Relabel the Noise: Joint Extraction of Entities and Relations via Cooperative Multiagents

Daoyuan Chen[1] **Yaliang Li**[1] **Kai Lei**[2] **Ying Shen**[3] *

[1]Alibaba Group

[2]School of Electronics and Computer Engineering, Peking University

[3]School of Intelligent Systems Engineering, Sun Yat-Sen University

[1]{daoyuanchen.cdy, yaliang.li}@alibaba-inc.com

[2]leik@pkusz.edu.cn, [3]sheny76@mail.sysu.edu.cn

Abstract

Distant supervision based methods for entity and relation extraction have received increasing popularity due to the fact that these methods require light human annotation efforts. In this paper, we consider the problem of *shifted label distribution*, which is caused by the inconsistency between the noisy-labeled training set subject to external knowledge graph and the human-annotated test set, and exacerbated by the pipelined entity-then-relation extraction manner with noise propagation. We propose a joint extraction approach to address this problem by re-labeling noisy instances with a group of cooperative multiagents. To handle noisy instances in a fine-grained manner, each agent in the cooperative group evaluates the instance by calculating a continuous confidence score from its own perspective; To leverage the correlations between these two extraction tasks, a confidence consensus module is designed to gather the wisdom of all agents and re-distribute the noisy training set with confidence-scored labels. Further, the confidences are used to adjust the training losses of extractors. Experimental results on two real-world datasets verify the benefits of re-labeling noisy instance, and show that the proposed model significantly outperforms the state-of-the-art entity and relation extraction methods.

1 Introduction

The extraction of entities and relations has long been recognized as an important task within natural language processing, as it facilitates text understanding. The goal of the extraction task is to identify entity mentions, assign predefined entity types, and extract their semantic relations from text corpora. For example, given a sentence *"Washington is the president of the United States of America"*, an extraction system will find a PRESIDENT_OF relation between PERSON entity "*Washington*" and COUNTRY entity "*United States of America*".

A major challenge of the entity and relation extraction task is the absence of large-scale and domain-specific labeled training data due to the expensive labeling efforts. One promising solution to address this challenge is distant supervision (DS) (Mintz et al., 2009; Hoffmann et al., 2011), which generates labeled training data automatically by aligning external knowledge graph (KG) to text corpus. Despite its effectiveness, the aligning process introduces many noisy labels that degrade the performance of extractors. To alleviate the introduced noise issue of DS, extensive studies have been performed, such as using probabilistic graphical models (Surdeanu et al., 2012), neural networks with attention (Zeng et al., 2015; Lin et al., 2016) and instance selector with reinforcement learning (RL) (Qin et al., 2018; Feng et al., 2018).

However, most existing works overlooked the **shifted label distribution** problem (Ye et al., 2019), which severely hinders the performance of DS-based extraction models. Specifically, there is a label distribution gap between DS-labeled training set and human-annotated test data, since two kinds of noisy labels are introduced and they are subject to the aligned KG: (1) False Positive: unrelated entity pair in the sentence while labeled as relations in KG; and (2) False Negative: related entity pair while neglected and labeled as NONE. Existing denoising works assign low weights to noisy instances or discard false positives while not recovering the original labels, leaving the shifted label distribution problem unsolved.

Moreover, most denoising works assume that the target entities have been extracted, i.e., the entity and relation extraction is processed in a pipe-lined manner. By extracting entities first and then classifying predefined relations, the entity extraction

*Corresponding author.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5940–5950
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

