Best from Top *k* Versus Top 1: Improving Distant Supervision Relation Extraction with Deep Reinforcement Learning

Yaocheng Gui[1], Qian Liu[2], Tingming Lu[1], and Zhiqiang Gao[1(✉)]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, China
{yaochgui,zqgao}@seu.edu.cn, lutingming@163.com
[2] School of Computer Science and Technology,
Nanjing University of Posts and Telecommunications, Nanjing, China
qianliu@njupt.edu.cn

Abstract. Distant supervision relation extraction is a promising approach to find new relation instances from large text corpora. Most previous works employ the *top 1* strategy, i.e., predicting the relation of a sentence with the highest confidence score, which is not always the optimal solution. To improve distant supervision relation extraction, this work applies the *best from top k* strategy to explore the possibility of relations with lower confidence scores. We approach the *best from top k* strategy using a deep reinforcement learning framework, where the model learns to select the optimal relation among the top k candidates for better predictions. Specifically, we employ a deep Q-network, trained to optimize a reward function that reflects the extraction performance under distant supervision. The experiments on three public datasets - of news articles, Wikipedia and biomedical papers - demonstrate that the proposed strategy improves the performance of traditional state-of-the-art relation extractors significantly. We achieve an improvement of 5.13% in average F_1-score over four competitive baselines.

Keywords: Distant supervision · Relation extraction · Deep reinforcement learning · Deep Q-networks

1 Introduction

Relation extraction aims to predict the relation for entities in a sentence [20]. It is an important task in information extraction and natural language understanding. However, for the early development of relation extraction applications, a major issue is creating human labeled training sets which is both time-consuming and expensive.

Therefore, a new task in terms of distant supervision relation extraction [2, 4,8,13,15,18] becomes popular, since it uses entity pairs and their relations from knowledge bases to heuristically create training sets. The definition of distant supervision relation extraction is as follows:

Q. Yang et al. (Eds.): PAKDD 2019, LNAI 11441, pp. 199–211, 2019.
https://doi.org/10.1007/978-3-030-16142-2_16

Definition 1. *Let $\mathcal{X}$ be the sentence space and $\mathcal{Y}$ the set of relations,* ***distant supervision relation extraction*** *aims to learn a function $f : 2^{\mathcal{X}} \rightarrow 2^{\mathcal{Y}}$ from a given data set $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)\}$, where $X_i \subseteq \mathcal{X}$ is a set of sentences $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|X_i|}\}$, $Y_i \subseteq \mathcal{Y}$ is a set of relations $\{y_1, y_2, \ldots, y_{|Y_i|}\}$.*

Here X_i denotes the set of sentences that relates to the ith entity pair and Y_i its relations, $|X_i|$ denotes the number of sentences in X_i and $|Y_i|$ the number of relations in Y_i.

Strategy *Top 1.* Most previous works resolve distant supervision relation extraction by a sentence-level extractor along with an entity-pair-level predicator to make the final decision [2,4,15,18]. The sentence-level extractor outputs a set of real-valued scores for each sentence $\mathbf{x}$, the score $h(\mathbf{x}, y)$ indicates the confidence of sentence $\mathbf{x}$ describes relation y. For each sentence, at least one relation should be selected and fed to the entity-pair-level predictor, which will make the final prediction based on all the selected relations for all the sentences. Existing distant supervision relation extraction models usually employ the *top 1* strategy, i.e., selecting $\arg\max_{y \in \mathcal{Y}} h(\mathbf{x}, y)$ as the predicted relation for $\mathbf{x}$. However, the relation with the highest confidence score, i.e., $\arg\max_{y \in \mathcal{Y}} h(\mathbf{x}, y)$ is not always the optimal option, existing models have not explored the possibility of other relations with lower confidence scores.

For example, Fig. 1 shows a sample sentence that describes the relation instance (Ernst Haefliger, *place_of_birth*, Davos). As shown in the bottom of the figure, a sentence-level extractor outputs the confidence score for each relation. Obviously, the relation with the highest confidence score (i.e., *place_lived*) is not the best choice for the sentence.

