Distant Supervision for Relation Extraction

AntNLP -- Changzhi Sun

changzhisun@stu.ecnu.edu.cn

DSGAN: Generative Adversarial Training for Distant Supervision Relation Extraction

Pengda Qin[♯], Weiran Xu[♯], William Yang Wang[♭]
[♯]Beijing University of Posts and Telecommunications, China
[♭]University of California, Santa Barbara, USA
{qinpengda, xuweiran}@bupt.edu.cn
{william}@cs.ucsb.edu

Robust Distant Supervision Relation Extraction via Deep Reinforcement Learning

Pengda Qin[♯], Weiran Xu[♯], William Yang Wang[♭]
[♯]Beijing University of Posts and Telecommunications, China
[♭]University of California, Santa Barbara, USA
{qinpengda, xuweiran}@bupt.edu.cn
{william}@cs.ucsb.edu

Outline

1. Background
2. Adversarial Learning for DS
3. Experments & Conclusions
4. Reinforcement Learning for DS
5. Experments & Conclusions

Distant Supervision

- **Relation:** $< e_1, r, e_2 >$
- **Assumption:** If two entities have a relationship in a known knowledge base, then all sentences that mention these two entities will express that relationship in some way
- **Advantage:** does not require labeled corpora

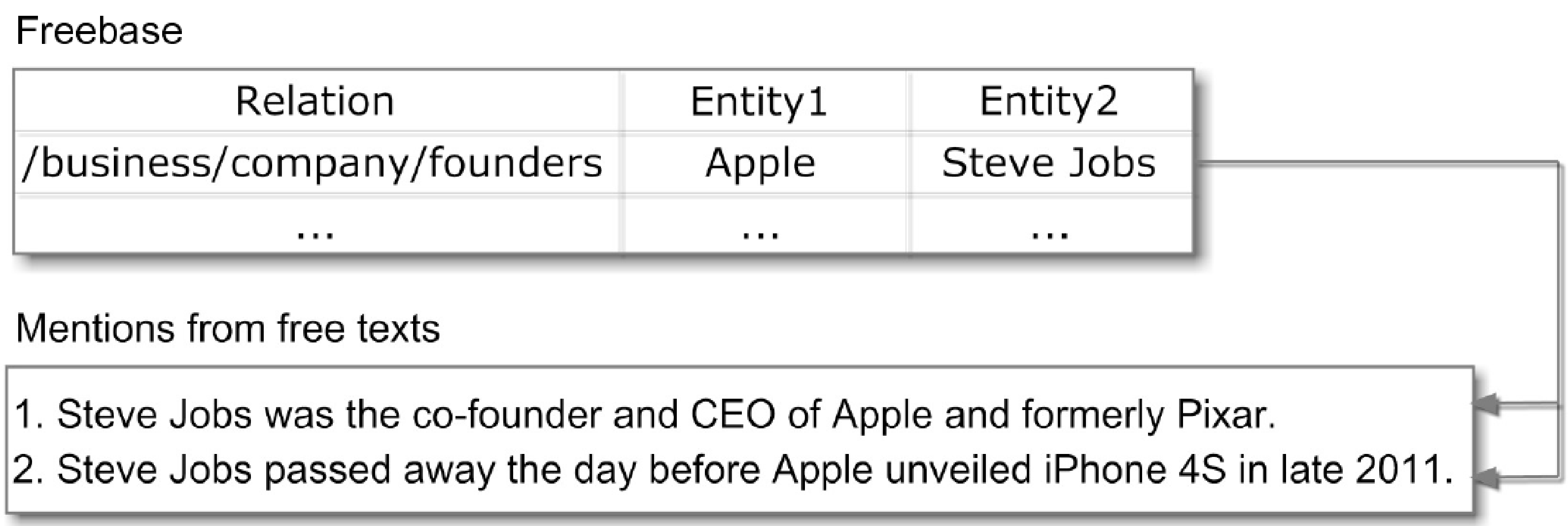

Figure 1: Training instances generated through distant supervision. Upper sentence: correct labeling; lower sentence: incorrect labeling.