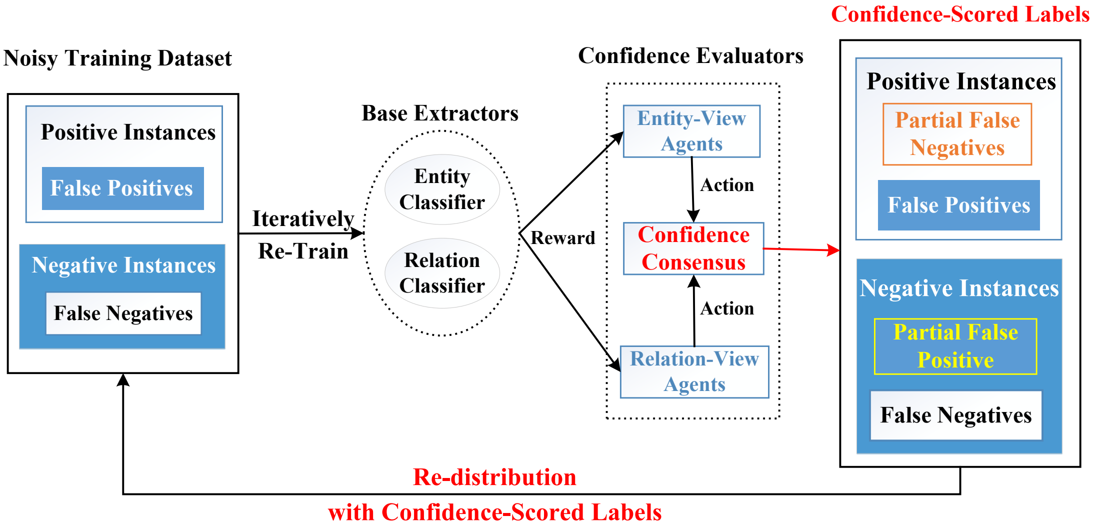

Figure 1: Overview of the proposed method. A group of multiagents are leveraged to evaluate the confidences of noisy instances from different extraction views. Base extractors are refined by iteratively training on the re-distributed instances with confidence-scored labels.

errors will be propagated to the relation extractor, introducing more noisy labels and exacerbating the shifted label problem. Besides, there are some correlations and complementary information between the two extraction tasks, which are under-utilized but can provide hints to reduce noises more precisely, e.g., it is unreasonable to predict two COUNTRY entities as the relation PRESIDENT_OF.

In this paper, to reduce the shifted label distribution gap and further enhance the DS-based extraction models, we propose a novel method to re-label the noisy training data and jointly extract entities and relations. Specifically, we incorporate RL to re-label noisy instances and iteratively re-train entity and relation extractors with adjusted labels, such that the labels can be corrected by trial and error. To leverage the correlations between the two extraction tasks, we train a group of cooperative multiagents to evaluate the instance confidence from different extraction views. Through a proposed confidence consensus module, the instances are re-labeled with confidence-scored labels, and such confidence information will be used to adjust the training loss of extractors. Finally, the performances of extractors are refined by exploring suitable label distributions with iterative re-training.

Empirical evaluations on two real-world datasets show that the proposed approach can effectively help existing extractors to achieve remarkable extraction performance with noisy labels, and the agent training is efficient with the help of correlations between these two extraction tasks.

2 Methodology

2.1 Overview

In this research, we aim to refine entity extractor and relation extractor trained with DS, by incorporating a group of cooperative multiagents. Formally, given a DS training corpus $D = \{s_1, \ldots, s_n\}$, an entity extractor θ'_e and a relation extractor θ'_r trained on D are input into the multiagents. The agents re-distribute D with confidence-scored labels and output two refined extractors θ^*_e and θ^*_r using the adjusted labels.

Towards this purpose, we model our problem as a decentralized multiagents RL problem, where each agent receives local environmental observation and takes action individually without inferring the policies of other agents. It is hard to directly evaluate the correctness of adjusted noisy labels since we do not know the "gold" training label distributions suitable to the test set. Nonetheless, we can apply RL to indirectly judge the re-labeling effect by using performance scores on an independent validation set as rewards, which is delayed over the extractor re-training. Further, the decentralization setting allows the interaction between the distinct information of entity and relation extractors via intermediate agents.

As shown in Figure 1, a group of agents acts as confidence evaluators, and the external environment consists of training instances and classification results of extractors. Each agent receives a private observation from the perspective of entity extractor or relation extractor, and makes an independent action to compute a confidence score of the instance. These actions (confidence scores) will then be considered together by the confidence consensus module, which determines whether the current sentence is positive or negative and assigns a confidence score. Finally, the updated confidences are used to retrain extractors, the performance score on validation set and the consistent score of the two extractors are combined into rewards for agents.

The proposed method can be regarded as a post-processing plugin for existing entity and relation extraction model. That is, we design a general framework of the states, actions and rewards by reusing the inputs and outputs of the extractors.

2.2 Confidence Evaluators as Agents

A group of cooperative multiagents are used to evaluate the confidence of each instance. These multiagents are divided into two subgroups, which act from the perspective of entity and relation respectively. There can be multiple agents in each subgroup for the purpose of scaling to larger observation space and action space for better performance. Next, we will detail the states, actions and rewards of these agents.

States The states S_e for entity-view agents and S_r for relation-view agents represent their own viewpoint to evaluate the instance confidence. Specifically, entity-view agents evaluate sentence confidence according to three kinds of information: current sentence, the entity extraction results (typed entity) and the noisy label types. Similarly, relation-view agents make their decisions depending on the current sentence, the relation types from relation extractor and the noisy label types from DS.