Strategy *Best from Top* k. This paper proposes a strategy to address the issue in existing models. Instead of employing the *top 1* strategy, we investigate the possibility of improving distant supervision relation extraction by using the *best from top k* strategy, i.e., we choose the best prediction from the top k candidates $\{y | \forall y \in \mathcal{Y}, rank(h(\mathbf{x}, y)) \leq k\}$, where $rank(h(\mathbf{x}, y))$ returns the rank of y derived from $h(\mathbf{x}, y)$, and then feed it to the entity-pair-level predictor. For example, there is a chance to make an optimal selection for the sentence in Fig. 1 (i.e., *place_of_birth*) using the *best from top k* strategy ($k = 3$).

Relation instance:
(Ernst Haefliger, *place_of_birth*, Davos)

Sample sentence:
Ernst Haefliger (pronounced heff-ligger) was born in Davos on July 6, 1919 , and studied at the Wettinger seminary and the Zurich conservatory before moving to Vienna , where he became a student of the Tenor Julius Patzak .

Top k candidates:
⟨*place_lived* 0.541, *place_of_birth* 0.311, *nationality* 0.072⟩

Fig. 1. The top k ($k = 3$) outputs of the sentence-level extractor.

Specifically, we address the *best from top k* strategy using a deep reinforcement learning (RL) framework that learns to predict a set of most possible relations for each entity pair based on the top k candidate relations from the sentence-level extractor. To effectively select among the top k candidate relations, the state representation encodes information about the confidence scores and the context in which the entity pair appears. We train the RL model using a deep Q-network (DQN) [9], whose goal is to learn to select good actions in order to optimize the reward function, which reflects the extraction performance under distant supervision.

While we use the sentence-level extractors of four state-of-the-art models in the experiments, i.e., MultiR [2], MIMLRE [15], CNN+ATT and PCNN+ATT [4], this method can be inherently applied to other models. The experiments on three public datasets from different domains, the New York Times news articles, the Wikipedia articles, and the PubMed paper abstracts, demonstrate that the proposed method outperforms four comparative baselines significantly. The average F_1-score has an improvement of 5.13% compared with baseline models.

The contributions in this work include:

- This work proposes the *best from top k* strategy, which is implemented with a novel deep reinforcement learning framework, to improve existing distant supervision relation extraction models.
- The proposed strategy can be applied to any distant supervision relation extractors that output confidence scores for predicted relations.

2 Related Work

Pioneer work in distant supervision relation extraction used a set of frequent relations in Freebase to train relation extractors over Wikipedia without labeled data [8]. Since then, a lot of works focused on relation extraction using distant supervision. However, using distant supervision to annotate training data would introduce a lot of false positive labels [13].

To alleviate the wrong label issue, a series of graphical models have been proposed based on hand-craft features. A joint model was proposed to learn with multiple relations [2]. Later, a multi-instance multi-label learning (MIML) framework was proposed to further improve the performance [15]. Additional information has been employed to reduce wrong labels of training data upon these models. For example, the fine-grained entity types [3], the document structure [6], the side information about rare entities [14], and the human labeled data [7].

Neural network models have shown superior performance over approaches using hand-crafted features in distant supervision relation extraction [4,18]. Convolutional neural networks (CNN) and piecewise convolutional neural networks (PCNN) are among the first deep neural network models that have been applied to this task [18]. An instance-level selective attention mechanism was introduced for multi-instance multi-label learning [4], and has significantly improved the prediction accuracy for several of these base deep neural network models.

Recently, deep reinforcement learning have been applied to distant supervision relation extraction [1,12,19]. The relation extractor is regarded as a reinforcement learning agent and the goal is to achieve higher long-term reward [19]. To further improve the performance, an instance selector was proposed to cast the sentence selection task as a reinforcement learning problem to choose high-quality training sentence for a relation classifier [1], and a false-positive indicator was proposed to automatically recognize false positive labels and then redistribute them into negative examples [12].

