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

Abstract

Distant supervision can effectively label data for relation extraction, but suffers from the noise labeling problem. Recent works mainly perform soft bag-level noise reduction strategies to find the relatively better samples in a sentence bag, which is suboptimal compared with making a hard decision of false positive samples in sentence level. In this paper, we introduce an adversarial learning framework, which we named DSGAN, to learn a sentence-level true-positive generator. Inspired by Generative Adversarial Networks, we regard the positive samples generated by the generator as the negative samples to train the discriminator. The optimal generator is obtained until the discrimination ability of the discriminator has the greatest decline. We adopt the generator to filter distant supervision training dataset and redistribute the false positive instances into the negative set, in which way to provide a cleaned dataset for relation classification. The experimental results show that the proposed strategy significantly improves the performance of distant supervision relation extraction comparing to state-of-the-art systems.

1 Introduction

Relation extraction is a crucial task in the field of natural language processing (NLP). It has a wide range of applications including information retrieval, question answering, and knowledge base completion. The goal of relation extraction system is to predict relation between entity pair in a sentence (Zelenko et al., 2003; Bunescu and Mooney, 2005; GuoDong et al., 2005). For example, given a sentence "The $[owl]_{e1}$ held the mouse in its $[claw]_{e2}$.", a relation classifier should figure out the relation **Component-Whole** between entity owl and $claw$.

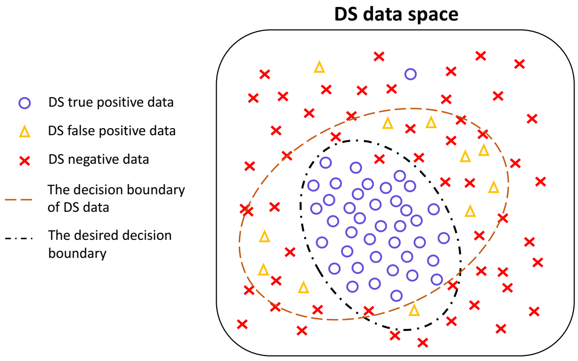

Figure 1: Illustration of the distant supervision training data distribution for one relation type.

With the infinite amount of facts in real world, it is extremely expensive, and almost impossible for human annotators to annotate training dataset to meet the needs of all walks of life. This problem has received increasingly attention. Few-shot learning and Zero-shot Learning (Xian et al., 2017) try to predict the unseen classes with few labeled data or even without labeled data. Differently, distant supervision (Mintz et al., 2009; Hoffmann et al., 2011; Surdeanu et al., 2012) is to efficiently generate relational data from plain text for unseen relations with distant supervision (DS). However, it naturally brings with some defects: the resulted distantly-supervised training samples are often very noisy (shown in Figure 1), which is the main problem of impeding the performance (Roth et al., 2013). Most of the current state-of-the-art methods (Zeng et al., 2015; Lin et al., 2016) make the denoising operation in the sentence bag of entity pair, and integrate this process into the distant supervision relation ex-

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 496–505
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

traction. Indeed, these methods can filter a substantial number of noise samples; However, they overlook the case that all sentences of an entity pair are false positive, which is also the common phenomenon in distant supervision datasets. Under this consideration, an independent and accurate **sentence-level** noise reduction strategy is the better choice.

In this paper, we design an adversarial learning process (Goodfellow et al., 2014; Radford et al., 2015) to obtain a sentence-level generator that can recognize the true positive samples from the noisy distant supervision dataset without any supervised information. In Figure 1, the existence of false positive samples makes the DS decision boundary suboptimal, therefore hinders the performance of relation extraction. However, in terms of quantity, the true positive samples still occupy most of the proportion; this is the prerequisite of our method. Given the discriminator that possesses the decision boundary of DS dataset (the brown decision boundary in Figure 1), the generator tries to generate true positive samples from DS positive dataset; Then, we assign the generated samples with negative label and the rest samples with positive label to challenge the discriminator. Under this adversarial setting, if the 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. Empirically, we show that our method has brought consistent performance gains in various deep-neural-network-based models, achieving strong performances on the widely used New York Times dataset (Riedel et al., 2010). Our contributions are three-fold:

- We are the first to consider adversarial learning to denoise the distant supervision relation extraction dataset.
- Our method is sentence-level and model-agnostic, so it can be used as a plug-and-play technique for any relation extractors.
- We show that our method can generate a cleaned dataset without any supervised information, in which way to boost the performance of recently proposed neural relation extractors.

In Section 2, we outline some related works on distant supervision relation extraction. Next, we describe our adversarial learning strategy in Section 3. In Section 4, we show the stability analyses of DSGAN and the empirical evaluation results. And finally, we conclude in Section 5.

2 Related Work

To address the above-mentioned data sparsity issue, Mintz et al. (2009) first align unlabeled text corpus with Freebase by distant supervision. However, distant supervision inevitably suffers from the wrong labeling problem. Instead of explicitly removing noisy instances, the early works intend to suppress the noise. Riedel et al. (2010) adopt multi-instance single-label learning in relation extraction; Hoffmann et al. (2011) and Surdeanu et al. (2012) model distant supervision relation extraction as a multi-instance multi-label problem.