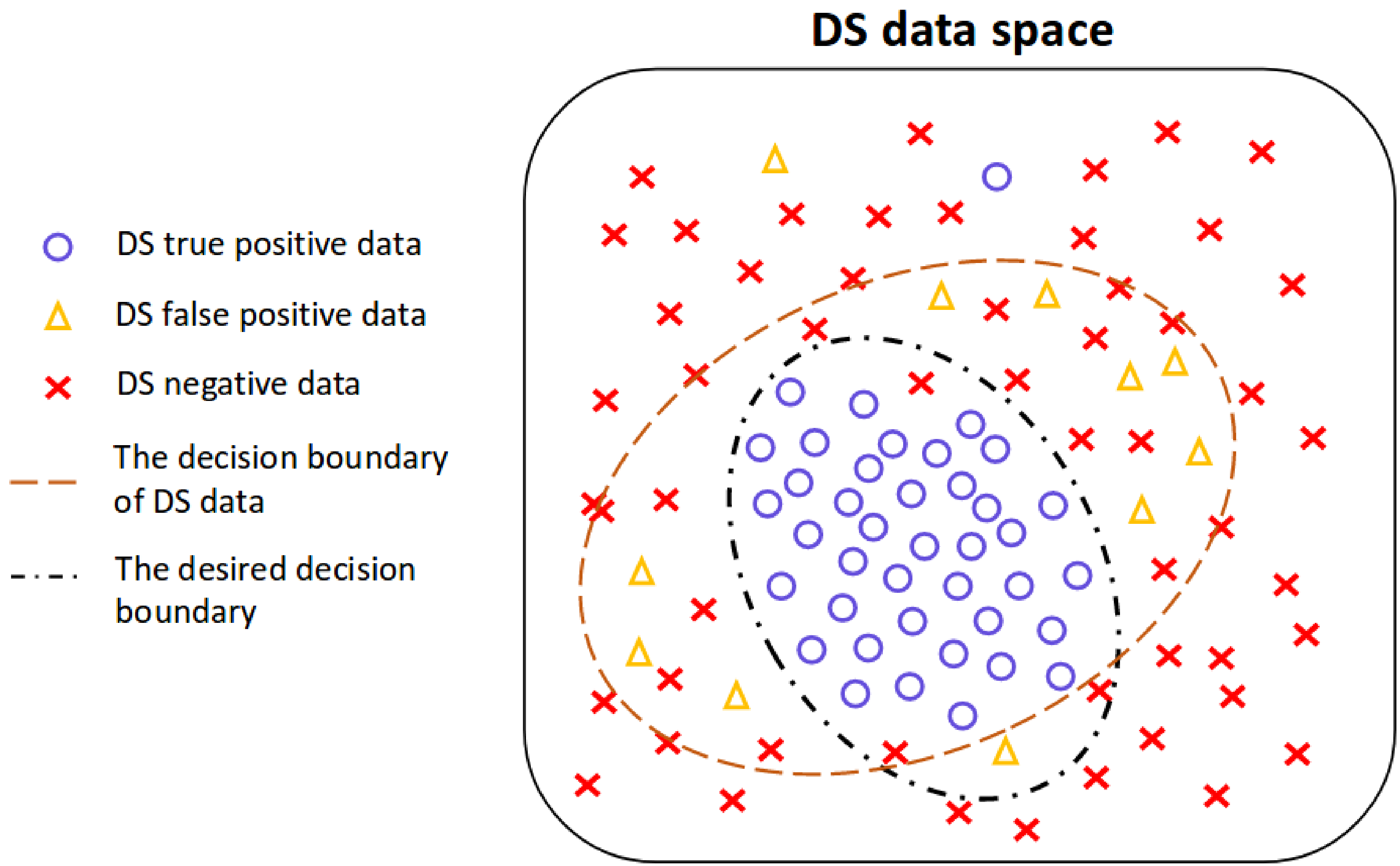

Figure 1: Illustration of the distant supervision training data distribution for one relation type.

Motivation

- the generator tries to generate true positive samples from DS positive dataset
- assign the generated samples with negative label and the rest samples with positive label to challenge the discriminator
- generated sample set includes more true positive samples and more false positive samples are left in the rest set
- the classification ability of the discriminator will drop faster

Pre-Training

- **discriminator**:DS positive dataset P and DS negative set N^D
  - accuracy reaches 90% or more
- **generator**: DS positive dataset P and DS negative set N^G
  - give high probabilities to all of the noisy DS positive samples
- $p_G(s_j)$ and $p_D(s_j)$: the probability of being true positive sample
- split $P = \{s_1, ..., s_j, ...\}$ into N bags $B = \{B^1, ..., B^N\}$

Generator

$$T = \{s_j\}, s_j \sim p_G(s_j), j = 1, 2, ..., |B_i|$$

$$F = B_i - T$$

- T: high-confidence sentences (negative for discriminator)
- F: low-confidence sentences (positive for discriminator)

$$L_G = \sum_{s_j \in T} \log p_D(s_j)$$

- policy-gradient-based

Discriminator

$$L_D = -(\sum_{s_j \in F} \log p_D(s_j) + \sum_{s_j \in T} \log(1 - p_D(s_j)))$$

- loads the `same pre-trained parameter` set at the beginning of each epoch
  - robust generator rather than a discriminator
  - generator is to sample data rather than generate new data from scratch, Therefore, discriminator is relatively easy to be collapsed
  - the robustest generator is yielded when the discriminator has the largest drop of performance in one epoch

Optimizing Generator

$$r_1 = \frac{1}{|T|} \sum_{s_j \in T} p_D(s_j) - b_1$$

$$\tilde{p} = \frac{1}{|N^D|} \sum_{s_j \in N^D} p_D(s_j)$$

$$r_2 = \eta(\tilde{p}_i^m - b_2), b_2 = \max(\tilde{p}_i^m), m = 1, ..., k-1$$

$$\nabla_{\theta_G} L_G = \sum_{s_j \in B_i} \mathbb{E}_{s_j \sim p_G(s_j)} r \nabla_{\theta_G} \log p_G(s_j)$$

$$\nabla_{\theta_G} L_G = \frac{1}{|T|} \sum_{s_j \in T} r \nabla_{\theta_G} \log p_G(s_j)$$