This work relates to the previous works that based on graphical models and neural network models because their sentence-level extractors can be reused in our model. This work also relates to the previous works that based on deep RL methods as we also learn a RL agent. The main differences between our work and existing deep RL methods are that our RL agent tries to improve the testing process of relation extraction while theirs are designed for better training process, and our RL agent is based on deep Q-networks while theirs are mainly based on policy gradient. Considering the training cost of the model, we do not update the parameters of sentence-level extractors during training the RL agent. Thus the learning process tends to be faster than the previous works.

3 Framework

The task of improving relation extraction models under distant supervision can be modeled as a markov decision process (MDP), which learns to utilize the outputs of a sentence-level extractor to improve extractions. We represent the MDP as a tuple $\langle S, A, T, R \rangle$, where $S = \{s\}$ is the space of all possible states, $A = \{a\}$ is the set of all actions, $R(s, a)$ is the reward function, and $T(s'|s, a)$ is the transition function. The overall framework of the task is shown in Fig. 2. Given a set of sentences, the sentence-level extractor produces the predicated relations and their confidence scores. The RL agent selects one action for each state to produce the best relation, which is merged into the selected relation set.

States. The state s in our MDP consists of the sentence-level extractor's confidence scores of the predicted relations and the context in which the entity pair appears. As shown in Fig. 2 (the bottom boxes), we represent the state as a continuous real-valued vector incorporating the following pieces of information:

- Confidence scores of current selected relations between the entities.
- Confidence scores of the newly predicted relations between the entities in the new sentence.
- One-hot encoding of matches between current and newly predicted relations.
- TF-IDF counts[1] of context words, which occur in the neighborhood of the entities in a sentence.

[1] TF-IDF counts are computed based on the training sentences.

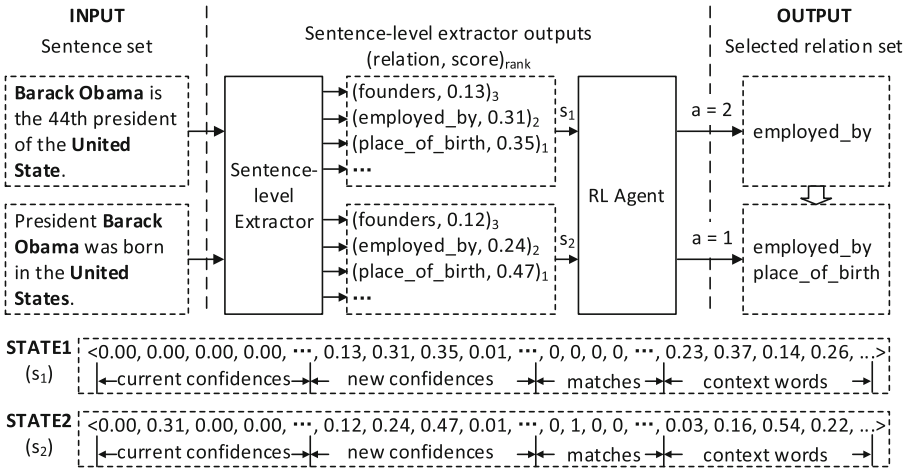

Fig. 2. Overall framework. The left boxes are input sentences. The middle boxes are predicted relations and their confidence scores of the sentence-level extractor. The right boxes are relations that selected by the RL agent step by step from the current episode. The bottom boxes are two sample states corresponding to the input sentences.

Actions. We define an action $a \in \{0, 1, 2, \ldots, k\}$[2] to indicates whether the predicted relations by the sentence-level extractor should be rejected or accepted. Here the number k corresponds to the k in the *best from top k* strategy. The decision can be one of the following types: (1) reject all the relations, i.e., $a = 0$, or (2) accept the lth ($1 \leq l \leq k$) relation according to the ranked predicted confidence scores, i.e., $a = l$. The agent continues to inspect more sentences until the episode ends. The current relation and confidence scores are simply updated with the accepted relation and the corresponding confidences.

