- [18] B. Rink and S. Harabagiu, "UTD: Classifying semantic relations by combining lexical and semantic resources," in *Proc. 5th Int. Workshop Semantic Eval.*, Jul. 2010, pp. 256–259.
- [19] N. Kambhatla, "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations," in *Proc. ACL Interact. Poster Demonstration Sessions*, Jul. 2004, Art. no. 22.
- [20] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 1785–1794.
- [21] S. Zheng, J. Xu, P. Zhou, H. Bao, Z. Qi, and B. Xu, "A neural network framework for relation extraction: Learning entity semantic and relation pattern," *Knowl.-Based Syst.*, vol. 114, no. 1, pp. 12–23, Dec. 2016.
- [22] S. Zheng, Y. Hao, D. Lu, H. Bao, J. Xu, H. Hao, and B. Xu, "Joint entity and relation extraction based on a hybrid neural network," *Neurocomputing*, vol. 257, pp. 59–66, Sep. 2017.

- [23] A. Katiyar and C. Cardie, "Going out on a limb: Joint extraction of entity mentions and relations without dependency trees," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2017, pp. 917–928.
- [24] X. Yu and W. Lam, "Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach," in *Proc. 23rd Int. Conf. Comput. Linguistics*, Aug. 2010, pp. 1399–1407.
- [25] B. Yang and C. Cardie, "Joint inference for fine-grained opinion extraction," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2013, pp. 1640–1649.
- [26] S. Singh, S. Riedel, B. Martin, J. Zheng, and A. McCallum, "Joint inference of entities, relations, and coreference," in *Proc. Workshop Automated Knowl. Base Construct.*, Oct. 2013, pp. 1–6.
- [27] Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, Jun. 2014, pp. 402–412.
- [28] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1858–1869.
- [29] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2016, pp. 1105–1116.
- [30] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Joint entity recognition and relation extraction as a multi-head selection problem," *Expert Syst. Appl.*, vol. 114, pp. 34–45, Dec. 2018.
- [31] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2017, pp. 1227–1236.
- [32] S. Wang, Y. Zhang, W. Che, and T. Liu, "Joint extraction of entities and relations based on a novel graph scheme," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4461–4467.
- [33] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1532–1543.
- [34] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2015, pp. 1–15.
- [35] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mai, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.
- [36] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.
- [37] M. R. Gormley, M. Yu, and M. Dredze, "Improved relation extraction with feature-rich compositional embedding models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 1774–1784.
- [38] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, and J. Han, "CoType: Joint extraction of typed entities and relations with knowledge bases," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 1015–1024.
- [39] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "Extracting relational facts by an end-to-end neural model with copy mechanism," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2018, pp. 506–514.



LUPING LIU was born in Sichuan, China, in 1988. He received the M.S. degree from Sichuan University, Chengdu, China, in 2015. He is currently pursuing the Ph.D. degree with the College of Electronics and Information Engineering. His current research interests include information extraction, knowledge, and intelligent question answering.



XIAODONG LUO was born in Sichuan, China, in 1981. He received the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2013. He is currently pursuing the Ph.D. degree with the College of Electronics and Information Engineering, Sichuan University. His current research interests include NLP and intelligent question answering.



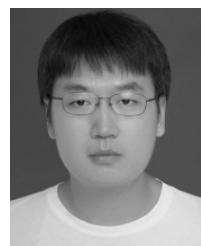
HAIQIANG CHEN received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2009. He is currently an Associate Research Fellow with the China Information Technology Security Evaluation Center, Beijing. His area of expertise revolves around data mining and nature language processing.



LINBO QING (M'16) received the B.S. degree in electronic information science and technology from Sichuan University, China, in 2003, and the Ph.D. degree in communication and information system from Sichuan University, where he is currently an Associate Professor with the College of Electronics and Information Engineering. His main research interests include image processing, visual computing, data mining, and video coding and transmission.



XIAOHAI HE (M'16) received the B.S. degree in electronics and the M.S. degree in electrical engineering from Sichuan University, Chengdu, China, in 1985 and 1991, respectively, and the Ph.D. degree in biomedical engineering from Sichuan University, where he is currently a Professor with the College of Electronics and Information Engineering. His research interests include image processing, pattern recognition, computer vision, image communication, and software engineering. He is a Senior Member of the Chinese Institute of Electronics. He is an Editor of the *Journal of Information and Electronic Engineering* and was an Editor of the *Journal of Data Acquisition and Processing*.



XIN ZHOU received the M.S. degree in software engineering from Sichuan University, Chengdu, China, in 2013, where he is currently pursuing the Ph.D. degree in communication and information systems. His current research interests include data mining, deep learning, and nature language processing.