Most entity and relation extractors encode the semantic and syntactic information of extracted sentences into low-dimension embeddings as their inputs. For entity types and relation types, we also encode them into embeddings and some extractors have learned these vectors such as CoType (Ren et al., 2017). Given reused extractors, we denote the encoded sentence vector as $\mathbf{s}$, the extracted type vector as $\mathbf{t^e}$ and $\mathbf{t^r}$ for entity and relation respectively, and DS type vectors as $\mathbf{t_d^e}$ and $\mathbf{t_d^r}$ for entity and relation respectively. We reuse the sentence and type vectors of base extractors to make our approach lightweight and pluggable. Finally, we average the extracted and DS type embeddings to decrease the size of observation space, and concatenate them with the sentence embedding $\mathbf{s}$ to form the states $\mathbf{S^e}$ and $\mathbf{S^r}$ for entity/relation agents respectively as follows:

$$\mathbf{S^e} = \mathbf{s} \| (\mathbf{t^e} + \mathbf{t_d^e})/\mathbf{2}, \quad \mathbf{S^r} = \mathbf{s} \| (\mathbf{t^r} + \mathbf{t_d^r})/\mathbf{2}, \tag{1}$$

Note that we have encoded some semantics into the type vectors, e.g., the margin-based loss used in CoType enforces the type vectors are closer to their candidate type vectors than any other non-candidate types. Intuitively, in the representation spaces, the average operation leads in the midpoint of extracted type vector and DS type vector, which partially preserves the distance property among the two vectors and other type vectors, so that helps form distinguishable states.

Actions To assign confidence in a fine-grained manner and accelerate the learning procedure, we adopt a continuous action space. Each agent uses a neural policy network Θ to determine whether the current sentence is positive (conform with the extracted type t_i) or negative ("None" type) and computes a confidence score c. We model this action as a conditional probability prediction, i.e., estimate the probability as confidence given by the extracted type t_i and the current state $\mathbf{S}$: $c = p(positive|t_i, \Theta, \mathbf{S})$. We adopt gated recurrent unit (GRU) as policy network, which outputs the probability value using sigmoid function. A probability value (confidence score) which is close to 1/0 means that the agent votes a sentence as positive/negative with a high weight.

To handle huge state spaces (e.g., there are thousands of target types in our experimental dataset) and make our approach scalable, here we divide and conquer the state space by using more than one agent in entity-view and relation-view groups. The target type set is divided equally by agent number and each agent only is in charge of a part of types. Based on the allocation and DS labels, one sentence is evaluated by only one relation agent and two entity agents at a time, meanwhile, the other agents are masked.

Re-labeling with Confidence Consensus To leverage the wisdom of crowds, we design a consensus strategy for the evaluated confidences from multiagents. This is conducted by two steps: gather confidences and re-label with confidence score. Specifically, we calculate an averaged score as $\bar{c} = c_{sum}/3$, where c_{sum} is the sum of all agent confidences and the dividing means three agents evaluated the present sentence due to the above masking action strategy. Then we label the current sentence as negative (“None” type) with confidence $C = 1 - \bar{c}$ if $\bar{c} \leq 0.5$, otherwise we label the current sentence as positive (replace noisy label with extracted type) with confidence $C = \bar{c}$. This procedure can be regarded as weighted voting and re-distribute the training set with confidence-scored labels as shown in the right part of Figure 1, where some falsely labeled instances are put into intended positions or assigned with low confidences.

Rewards The reward of each agent is composed of two parts: shared global reward g expressing correlations among sub-tasks, and separate local rewards restricting the reward signals to different three agents for different sentences (recall that we evaluate each sentence by different agents w.r.t their responsible types). Specifically, the global reward g can give hints for denoising and here we adopt a general, translation-based triple score as used in TransE (Bordes et al., 2013) $g = ||\mathbf{t_1} + \mathbf{t_r} - \mathbf{t_2}||$, where $\mathbf{t_1}$, $\mathbf{t_r}$ and $\mathbf{t_2}$ are embeddings for triple (E_1, R, E_2) and pre-trained by TransE. The score is used to measure the semantic consistency of each triple and can be easily extended with many other KG embedding methods (Wang et al., 2017). As for the separate local reward, we use F1 scores F_1^e and F_1^r to reflect the extractor performance, which are gained by entity extractor and relation extractor on an independent validation dataset [1] respectively. Finally, to control the proportions of two-part rewards, we introduce a hyper-parameter α, which is shareable for ease of scaling to multiple agents as:

$$r_e = \alpha * F_1^e - g, \quad r_r = \alpha * F_1^r - g. \tag{2}$$

2.3 Model Learning

2.3.1 Loss Correction for Extractors

With the evaluated confidences and re-labeled instances, we adjust the training losses of entity extractor and relation extractor to alleviate the performance harm from noise and shifted label distribution. Denote the original loss of extractor as ℓ, the new loss ℓ' is adjusted by an exponential scaling factor λ and confidence C as : $\ell' = C^{\lambda}\ell$. Intuitively, a small confidence score C and a large λ indicate that the current instance has almost no impact on the model optimization. This can alleviate side-effects caused by noises and prevent the gradient being dominated by noisy labels, especially for those with divergent votes since the averaging in confidence consensus module leads to a small C.