Rewards. The reward function is an indicator of the quality of chosen relations. For a certain set of training sentences $X_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|X_i|}\}$ of an episode, the agent selects an action for each sentence to determine whether the sentence-level extractor's outputs should be accepted or not. We assume that the agent has a terminal reward when it finishes all the selection. Therefore we receive a delayed reward at the terminal state $s_{|X_i|+1}$ based on the performance of current selected relation set Y^{cur} for the ith entity pair on X_i (see the outputs in Fig. 2).

At other states, after an action is taken (i.e., a relation is chosen), the reward is computed immediately based on the agent's performance on the newly predicted relation set Y_j^{new} ($j \leq |X_i|$) for the new sentence. The performances of Y^{cur} and Y_j^{new} are computed using the number of true positive (i.e., TP) and false positive (i.e., FP) relations compared with the distantly annotated data. The intuition is that, the reward is positive if true positive relations are more than false positive ones, and the reward is negative vice versa. Note that, the

[2] We choose k by ranging it from 1 to 5 in our experiments, the model achieves the best performance in most cases when $k = 3$.

reward is zero if the agent decided to reject all the predicated relations for the new sentence. Therefore, the reward function is defined as follows:

$$r(s_j|X_i) = \begin{cases} TP(Y_j^{new}) - FP(Y_j^{new}) & j < |X_i| + 1 \\ TP(Y^{cur}) - FP(Y^{cur}) & j = |X_i| + 1 \end{cases} \tag{1}$$

Transitions. Each episode starts off with an initial state that consists of an empty set of current relation and its confidence score respect to the entity pair (see the initial state s_1 in Fig. 2). The subsequent steps in the episode involve traversing the set of sentences and integrating the extracted new relation to the current relation set. The transition function $T(s'|s, a)$ incorporates the selected decision a from the agent in state s along with the relation from the next sentence and produces the next state s', e.g., $s = s_1, s' = s_2$ in Fig. 2.

Algorithm 1 details the MDP framework for the training phase of the *best from top k* strategy. During the testing phase, each sentence is handled only once in a single episode. The training process of our agent contains M epochs. For the ith entity pair, we first initialize an empty set to the current relation set, denoted as Y^{cur} and set the initial reward r to 0 (line 3), then traverse all training sentences in X_i to update the current relation set Y^{cur} and the immediate reward r according to the action taken by the agent based on the state s_j (lines 4–13). The terminal state $s_{|X_i|+1}$ and the delayed reward r for the ith entity pair based on X_i is then sent to the agent (line 14). After the training, the agent learns a policy to further improve the relation extraction results of sentence-level extractors.

Algorithm 1. MDP framework for the *best from top k* strategy

```
1: for epoch = 1, M do
2:   for i = 1, N do
3:     Y^cur ← {}, r ← 0
4:     for j = 1, |X_i| do
5:       Compute confidence score vector F(x_j)
6:       Compute context vector C(x_j)
7:       Form state s_j using Y^cur, F(x_j) and C(x_j)
8:       Send (s_j, r) to agent
9:       Get action a from agent
10:      Y_j^new ← Select(F(x_j), a)
11:      Y^cur ← Reconcile(Y^cur, Y_j^new)
12:      update r using Equation 1
13:    end for
14:    Send (s_{|X_i|+1}, r) to agent
15:  end for
16: end for
```

4 DQN Parameter Learning

For the purpose of learning a good policy for an agent, we utilize the deep reinforcement learning framework described in the previous section. Following previous work [10], the MDP can be viewed in terms of a sequence of transitions (s, a, r, s'). The agent seeks to learn a policy to determine which action a to perform in state s. A commonly used technique for learning an optimal policy is Q-learning [17], in which the agent iteratively updates $Q(s, a)$ using the rewards obtained from experiences. The updates are derived from the recursive Bellman equation [16] for the optimal Q:

$$Q^*(s, a) = E\left[r + \gamma \max_{a'} Q^*(s', a') \middle| s, a\right] \tag{2}$$

where r is the reward and γ is a factor discounting the value of future rewards and the expectation is taken over all transitions involving state s and action a.