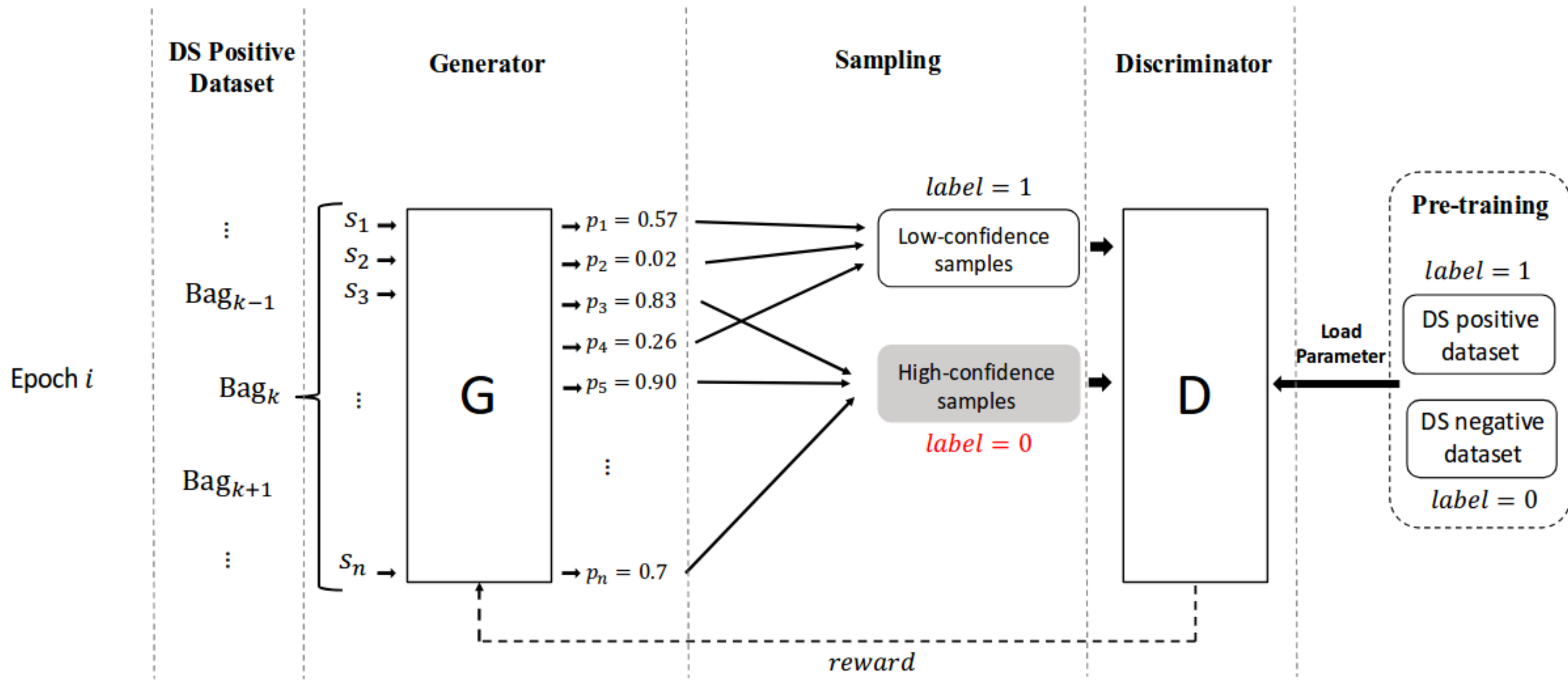

Figure 2: An overview of the DSGAN training pipeline. The generator (denoted by **G**) calculates the probability distribution over a bag of DS positive samples, and then samples according to this probability distribution. The high-confidence samples generated by G are regarded as true positive samples. The discriminator (denoted by **D**) receives these high-confidence samples but regards them as negative samples; conversely, the low-confidence samples are still treated as positive samples. For the generated samples, **G maximizes** the probability of being true positive; on the contrary, **D minimizes** this probability.

Algorithm 1 The DSGAN algorithm.

Data: DS positive set P, DS negative set N^G for generator G, DS negative set N^D for discriminator D

Input: Pre-trained G with parameters θ_G on dataset (P, N^G); Pre-trained D with parameters θ_D on dataset (P, N^D)

Output: Adversarially trained generator G

Load parameters θ_G for G
Split P into the bag sequence $P = \{B^1, B^2, ..., B^i, ..., B^N\}$
repeat
    Load parameters θ_D for D
    $G_G \leftarrow 0, G_D \leftarrow 0$
    for $B_i \in P, i = 1$ to N **do**
        Compute the probability $p_G(s_j)$ for each sentence s_j in B_i
        Obtain the generated part T by sampling according to $\{p_G(s_j)\}_{j=1...|B|}$ and the rest set $F = B_i - T$
        $G_D \leftarrow -\frac{1}{|P|}\{\bigtriangledown_{\theta_D} \sum^T \log(1 - p_D(s_j)) + \bigtriangledown_{\theta_D} \sum^F \log p_D(s_j)\}$
        $\theta_D \leftarrow \theta_D - \alpha_D G_D$
        Calculate the reward r
        $G_G \leftarrow \frac{1}{|T|} \sum^T r \bigtriangledown_{\theta_G} \log p_G(s_j)$
        $\theta_G \leftarrow \theta_G + \alpha_G G_G$
    end for
    Compute the accuracy ACC_D on N^D with the current θ_D
until ACC_D no longer drops
Save θ_G