Algorithm 1 Training Framework for Extractors

```
Input: Noisy training data $D$, pre-trained entity extractor $\theta'_e$, pre-trained relation extractor $\theta'_r$
Output: refined entity/relation extractor $\theta^*_e$, $\theta^*_r$
pre-train policy networks of agents based on $\theta'_e$ and $\theta'_r$
init: best $F1^*_e \leftarrow F1(\theta'_e)$, best $F1^*_r \leftarrow F1(\theta'_r)$
for epoch $i = 1 \rightarrow N$ do
    init: current extractors parameters $\theta_e \leftarrow \theta'_e$, $\theta_r \leftarrow \theta'_r$
    for batch $d_i \in D$ do
        extractors generate $\mathbf{S^e}/\mathbf{S^r}$ as Equ. (1)
        agents take actions (confidences)
        redistribute instances with confidences
        train $\theta_e/\theta_r$ with scaled losses $\ell'_e/\ell'_r$
        calculate rewards $r_e$ and $r_r$ as Equ. (2)
    end for
    if $F1(\theta_e) > F1^*_e$ then $F1^*_e \leftarrow F1(\theta_e), \theta^*_e \leftarrow \theta_e$
    if $F1(\theta_r) > F1^*_r$ then $F1^*_r \leftarrow F1(\theta_r), \theta^*_r \leftarrow \theta_r$
end for
```

2.3.2 Training Algorithm

Pre-training Many RL-based models introduce pre-training strategies to refine the agent training efficiency (Qin et al., 2018; Feng et al., 2018). In this study, we pre-train our models in two aspects: (1) we first pre-train entity and relation extractors to be refined as environment initialization, which is vital to provide reasonable agent states (embeddings of sentences and extracted types). (2) we then pre-train the policy networks of agents to gain a preliminary ability to evaluate confidence. In order to guide the instance confidence evaluation, we extract a small part of the valid data. The relatively clean DS type labels of the valid data are used to form states. The binary label is assigned according to the valid data and the policy networks are pre-trained for several epochs. Although the binary labels from valid data are not exactly the continuous confidence, the policy networks gain a better parameter initialization than random initialization by this approximate training strategy.

[1] To gain a relatively clean data, we randomly select 20% data from the original training set, extract them using pre-trained CoType model and retain only one instance for each sentence whose DS label is the same as the extracted label.

Iterative Re-training With the pre-trained extractors and policy networks, we retrain extractors and agents as Algorithm 1 detailed. The agents refine extractors in each epoch and we record parameters of extractors that achieve best F1 performance. For each data batch, entity and relation extractor perform extraction, form the states $\mathbf{S^e}$ and $\mathbf{S^r}$ as Equation (1), and send them to entity and relation agents respectively. Then agents take actions (evaluate confidences) and redistribute instance based on confidences consensus module (Section 2.2). Finally extractors are trained with confidences and give rewards as Equation (2).

Curriculum Learning for Multiagents It is difficult to learn from scratch for many RL agents. In this study, we extend the curriculum learning strategy (Bengio et al., 2009) to our cooperative multiagents. The motivation is that we can leverage the complementarity of the two tasks and enhance the agent exploration by smoothly increasing the policy difficulty. To be more specific, we maintain a priority queue and sample instances ordered by their reward values. Once the reward of current sentence excesses the training reward threshold $r_{threshold}$ or the queue is full, we then learn agents policies using Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm, which achieves good performances in many continuous control tasks. Algorithm 2 details the training procedure.

3 Experiments

3.1 Experimental Setup

Datasets We evaluate our approach on two public datasets used in many extraction studies (Pyysalo et al., 2007; Ling and Weld, 2012; Ren et al., 2017): **Wiki-KBP**: the training sentences are sampled from Wikipedia articles and the test set are manually annotated from 2013 KBP slot filling task; **BioInfer**: the dataset is sampled and manually annotated from biomedical paper abstracts. The two datasets vary in domains and scales of type set, detailed statistics are shown in Table 1.

Datasets	Wiki-KBP	BioInfer
#Relation / entity types	19 / 126	94 / 2,200
#Train M^r / M^e	148k / 247k	28k / 53k
#Test M^r / M^e	2,948 / 1,285	3,859 / 2,389

Table 1: Datasets statistics. M^r and M^e indicates relation and entity mentions respectively.