We use DQN [9] as a function approximator $Q(s, a) \approx Q(s, a; \theta)$, since our problem involves a continuous state space. The DQN has been shown to learn better value functions than linear approximators [9] and can capture non-linear interactions between different pieces of information in continuous state [10]. We use a DQN that consists of two linear layers (20 hidden units each) followed by rectified linear units (ReLU), along with a separate output layer.

The parameters θ of the DQN are learnt using stochastic gradient descent with RMSprop[3]. The parameter update aims to close the gap between the $Q(s, a; \theta)$ predicted by the DQN and the expected Q-value from the experiences. Following previous work [9], we make use of a (separate) target Q-network to calculate the expected Q-value, in order to have stable updates. The target Q-network parameters $\hat{\theta}$ is periodically updated with the current parameters θ. We also make use of an experience replay memory $\mathcal{D}$ to store transitions. To perform updates, we sample a batch of transitions (s, a, r, s') randomly from $\mathcal{D}$ and minimize the loss function:

$$\mathcal{L}(\theta) = E_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q(s', a'; \hat{\theta}) - Q(s, a; \theta)\right)^2\right] \tag{3}$$

The learning updates are made every training step using the following gradients:

$$\nabla_\theta \mathcal{L}(\theta) = E_{s,a,r,s'}\left[2\left(r + \gamma \max_{a'} Q(s', a'; \hat{\theta}) - Q(s, a; \theta)\right) \nabla_\theta Q(s, a; \theta)\right] \tag{4}$$

5 Experimental Setup and Results

In our experiments, we first evaluate the performance of the proposed model compared with four state-of-the-art baseline models. Then to further illustrate the effectiveness of the *best from top k* strategy, we also evaluate the performances of the models that apply different strategies respectively.

[3] See http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

5.1 Dataset

Our experiments use three public datasets from different domains. (1) **NYT** [13] is constructed from New York Times news articles. It contains 522,611 sentences in the training set, and 172,448 sentences in the testing set. Among these data, there are 53 unique relations from Freebase including a special relation *NA* that signifies no relation between two entities in a sentence. (2) **Wiki-KBP** [5] is derived from Wikipedia articles. It contains 23,111 sentences in the training set, and 15,847 sentences in the testing set. There are 7 unique relations from the KBP 2013 slot filling database including a *NA* relation. (3) **BioInfer** [11] is sampled from PubMed paper abstracts. It contains 1,139 sentences in the training set, and 876 sentences in the testing set. There are 92 unique relations including a *NA* relation among these data.

5.2 Baseline Extractors

We compare the proposed model with the following distant supervision relation extraction models in our experiments. Note that the sentence-level extractors of these baseline models are also used in our model.

MultiR [2] is a typical work based on probabilistic graphical model for multi-instance learning. It uses the perceptron algorithm for learning and a greedy search algorithm for inference. We implemented this model using the publicly available code[4].

MIMLRE [15] is a graphical model for multiple instances and multiple relations. It is trained by using hard discriminative Expectation-Maximization. We use the publicly available code provided by the authors[5].

CNN+ATT and **PCNN+ATT** [4] are two state-of-the-art neural networks for relation extraction, which adopt a sentence-level attention over the sentences and thus can reduce the weights of noisy sentences. We implemented the two models using the publicly available code[6].

5.3 RL Models

We train a RL model using the proposed *best from top k* strategy based on each sentence-level extractor respectively. For example, **MultiR+RL** uses the same sentence-level extractor as in **MultiR**, then learn a RL model to generate the final predictions in the entity-pair-level.

We used the same network architecture, hyperparameter values and learning procedure throughout to demonstrate that our approach robustly learns successful policies over a variety of datasets based only on distant supervision knowledge. The RL models are trained for 10,000 steps every epoch using the sentence-level extractors, and evaluate the entire test set every epoch. The final

[4] http://www.cs.washington.edu/ai/raphaelh/mr/.
[5] http://nlp.stanford.edu/software/mimlre.shtml.
[6] https://github.com/thunlp/NRE/.

evaluation metrics reported are averaged over 20 epochs after 100 epochs of training. We used a replay memory $\mathcal{D}$ of size $500k$, and a discount (γ) of 0.8. We set the learning rate to $2.5E^{-5}$. The ϵ-greedy exploration is annealed from 1 to 0.1 over $500k$ transitions. The target-Q network is updated every $5k$ steps.