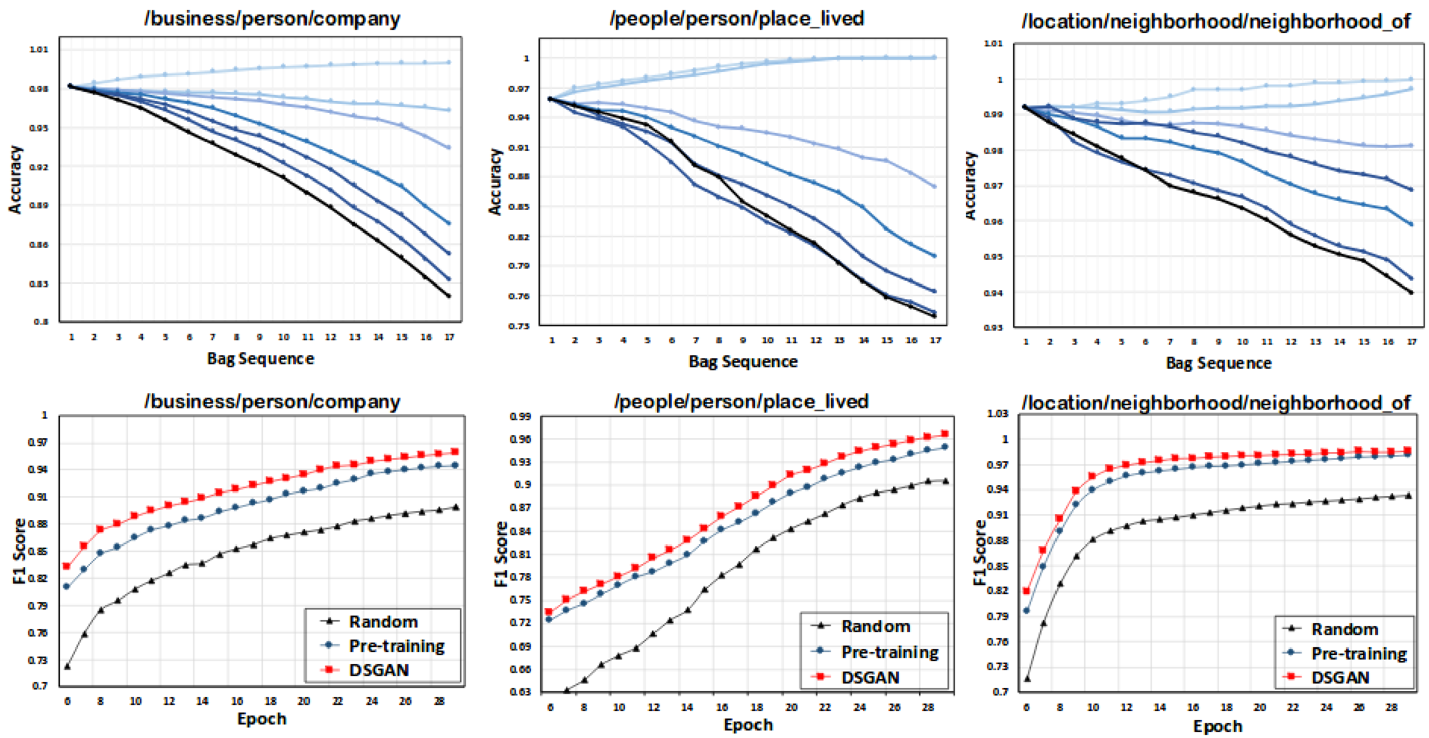

Figure 3: The convergence of the DSGAN training process for 3 relation types and the performance of their corresponding generators. The figures in the first row present the performance change on N^D in some specific epochs during processing the $B = \{B^1, B^2, ...B^N\}$. Each curve stands for one epoch; The color of curves become darker as long as the epoch goes on. Because the discriminator reloads the pre-trained parameters at the beginning of each epoch, all curves start from the same point for each relation type; Along with the adversarial training, the generator gradually collapses the discriminator. The figures in the second row reflect the performance of generators from the view of the difficulty level of training with the positive datasets that are generated by different strategies. Based on the noisy DS positive dataset P, *DSGAN* represents that the cleaned positive dataset is generated by our DSGAN generator; *Random* means that the positive set is randomly selected from P; *Pre-training* denotes that the dataset is selected according to the prediction probability of the pre-trained generator. These three new positive datasets are in the same size.

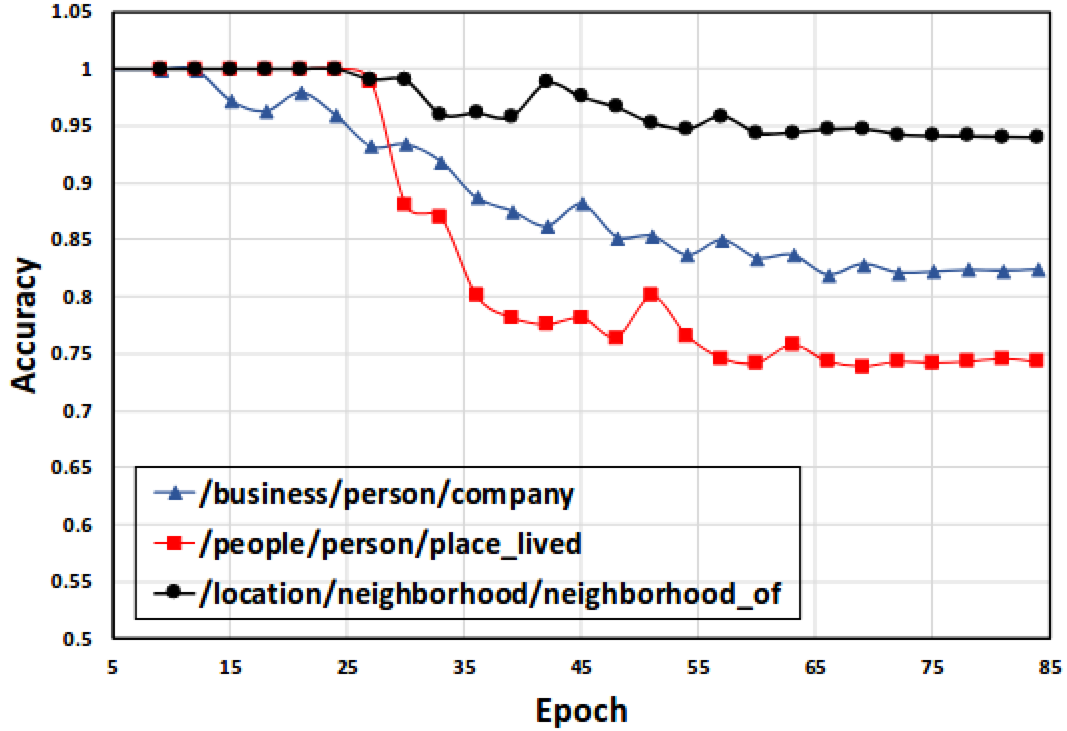

Figure 4: The performance change of the discriminator on N^D during the training process. Each point in the curves records the prediction accuracy on N^D when finishing each epoch. We stop the training process when this accuracy no longer decreases.