Algorithm 2 Curriculum Training with PPO for each Agent

```
Input: Data batch $d_i$, queue size $l$, pre-trained policy network with parameter $\Theta'$
Output: Policy network parameter $\Theta$
initialize an empty priority queue $q$ with size $l$
for sentence $s_j \in d_i$ do
    if $r_c > r_{threshold}$ or $q$ is full then
        run policy $\Theta'$ on environment $s_j$
        compute advantage estimate $\hat{A}$ using Generalized Advantage Estimator (GAE) (Schulman et al., 2015)
        optimize agent loss $\mathcal{L}$ (adaptive KL penalty form) w.r.t $\Theta$ using SGD
        $\Theta' \leftarrow \Theta$
        if $q$ is full then
            pull highest priority sentence
        end if
    else
        insert $s_j$ into $q$ with priority $r_c$
    end if
end for
```

Baselines For relation extraction, we compare with both pipe-lined methods and joint extraction methods: **MintZ** (Mintz et al., 2009) is a feature-based DS method using a logistic classifier; **MultiR** (Hoffmann et al., 2011) models noisy DS labels with multi-instance multi-label learning; **DS-Joint** (Li and Ji, 2014) jointly extracts entities and relations using structured perceptron; **FCM** (Gormley et al., 2015) introduces a neural model to learn linguistic compositional representations; **PCNN** (Zeng et al., 2015) is an effective relation extraction architecture with piece-wise convolution; **CoType** (Ren et al., 2017) is a state-of-the-art joint extraction method leveraging representation learning for both entity and relation types; **RRL-PCNN** (Qin et al., 2018) is a state-of-the-art RL-based method, which takes PCNN as base extractor and can also be a plugin to apply to different relation extractors; **ARNOR** (Jia et al., 2019) is a state-of-the-art de-noising method, which proposes attention regulation to learn relation patterns; **BA-fix-PCNN** (Ye et al., 2019) greatly improves the extraction performance by introducing 20% samples of the test set and estimate its label distribution to adjust the classifier of PCNN.

For entity extraction methods, we compare with a supervised type classification method, **HYENA** (Yosef et al., 2012); a heterogeneous partial-label

	Wiki-KBP			BioInfer		
Methods	**S-F1**	**Ma-F1**	**Mi-F1**	**S-F1**	**Ma-F1**	**Mi-F1**
HYENA	0.26	0.43	0.39	0.52	0.54	0.56
FIGER	0.29	0.56	0.54	0.69	0.71	0.71
WSABIE	0.35	0.55	0.50	0.64	0.66	0.65
PLE	0.37	0.57	0.53	0.70	0.71	0.72
CoType	0.39	0.61	0.57	0.74	0.76	0.75
MRL-CoType	**0.42**±7.2e-3	**0.64**±1.1e-2	**0.60**±8.3e-3	**0.77**±6.5e-3	**0.79**±1.3e-2	**0.78**±7.4e-3
(improvements)	(+7.69%)	(+4.92%)	(+5.26%)	(+4.05%)	(+3.95%)	(+4.00%)

Table 2: NER performance on two datasets, 3-time average results with standard deviations are reported.

	Wiki-KBP			BioInfer		
Methods	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
MintZ	0.296	0.387	0.335	0.572	0.255	0.353
MultiR	0.325	0.278	0.301	0.459	0.221	0.298
DS-Joint	0.444	0.043	0.078	0.584	0.001	0.002
FCM	0.151	**0.500**	0.301	0.535	0.168	0.255
ARNOR	0.453	0.338	0.407	0.589	0.382	0.477
BA-Fix-PCNN	0.457	0.341	0.409	0.587	0.384	0.478
RRL-PCNN	0.435	0.322	0.392	0.577	0.381	0.470
PCNN	0.423	0.310	0.371	0.573	0.369	0.461
MRL-PCNN	**0.461**±2.5e-3	0.325±2.3e-3	0.407±1.4e-3	0.590±1.1e-3	0.386±2.3e-3	0.483±2.8e-3
(improvements)	(+8.98%)	(+4.83%)	(+9.70%)	(+2.97%)	(+4.61%)	(+4.77%)
CoType	0.348	0.406	0.369	0.536	0.424	0.474
MRL-CoType	0.417±1.9e-3	0.415±1.6e-3	**0.416**±1.7e-3	**0.595**±2.1e-3	**0.437**±1.8e-3	**0.498**±2.0e-3
(improvements)	(+19.83%)	(+2.22%)	(+12.74%)	(+11.01%)	(+3.01%)	(+5.63%)

Table 3: End-to-end relation extraction performance, 3-time average results with standard deviations are reported.

embedding method, **PLE** (Ren et al., 2016); and two DS methods **FIGER** (Ling and Weld, 2012) and **WSABIE** (Yogatama et al., 2015).

Multiagents Setup To evaluate the ability of our approach to refine existing extractors, we choose two basic extractors for our **M**ultiagent **RL** approach, CoType and PCNN, and denote them as **MRL-CoType** and **MRL-PCNN** respectively. Since PCNN is a pipe-lined method, we reuse a pre-trained and fixed CoType entity extractor, and adopt PCNN as base relation extractor to adapt to the joint manner. For the CoType, we use the implementation of the original paper [2], and adopt the same sentence dimension, type dimension and hyper-parameters settings as reported in (Ren et al., 2017). For the PCNN, we set the number of kernel to be 230 and the window size to be 3. For the KG embeddings, we set the dimension to be 50 and pre-train them by TransE. We use Stochasitc Gradient Descent and learning rate scheduler with cosine annealing to optimize both the agents and extractors, the learning rate range and batch size is set to be [1e-4, 1e-2] and 64 respectively.