5.4 Evaluation Metrics

Similar to the previous works [13], we adopt the held-out evaluation to evaluate our models, which can provide an approximate measure of the classification ability without costly human evaluation. The held-out evaluation compares the predicted relations of the entity pair with the gold relations, which is automatically labeled by knowledge bases. It's an effective evaluation method for large dataset. Precision (P), recall (R), and F_1-score (F_1) are used as our evaluation metrics.

We compute the evaluation metrics based on the distinct occurrence of each relation instance, i.e., any occurrence of the extracted relation instance is considered as one extraction. All compared models are evaluated use the same method.

5.5 Experimental Results

The precisions, recalls and F_1-scores of eight compared models evaluated on three datasets are shown in Table 1. We observe from the table that all RL models yield obvious and steady improvements compared with baseline models on all datasets except the PCNN+ATT+RL model on the BioInfer dataset. It not only demonstrates the rationality of our *best from top k* strategy, but also verifies our hypothesis that the state-of-the-art distant supervision relation extractors can be further improved by the *best from top k* strategy.

Specifically, the F_1 score of CNN+ATT+RL has 18.2% improvement compared with CNN+ATT on the NYT dataset, and the average F_1 score of all RL models on all datasets has 5.13% improvement compared with that of all baseline models. These comparable results illustrate that our approach is capable in improving relation extraction based on distant supervision in different domains, and making RL models develop towards a good direction.

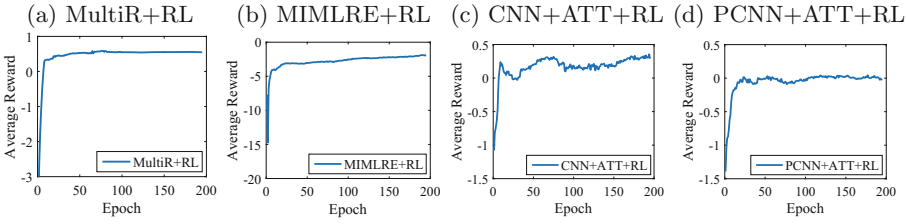

Fig. 3. Training curves tracking the RL model's average reward achieved per episode for models (a) MultiR+RL, (b) MIMLRL+RL, (c) CNN+ATT+RL and (d) PCNN+ATT+RL on the dataset NYT.

Table 1. Precision, Recall and F_1-score of the compared models on three datasets. The RL models use the same sentence-level extractor as in the baseline models, and apply the proposed *best from top k* strategy. The average F_1-score improvement of RL models over the baseline models is 5.13%.

System	NYT			Wiki-KBP			BioInfer		
	P	R	F_1	P	R	F_1	P	R	F_1
MultiR	0.756	0.371	0.497	0.444	0.427	0.435	0.102	0.087	0.094
MultiR+RL	0.731	0.412	**0.527**	0.421	0.620	**0.501**	0.117	0.114	**0.116**
MIMLRE	0.529	0.506	0.517	0.489	0.461	0.475	0.059	0.049	0.054
MIMLRE+RL	0.722	0.427	**0.537**	0.515	0.677	**0.585**	0.073	0.148	**0.097**
CNN+ATT	0.965	0.426	0.591	0.654	0.680	0.667	0.119	0.100	0.109
CNN+ATT+RL	0.773	0.773	**0.773**	0.652	0.700	**0.675**	0.113	0.119	**0.116**
PCNN+ATT	0.938	0.504	0.656	0.604	0.593	0.599	0.193	0.179	**0.186**
PCNN+ATT+RL	0.764	0.779	**0.772**	0.592	0.637	**0.614**	0.190	0.176	0.183

Figure 3 shows the training curves tracking the average reward achieved per episode for each RL model on the dataset NYT. We can see from the figures that our RL models are able to improve the performance of all the traditional distant supervision relation extraction models in a stable manner. The same conclusion can be derived from the results on other datasets.