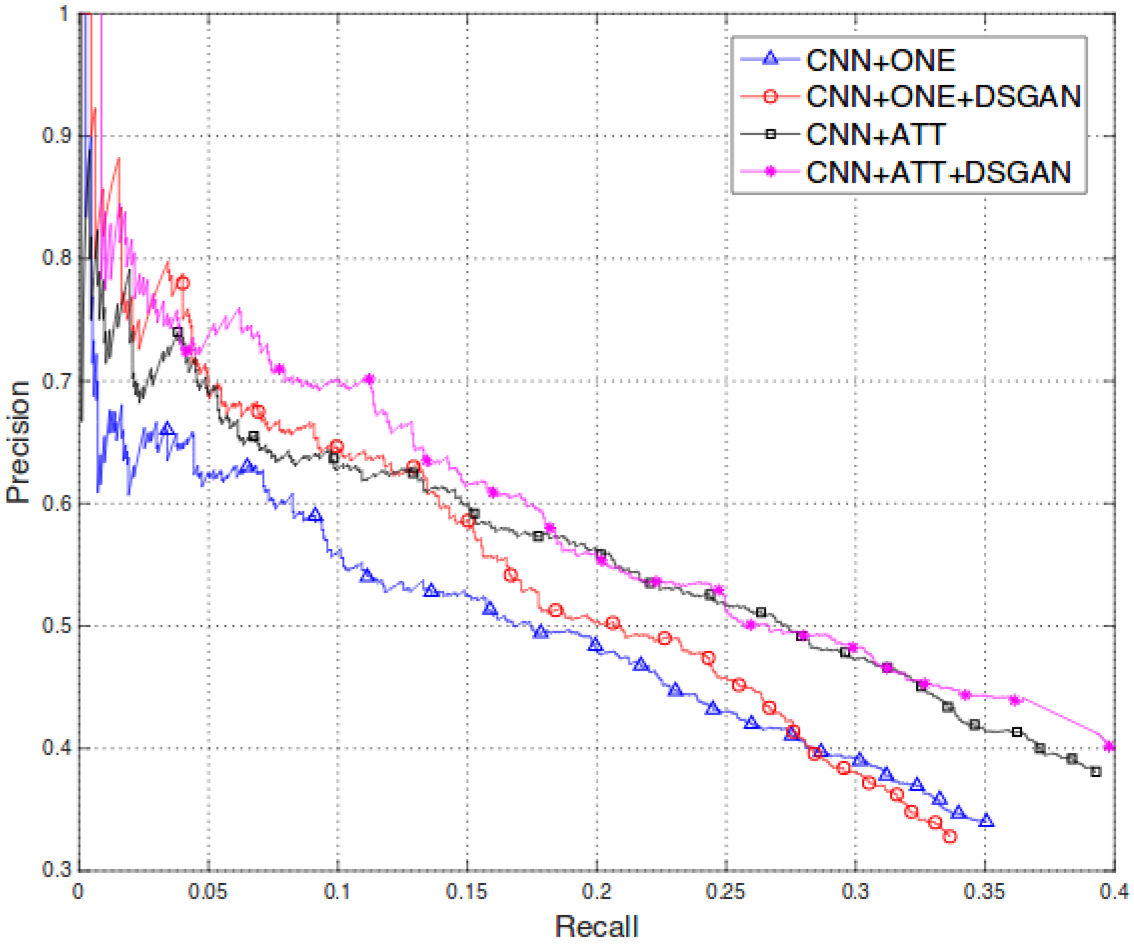

Figure 5: Aggregate PR curves of CNN˙based model.

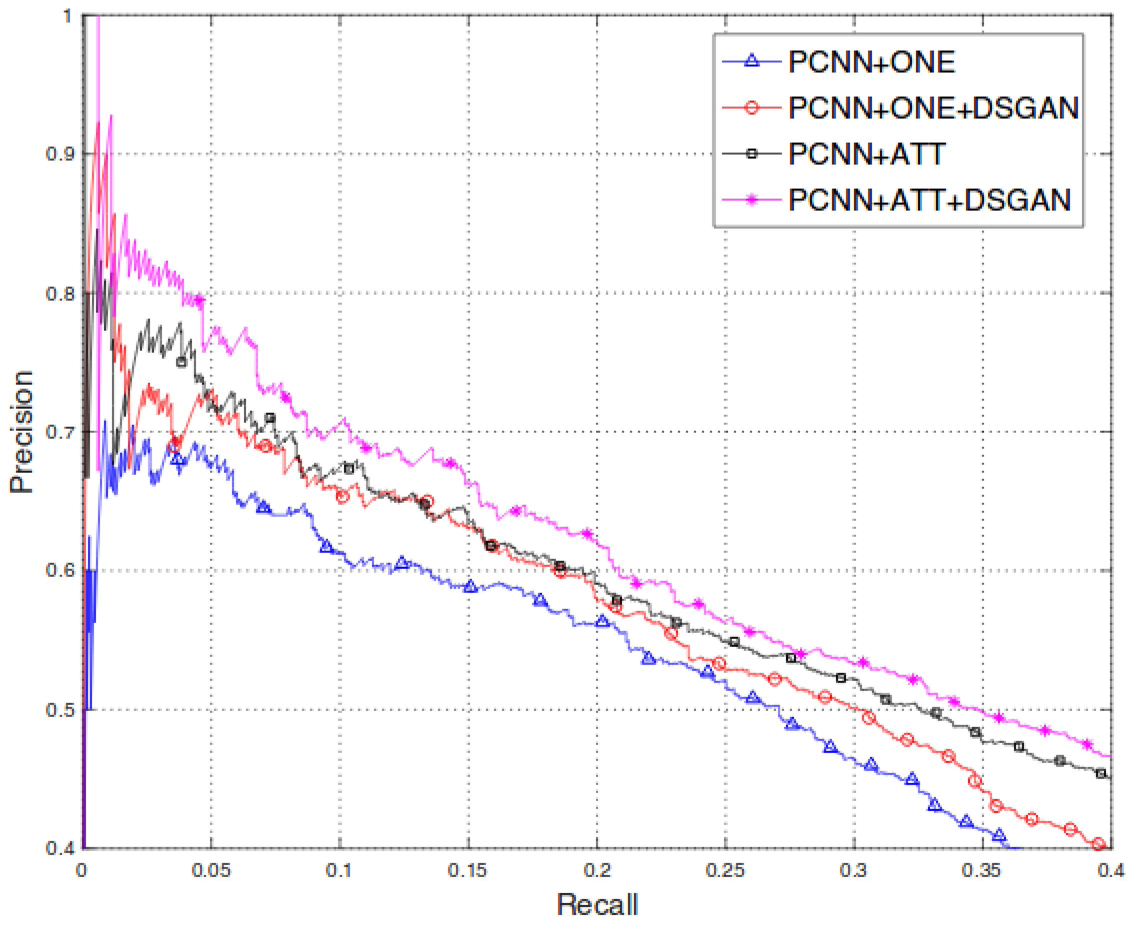

Figure 6: Aggregate PR curves of PCNN˙based model.