We implement our RL agents using a scalable RL library, RLlib (Liang et al., 2018), and adopt 2/8 relation agents and 2/16 entity agents for Wiki-KBP/BioInfer datasets respectively, according to their scales of type sets. For the multi-agents, due to the limitation of RL training time, we set the PPO parameters as default RLlib setting and perform preliminary grid searches for other parameters. For the PPO algorithm, we set the GAE lambda parameter to be 1.0, the initial coefficient for KL divergence to be 0.2. The loss adjusting factor λ is searched among $\{1, 2, 4\}$ and set to be 2, the reward control factors α is searched among $\{2\text{e-}1, 1, 2, 4\}$ and set to be 2. For all agents, the dimensions of GRU is searched among $\{32, 64\}$, and the setting as 64 achieved sightly better performance than setting as 32, while the larger dimension setting leads to higher memory overhead for each agent. Hence we set it to be 32 to enable a larger scale of the agents.

3.2 Effectiveness of Multiagents

3.2.1 Performance on Entity Extraction

We adopt the Macro-F1, Micro-F1 and Strict-F1 metrics (Ling and Weld, 2012) in the entity extraction evaluation. For Strict-F1, the entity prediction is considered to be "strictly" correct if and only if when the true set of entity tags is equal to the prediction set. The results are shown in Table 2 and we can see that our approach can effectively refine the base extractors and outperform all baseline

[2] https://github.com/INK-USC/DS-RelationExtraction

	Wiki-KBP			BioInfer		
Settings	**Precision(%)**	**Recall(%)**	**F1(%)**	**Precision(%)**	**Recall(%)**	**F1(%)**
Curriculum	41.7±0.19	41.5±0.16	41.6±0.17	59.5±0.21	43.7±0.18	49.8±0.20
Joint (w/o curriculum)	41.3±0.22	40.9±0.20	41.1±0.21	58.7±0.24	42.6±0.19	48.5±0.23
Separate (w/o joint)	38.8±0.24	40.5±0.27	38.4±0.25	54.7±0.27	41.3±0.23	47.6±0.26

Table 4: Ablation results of the MRL-CoType for end-to-end relation extraction.

methods on all metrics. Note that the refinements on BioInfer is significant (t-test with $p < 0.05$) even though the BioInfer has a large entity type set (2,200 types) and the base extractor CoType has achieved a high performance (0.74 S-F1), which shows that our agents are capable of leading entity extractors towards a better optimization with noisy.

3.2.2 Performance on Relation Extraction

Another comparison is the end-to-end relation extraction task, we report the precision, recall and F1 results in Table 3 and it illustrates that:

(1) Our method achieves best F1 for Wiki-KBP, outperforms all baselines on all metrics for BioInfer data, and significantly refines both the two base extractors, PCNN and CoType (t-test with $p < 0.05$), demonstrating the effectiveness of our approach.

(2) The improvements for CoType are larger than PCNN. Since CoType is a joint extraction model and leverages multi-agents better than the single-task extractor with fixed entity extractor. This shows the benefit of correlations between the two extraction tasks.

(3) Using the same base relation extractor, the MRL-PCNN achieves significantly better improvements than RRL-PCNN (t-test with $p < 0.05$). Besides, the precision of RRL-PCNN method is relatively worse than recall, which is mainly caused by the noise propagation of entity extraction and its binary discard-or-retain action. By contrast, our model achieves better and more balanced results by leveraging the cooperative multiagents with fine-grained confidences.

(4) The MRL-PCNN gains comparable performance with BA-Fix-PCNN, which leverages the additional information from the test set to adjust softmax classifier. This verifies the effectiveness and the robustness of the proposed RL-based relabeling method to reduce the shifted label distribution gap without knowing the test set.

3.3 Ablation Analysis

To evaluate the impact of curriculum learning strategy and joint learning strategy of our method, we compare three training settings: **curriculum learning**, standard training procedure as described in Section 2.3; **joint multiagents training** without curriculum learning (randomly sample training instances); and **separate training** without the participation of other agents using a pipeline manner, i.e., train an entity agent with only entity extractor and train a relation agent with only relation extractor.