5.6 Analysis and Case Study

We analyze the influence of different k values for RL models with *best from top k* strategy. Figure 4 shows precision, recall and F_1-score of the compared approaches on the NYT dataset. Sub-figure (a) shows the comparison results of MultiR, MultiR+RL ($k = 1$) and MultiR+RL ($k = 3$). The MultiR method uses the *top 1* strategy. The difference between MultiR and MultiR+RL ($k = 1$) is that they use different methods in entity-pair-level to make the final prediction. We can see from sub-figure (a) that MultiR+RL ($k = 3$) achieves the best F_1-score. The same observations can be derived from other sub-figures. It illustrates that models applying the *best from top k* strategy ($k = 1$) is as good as those applying the *top 1* strategy, and models applying the *best from top k* strategy ($k > 1$) can achieve significant improvements compared with the baselines.

Table 2 shows three examples of sentence-level relation extraction for PCNN+ATT and PCNN+ATT+RL. For the first sentence, both models select the correct relation for the entities, which are labeled with subscripts. For the second sentence, PCNN+ATT selects the most possible but wrong relation, i.e., *NA* based on the predicted confidence scores in the brackets, while PCNN+ATT+RL selects the correct relation, i.e., *nationality*. This case explains why our RL models can achieve higher recalls than baseline models in most cases. For the third sentence, PCNN+ATT selects a wrong relation, i.e., *NA*, while PCNN+ATT+RL rejects all the predicted relations by PCNN+ATT since the

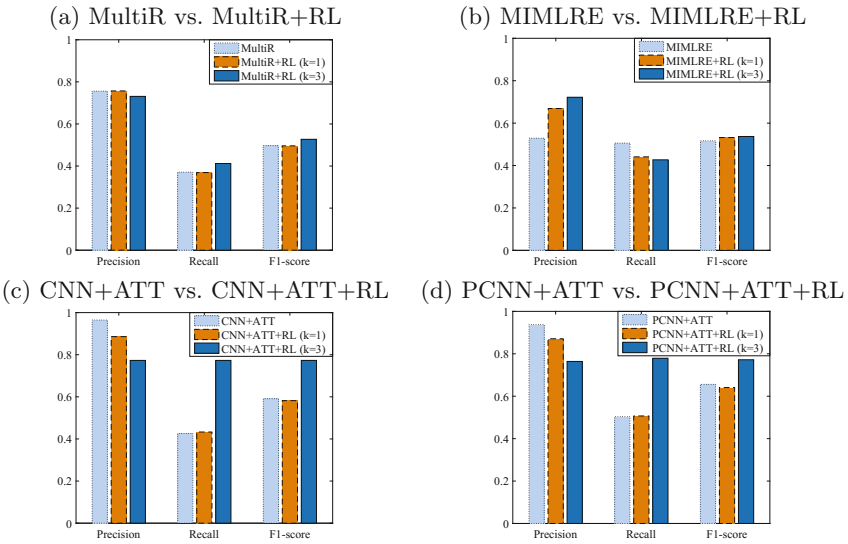

Fig. 4. Precision, Recall and F_1-score of the compared models that use the *top 1* strategy and the *best from top k* strategy ($k = 1$ and $k = 3$) on the NYT dataset.

Table 2. Relation extraction examples by different models. The correct relations between entities in the three sentences are *company*, *nationality* and *contains*.

Test sentence	PCNN+ATT	PCNN+ATT+RL
mel karmazin$_1$, the chief executive of **sirius satellite radio**$_2$, made a lot of ... radio on monday	**company(0.790)** NA(0.170) place_of_birth(0.011)	Company
a young cape verdean singer who was born in **portugal**$_2$, **lura**$_1$ specializes in bubbly, ... by cesaria evora	**NA(0.387)** nationality(0.159) place_lived(0.075)	Nationality
despite **madrid**$_2$'s efforts to catch up, barcelona arguably remains the design capital of **spain**$_1$, and vinçon ...	**NA(0.842)** nationality(0.042) place_of_birth(0.024)	/

correct relation, i.e., *contains* is not in the top 3 candidates. This case indicates that our RL models are able to prevent potential errors. It is clearly show that our model can do better relation extraction than traditional state-of-the-art distant supervision relation extraction models.