Model	-	+DSGAN	p-value
CNN+ONE	0.177	**0.189**	4.37e-04
CNN+ATT	0.219	**0.226**	8.36e-03
PCNN+ONE	0.206	**0.221**	2.89e-06
PCNN+ATT	0.253	**0.264**	2.34e-03

Table 2: Comparison of AUC values between previous studies and our DSGAN method. The *p-value* stands for the result of t-test evaluation.

Conclusions

- consider adversarial learning to denoise the distant supervision relation extraction dataset
- sentence-level and model-agnostic, so it can be used as a plug-and-play technique for any relation extractors
- can generate a cleaned dataset without any supervised information, boost the performance

Reinforcement Learning for Distant Supervision

- **states**:the information from the current sentence and the sentences that have been removed in early states
- **actions**:remove or retain the current instance
- **rewards**: $R_i = \alpha(F_1^i - F_1^{i-1})$

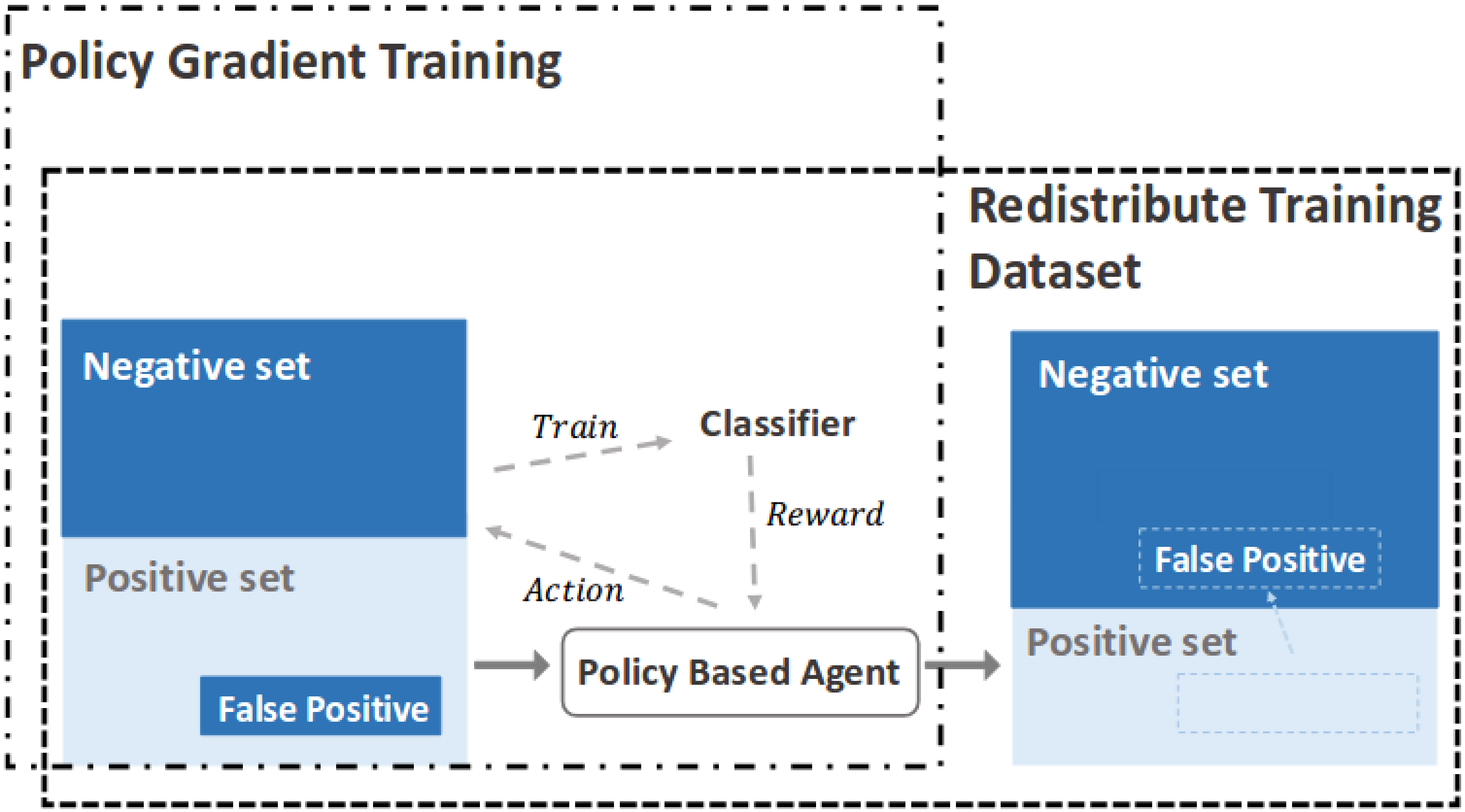

Figure 1: Our deep reinforcement learning framework aims at dynamically recognizing false positive samples, and moving them from the positive set to the negative set during distant supervision.

Pre-training Strategy

- distantly-supervised positive set
- randomly extract part of distantly supervised negative set (10x)
- stop training process when the accuracy reaches 85% ~ 90%

- $P_t^{ori}, P_v^{ori}, N_t^{ori}, N_v^{ori}$
- removes Φ_i from P_t^{ori} according to policy $\pi(a|s)$
- $P_t = P_t^{ori} - \Phi_i, N_t = N_t^{ori} + \Phi_i$
- validation set is same
- removing the fixed number of sentences in each epoch

Loss Function

- the different parts of the removed parts in different epochs are the determinant of the change of F_1 scores

$$\Omega_{i-1} = \Phi_{i-1} - (\Phi_i \bigcap \Phi_{i-1})$$

$$\Omega_i = \Phi_i - (\Phi_i \bigcap \Phi_{i-1})$$

$$J(\theta) = \sum^{\Omega_i} \log \pi(a|s;\theta)R + \sum^{\Omega_{i-1}} \log \pi(a|s;\theta)(-R)$$

Algorithm 1 Retraining agent with rewards for relation k. For a clearer expression, k is omitted in the following algorithm.