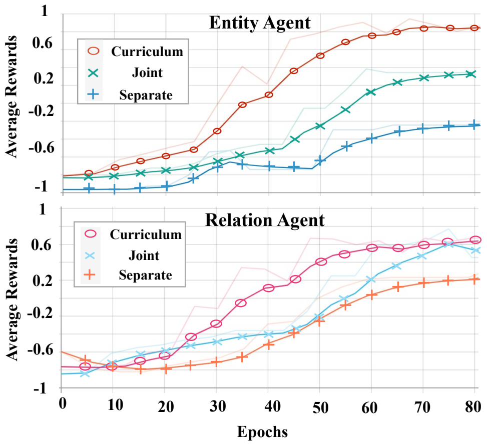

Figure 2: Smoothed average rewards on Wiki-KBP data for two agents of MRL-CoType. The light-colored lines are un-smoothed rewards.

The end-to-end relation extraction results are reported in Table 4. The curriculum setting and the joint setting achieve much better results than the separate training setting. This shows the superiority of cooperative multi-agents over single view extraction, which evaluates confidences with limited information. Besides, the curriculum setting achieves better results than the joint setting, especially on the BioInfer data, which has a larger type set and is more challenging than Wiki-KBP. This indicates the effectiveness of the curriculum learning strategy, which enhances the model ability to handle large state space with gradual exploration.

Training efficiency is an important issue for RL methods since the agents face the exploration-exploitation dilemma. We also compare the three settings from the view of model training. Figure 2 reports the average rewards for an entity agent and a relation agent on Wiki-KBP respectively. A high average reward indicates that the agent is trained

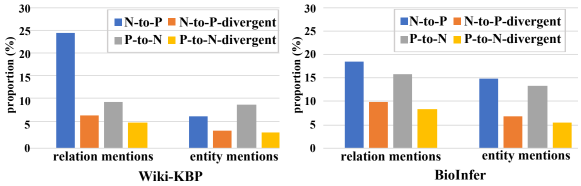

Figure 3: Proportions of re-labeled instances for MRL-CoType. "N-to-P" denotes the instances are re-labeled from negative to positive. "divergent" means that entity agents and relation agent have different evaluations about whether the instance is positive or negative.

effectively since it made valuable decisions and received positive feedback. From it we have the following observations: (1) The curriculum setting and the joint setting gain better performance than the separate training, which is consistent with the end-to-end extraction results. The improvement comes from the mutual enhancement among agents, since the correlations between the two tasks can restrict the reward signals to only those agents involved in the success or failure on the task; (2) The curriculum learning achieves higher rewards than the other two settings with fewer epochs, since that the convergence to local optimum can be accelerated by smoothly increasing the instance difficulty, and the multiagents provide a regularization effect.

3.4 Re-labeling Study

To gain insight into the proposed method, we conduct a statistic on the final re-labeled instances. Figure 3 reports the results and shows that our approach identifies some noisy instances including both positives and negatives, and leverage them in a fine-grained manner comparing with discard-or-retain strategy. Besides, the instances which are re-labeled from negatives to positives take a larger proportion than those with inverse re-labeling assignments, especially on Wiki-KBP data. This is in accordance with the fact that many noisy labels are "None" in DS setting. Note that some instances are re-labeled with divergent evaluations between entity-view and relation-view agents, which are usually get low confidences through the consensus module and have a small impact on the optimization with damping losses.

We further sample two sentences to illustrate the re-labeling processes. On Table 5, the first sentence has a noisy relation label **None**, while the relation extractor recognizes it as **country_of_birth** relation. Based on the extracted type, the relation-view agent evaluates it as a confidential positive instance due to the typical pattern "born in" in the sentence. The entity-view agents also evaluate it as positive with relatively lower confidences, and finally the sentence is re-labeled as positive by the consensus module. For the second sentence, agents disagree that it is positive. With the help of diverse extraction information, the consensus module re-labels the instance with low confidence score, and further alleviates the performance harm by loss damping.

4 Related Works

Many entity and relation extraction methods have been proposed with the pipelined fashion, i.e., perform named entity recognition (NER) first and then relation classification. Traditional NER systems usually focus on a few predefined types with supervised learning (Yosef et al., 2012). However, the expensive human annotation blocks the large-scale training data construction. Recently, several efforts on DS and weak supervision (WS) NER extraction have been made to address the training data bottleneck (Yogatama et al., 2015; Yang et al., 2018). For relation extraction, there are also many DS methods (Mintz et al., 2009; Min et al., 2013; Zeng et al., 2015; Han and Sun, 2016; Ji et al., 2017; Lei et al., 2018) and WS methods (Jiang, 2009; Ren et al., 2016; Deng et al., 2019) to address the limitation of supervised methods. Our method can be applied for a large number of those extractors as a post-processing plugin since the DS and WS usually incorporate many noises.