6 Conclusions

This paper proposed the *best from top k* strategy to improve existing distant supervision relation extraction models, which use the *top 1* strategy. The pro-

posed strategy chooses the best prediction from the top k candidates generated by the sentence-level extractor of the existing models. We approach the *best from top k* strategy using a deep RL framework, which employs a DQN to learn to select good actions for optimizing the reward function. Based on the deep RL framework, our model is capable to predict a set of possible relations for each entity pair in the entity-pair-level. In the experiments, we evaluate the performance of the proposed model compared with four state-of-the-art baselines, i.e., the MultiR, MIMLRE, CNN+ATT and PCNN+ATT models. The experimental results on three public datasets from different domains demonstrate that the proposed model that applies the *best from top k* strategy outperforms the comparative baselines that apply the *top 1* strategy significantly. The average F_1-score has 5.13% improvement compared with all baseline models.

Acknowledgements. This work is partially funded by the National Science Foundation of China under Grant 61170165, Grant 61702279, Grant 61602260, and Grant 61502095.

References

1. Feng, J., Huang, M., Zhao, L., Yang, Y., Zhu, X.: Reinforcement learning for relation classification from noisy data. In: Proceedings of AAAI 2018 (2018)
2. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of ACL 2011, pp. 541–550 (2011)
3. Koch, M., Gilmer, J., Soderland, S., Weld, D.S.: Type-aware distantly supervised relation extraction with linked arguments. In: Proceedings of EMNLP 2014, pp. 1891–1901 (2014)
4. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of ACL 2016, pp. 2124–2133 (2016)
5. Ling, X., Weld, D.S.: Fine-grained entity recognition. In: Proceedings AAAI 2012, vol. 12, pp. 94–100 (2012)
6. Lockard, C., Dong, X.L., Einolghozati, A., Shiralkar, P.: CERES: distantly supervised relation extraction from the semi-structured web. In: Proceedings of VLDB 2018, pp. 1084–1096 (2018)
7. Lourentzou, I., Alba, A., Coden, A., Gentile, A.L., Gruhl, D., Welch, S.: Mining relations from unstructured content. In: Phung, D., Tseng, V.S., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (eds.) PAKDD 2018. LNCS, vol. 10938, pp. 363–375. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93037-4_29
8. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of ACL 2009, pp. 1003–1011 (2009)
9. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
10. Narasimhan, K., Yala, A., Barzilay, R.: Improving information extraction by acquiring external evidence with reinforcement learning. In: Proceedings of EMNLP 2016, pp. 2355–2365 (2016)
11. Pyysalo, S., et al.: BioInfer: a corpus for information extraction in the biomedical domain. BMC Bioinform. **8**(1), 50 (2007)

12. Qin, P., Xu, W., Wang, W.Y.: Robust distant supervision relation extraction via deep reinforcement learning. In: Proceedings of ACL 2018, pp. 2137–2147 (2018)
13. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS, vol. 6323, pp. 148–163. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15939-8_10
14. Ritter, A., Zettlemoyer, L., Etzioni, O., et al.: Modeling missing data in distant supervision for information extraction. Trans. Assoc. Comput. Linguist. **1**, 367–378 (2013)
15. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: Proceedings of EMNLP 2012, pp. 455–465 (2012)
16. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning. MIT Press, Cambridge (1998)
17. Watkins, C.J., Dayan, P.: Q-learning. Mach. Learn. **8**(3–4), 279–292 (1992)
18. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of EMNLP 2015, pp. 1753–1762 (2015)
19. Zeng, X., He, S., Liu, K., Zhao, J.: Large scaled relation extraction with reinforcement learning. In: Proceedings of AAAI 2018 (2018)
20. Zhou, G., Su, J., Jie, Z., Zhang, M.: Exploring various knowledge in relation extraction. In: Proceedings of ACL 2005, pp. 427–434 (2005)