Require: Positive set $\{P_t^{ori}, P_v^{ori}\}$, Negative set $\{N_t^{ori}, N_v^{ori}\}$, the fixed number of removal γ_t, γ_v

Load parameters θ from pre-trained policy network
Initialize s^* as the all-zero vector with the same dimension of s_j
for epoch $i = 1 \to N$ **do**
    for $s_j \in P_t^{ori}$ **do**
        $\widetilde{s_j} = concatenation(s_j, s^*)$
        Randomly sample $a_j \sim \pi(a|\widetilde{s_j};\theta)$; compute $p_j = \pi(a = 0|\widetilde{s_j};\theta)$
        if $a_j == 0$ **then**
            Save tuple $t_j = (\widetilde{s_j}, p_j)$ in T and recompute the average vector of removed sentences s^*
        end if
    end for
    Rank T based on p_j from high to low, obtain T_{rank}
    for t_i in $T_{rank}[:\gamma_t]$ **do**
        Add $t_i[0]$ into Ψ_i
    end for
    $P_t^i = P_t^{ori} - \Psi_i$, $N_t^i = N_t^{ori} + \Psi_i$, and generate the new validation set $\{P_v^i, N_v^i\}$ with current agent
    Train the relation classifier based on $\{P_t^i, N_t^i\}$
    Calculate F_1^i on the new validation set $\{P_v^i, N_v^i\}$, and Save F_1^i, Ψ_i
    $\mathcal{R} = \alpha(F_1^i - F_1^{i-1})$
    $\Omega_{i-1} = \Psi_{i-1} - \Psi_i \cap \Psi_{i-1}$; $\Omega_i = \Psi_i - \Psi_i \cap \Psi_{i-1}$

    Updata θ: $g \propto \nabla_\theta \sum^{\Omega_i} \log \pi(a|s;\theta)\mathcal{R} + \nabla_\theta \sum^{\Omega_{i-1}} \log \pi(a|s;\theta)(-\mathcal{R})$
end for

ID	Relation	Original	Pretrain	RL
1	/peo/per/pob	55.60	53.63	**55.74**
2	/peo/per/n	78.85	80.80	**83.63**
3	/peo/per/pl	86.65	89.62	**90.76**
4	/loc/loc/c	80.78	83.79	**85.39**
5	/loc/cou/ad	**90.9**	88.1	89.86
6	/bus/per/c	81.03	82.56	**84.22**
7	/loc/cou/c	88.10	93.78	**95.19**
8	/loc/adm/c	86.51	85.56	**86.63**
9	/loc/nei/n	96.51	97.20	**98.23**
10	/peo/dec/p	82.2	83.0	**84.6**

Table 2: Comparison of F_1 scores among three cases: the relation classifier is trained with the original dataset, the redistributed dataset generated by the pre-trained agent, and the redistributed dataset generated by our RL agent respectively. The name of relation types are abbreviated: $/peo/per/pob$ represents $/people/person/place\_of\_birth$

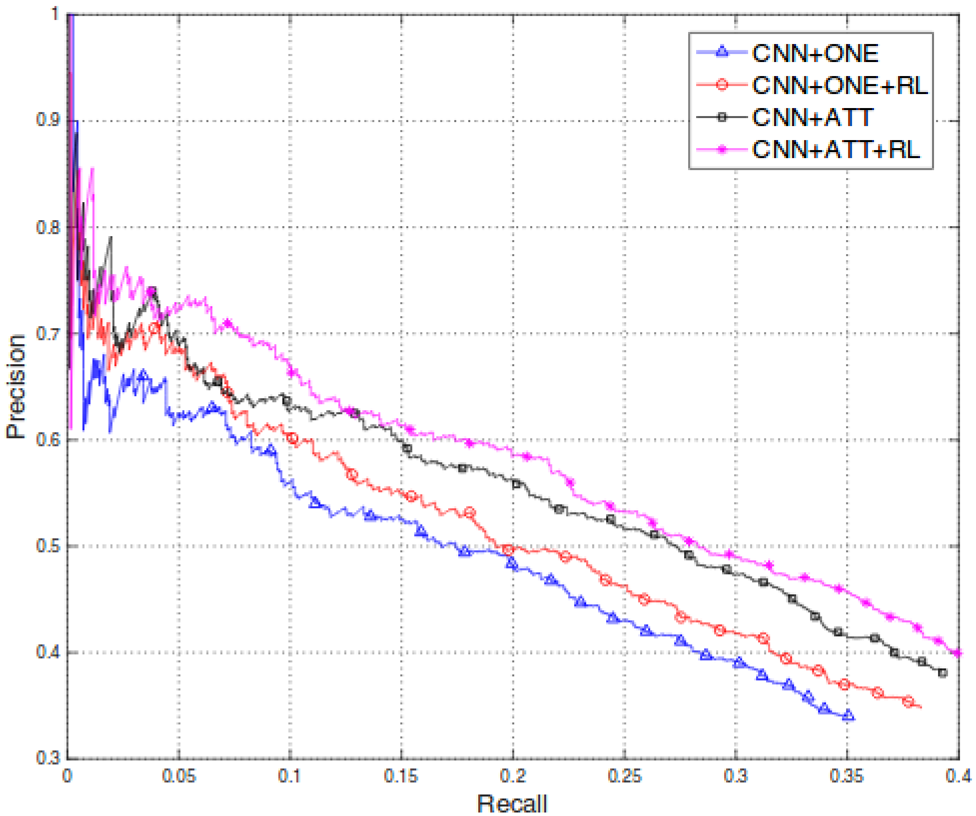

Figure 3: Aggregate PR curves of CNN˙based model.