A recent work CrossWeigh (Wang et al., 2019) estimates the label mistakes and adjusts the weights of sentences in the NER benchmark CoNLL03. They focus on the noises of supervised "gold standard" labels while we focus on the noises of automatically constructed "silver standard" labels. Moreover, we deal with the noises by considering the shifted label distribution problem, which is overlooked by most existing DS works. In Ye et al. (2019), this issue is analyzed and authors improve performance significantly by using the distribution information from test set. In this paper, we propose to use RL to explore suitable label distributions by re-distributing the training set with confidence-scored labels, which is practical and robust to label distribution shift since we may not know the distribution of test set in real-world applications.

Another extraction manner is joint extraction, such as methods based on neural network with parameter sharing (Miwa and Bansal, 2016), represen-

Sentence 1, False Negative, Label: (*Bashardost*[/person], **None**, *Ghazni*[/location])	Entity Extractor	Relation Extractor	Entity Agents	Relation Agent	Confidence Consensus
Bashardost, an ethnic Hazara, was born in *Ghazni* province to a family of government employees.	*Bashardost*[/person] *Ghazni*[/location]	**country_of_birth**	0.772 0.729	0.896	Positive (0.799)
Sentence 2, False Positive, Label: (*profilin*[/**Protein**], **POS_ACTION_Physical**, *actin*[/Protein])					
Acanthamoeba *profilin* affects the mechanical properties of nonfilamentous *actin*.	*profilin*[/**None**] *actin*[/Protein]	**None**	0.373 0.791	0.236	Negative (0.533)

Table 5: Confidence evaluations on two noisy instances using MRL-CoType.

tation learning (Ren et al., 2017) and new tagging scheme (Zheng et al., 2017). However, these works perform extraction without explicitly handling the noises. Our approach introduces multiagents to the joint extraction task and explicitly model sentence confidences. As for the RL-based methods, in Zeng et al. (2018), RL agent is introduced as bag-level relation predictor. Qin et al. (2018) and Feng et al. (2018) use agent as instance selectors to discard noisy instances in sentence-level. Different from adopting a binary action strategy and only focus on false positives in these works, we adopt a continuous action space (confidence evaluation) and handle the noises in a fine-grained manner. The binary selection strategy is also adopted in a related study, Reinforced Co-Training (Wu et al., 2018), which uses an agent to select instances and help classifiers to form auto-labeled datasets. An important difference is that they select unlabeled instances while we evaluate noisy instances and relabel them. More recently, HRL (Takanobu et al., 2019) uses a hierarchical agent to first identifies relation indicators and then entities. Different from using one task-switching agent of this work, we leverage a group of multiagents, which can be a pluggable helper to existing extraction models.

5 Conclusions

To deal with the noise labels and accompanying shifted label distribution problem in distant supervision, in this paper, we propose a novel method to jointly extract entity and relation through a group of cooperative multiagents. To make full use of each instance, each agent evaluates the instance confidence from different views, and then a confidence consensus module is designed to re-label noisy instances with confidences. Thanks to the exploration of suitable label distribution by RL agents, the confidences are further used to adjust the training losses of extractors and the potential harm caused by noisy instances can be alleviated. To demonstrate the effectiveness of the proposed method, we evaluate it on two real-world datasets and the results confirm that the proposed method can significantly improve extractor performance and achieve effective learning.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61602013), and the Shenzhen General Research Project (No. JCYJ20190808182805919).

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning.*, pages 41–48.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems.*, pages 2787–2795.

Yang Deng, Yaliang Li, Ying Shen, Nan Du, Wei Fan, Min Yang, and Kai Lei. 2019. Medtruth: A semi-supervised approach to discovering knowledge condition information from multi-source medical data. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 719–728.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Matthew R Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. *Proceedings of the 2015 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1774—–1784.

Xianpei Han and Le Sun. 2016. Global distant supervision for relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 541–550.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. Arnor: Attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Jing Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, pages 1012–1020.

Kai Lei, Daoyuan Chen, Yaliang Li, Nan Du, Min Yang, Wei Fan, and Ying Shen. 2018. Cooperative denoising for distantly supervised relation extraction. In *Proceedings of the 27th International Conference on Computational Linguistics.*, pages 426–436.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 402–412.

Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. 2018. RLlib: Abstractions for distributed reinforcement learning. In *Proceedings of the 35th annual international conference on machine learning.*

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2124–2133.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, volume 12, pages 94–100.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1011.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1105—-1116.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):50.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining.*, pages 1825–1834.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *WWW*, pages 1015–1024.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5157–5166.

Jiawei Wu, Lei Li, and William Yang Wang. 2018. Reinforced co-training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1252–1262.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169.

Qinyuan Ye, Liyuan Liu, Maosen Zhang, and Xiang Ren. 2019. Looking beyond label noise: Shifted label distribution matters in distantly supervised relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3832–3841.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 291–296.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. *Proceedings of the 21th International Conference on Computational Linguistics*, pages 1361–1370.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1753–1762.

Xiangrong Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Large scaled relation extraction with reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1227–1236.