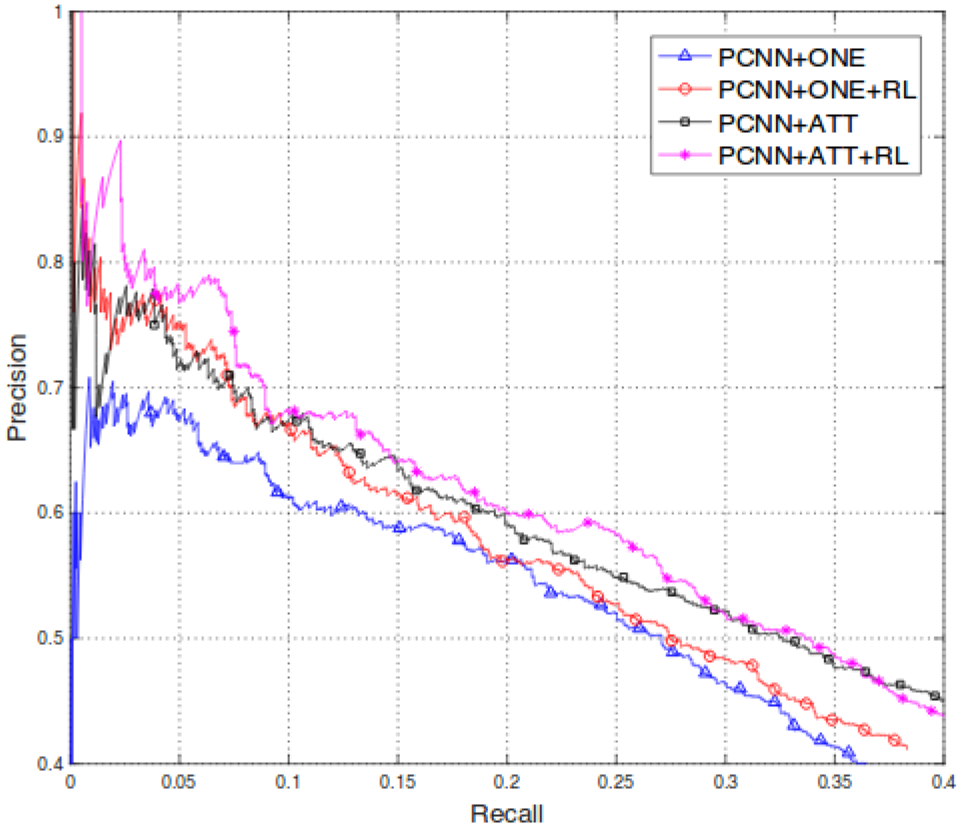

Figure 4: Aggregate PR curves of PCNN˙based model.

Model	**-**	**+RL**	**p-value**
CNN+ONE	0.177	**0.190**	1.24e-4
CNN+ATT	0.219	**0.229**	7.63e-4
PCNN+ONE	0.206	**0.220**	8.35e-6
PCNN+ATT	0.253	**0.261**	4.36e-3

Table 3: Comparison of AUC values between previous studies and our RL method, and the p-value of t-test.

Relation	/people/person/place_of_birth
FP	1. GHETTO SUPERSTAR (THE MAN THAT I AM) – Ranging from **Pittsburgh** to Broadway, **Billy Porter** performs his musical memoir.
FP	1. “They are trying to create a united front at home in the face of the pressures **Syria** is facing,“ said **Sami Moubayed**, a political analyst and writer here. 2. “Iran injected **Syria** with a lot of confidence: stand up, show defiance,“ said **Sami Moubayed**, a political analyst and writer in Damascus.
Relation	**/people/deceased_person/place_of_death**
FP	1. Some **New York city** mayors – **William O’Dwyer**, Vincent R. Impellitteri and Abraham Beame – were born abroad. 2. Plenty of local officials have, too, including two **New York city** mayors, James J. Walker, in 1932, and **William O’Dwyer**, in 1950.

Table 4: Some examples of the false positive samples detected by our policy-based agent. Each row denotes the annotated sentences of one entity pair.

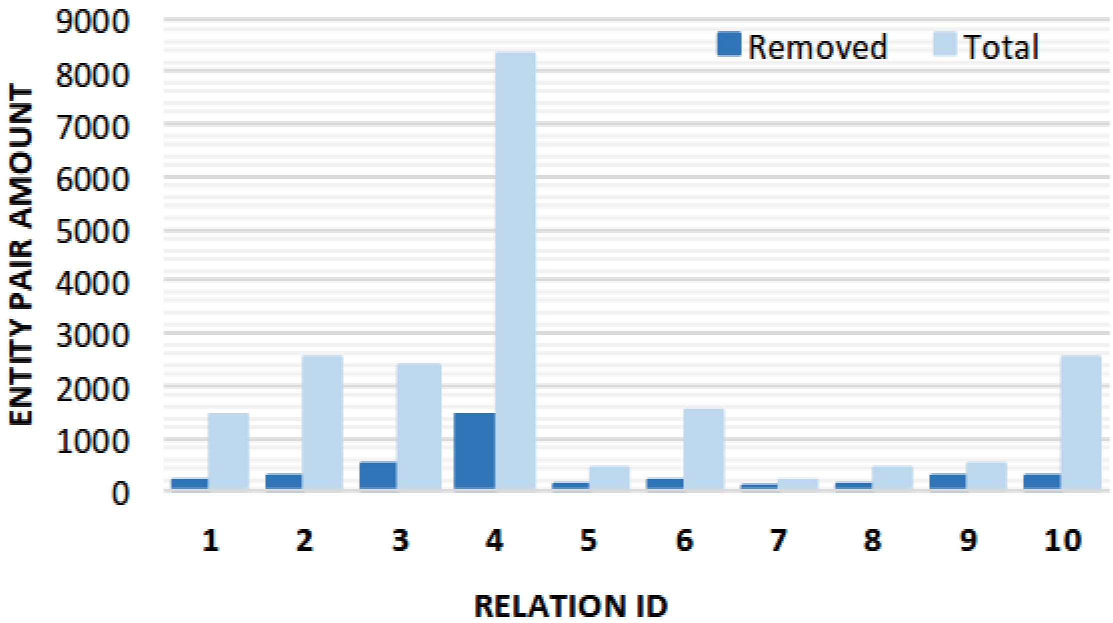

Figure 5: This figure presents the scale of the removed part for each relation type, where the horizontal axis corresponds to the IDs in Table 2.

Conclusions

- propose a novel deep reinforcement learning framework for robust distant supervision relation extraction
- model-independent, meaning that it could be applied to any state-of-the-art relation extractors
- can boost the performances of recently proposed neural relation extractors

Thanks Q&A