Recently, some deep-learning-based models (Zeng et al., 2014; Shen and Huang, 2016) have been proposed to solve relation extraction. Naturally, some works try to alleviate the wrong labeling problem with deep learning technique, and their denoising process is integrated into relation extraction. Zeng et al. (2015) select one most plausible sentence to represent the relation between entity pairs, which inevitably misses some valuable information. Lin et al. (2016) calculate a series of soft attention weights for all sentences of one entity pair and the incorrect sentences can be down-weighted; Base on the same idea, Ji et al. (2017) bring the useful entity information into the calculation of the attention weights. However, compared to these soft attention weight assignment strategies, recognizing the true positive samples from distant supervision dataset before relation extraction is a better choice. Takamatsu et al. (2012) build a noise-filtering strategy based on the linguistic features extracted from many NLP tools, including NER and dependency tree, which inevitably suffers the error propagation problem; while we just utilize word embedding as the input information. In this work, we learn a true-positive identifier (the generator) which is independent of the relation prediction of entity pairs, so it can be directly applied on top of any existing relation extraction classifiers. Then, we redistribute the false positive samples into the negative set, in which way to make full use of the distantly labeled resources.

3 Adversarial Learning for Distant Supervision

In this section, we introduce an adversarial learning pipeline to obtain a robust generator which can automatically discover the true positive samples from the noisy distantly-supervised dataset without any supervised information. The overview of our adversarial learning process is shown in Figure 2. Given a set of distantly-labeled sentences, the generator tries to generate true positive samples from it; But, these generated samples are regarded as negative samples to train the discriminator. Thus, when finishing scanning the DS positive dataset one time, the more true positive samples that the generator discovers, the sharper drop of performance the discriminator obtains. After adversarial training, we hope to obtain a robust generator that is capable of forcing discriminator into maximumly losing its classification ability.

In the following section, we describe the adversarial training pipeline between the generator and the discriminator, including the pre-training strategy, objective functions and gradient calculation. Because the generator involves a discrete sampling step, we introduce a policy gradient method to calculate gradients for the generator.

3.1 Pre-Training Strategy

Both the generator and the discriminator require the pre-training process, which is the common setting for GANs (Cai and Wang, 2017; Wang et al., 2017). With the better initial parameters, the adversarial learning is prone to convergence. As presented in Figure 2, the discriminator is pre-trained with DS positive dataset P (label 1) and DS negative set N^D (label 0). After our adversarial learning process, we desire a strong generator that can, to the maximum extent, collapse the discriminator. Therefore, the more robust generator can be obtained via competing with the more robust discriminator. So we pre-train the discriminator until the accuracy reaches 90% or more. The pre-training of generator is similar to the discriminator; however, for the negative dataset, we use another completely different dataset N^G, which makes sure the robustness of the experiment. Specially, we let the generator overfits the DS positive dataset P. The reason of this setting is that we hope the generator wrongly give high probabilities to all of the noisy DS positive samples at the beginning of the training process. Then, along with our adversarial learning, the generator learns to gradually decrease the probabilities of the false positive samples.

3.2 Generative Adversarial Training for Distant Supervision Relation Extraction

The generator and the discriminator of DSGAN are both modeled by simple CNN, because CNN performs well in understanding sentence (Zeng et al., 2014), and it has less parameters than RNN-based networks. For relation extraction, the input information consists of the sentences and entity pairs; thus, as the common setting (Zeng et al., 2014; Nguyen and Grishman, 2015), we use both word embedding and position embedding to convert input instances into continuous real-valued vectors.

What we desire the generator to do is to accurately recognize true positive samples. Unlike the generator applied in computer vision field (Im et al., 2016) that generates new image from the input noise, our generator just needs to discover true positive samples from the noisy DS positive dataset. Thus, it is to realize the "sampling from a probability distribution" process of the discrete GANs (Figure 2). For a input sentence s_j, we define the probability of being true positive sample by generator as $p_G(s_j)$. Similarly, for discriminator, the probability of being true positive sample is represented as $p_D(s_j)$. We define that one epoch means that one time scanning of the entire DS positive dataset. In order to obtain more feedbacks and make the training process more efficient, we split the DS positive dataset $P = \{s_1, s_2, ..., s_j, ...\}$ into N bags $B = \{B^1, B^2, ...B^N\}$, and the network parameters θ_G, θ_D are updated when finishing processing one bag B_i[1]. Based on the notion of adversarial learning, we define the objectives of the generator and the discriminator as follow, and they are alternatively trained towards their respective objectives.

Generator Suppose that the generator produces a set of probability distribution $\{p_G(s_j)\}_{j=1...|B_i|}$ for a sentence bag B_i. Based on these probabilities, a set of sentence are sampled and we denote this set as T.

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

[1] The *bag* here has the different definition from the sentence *bag* of an entity pair mentioned in the Section 1.

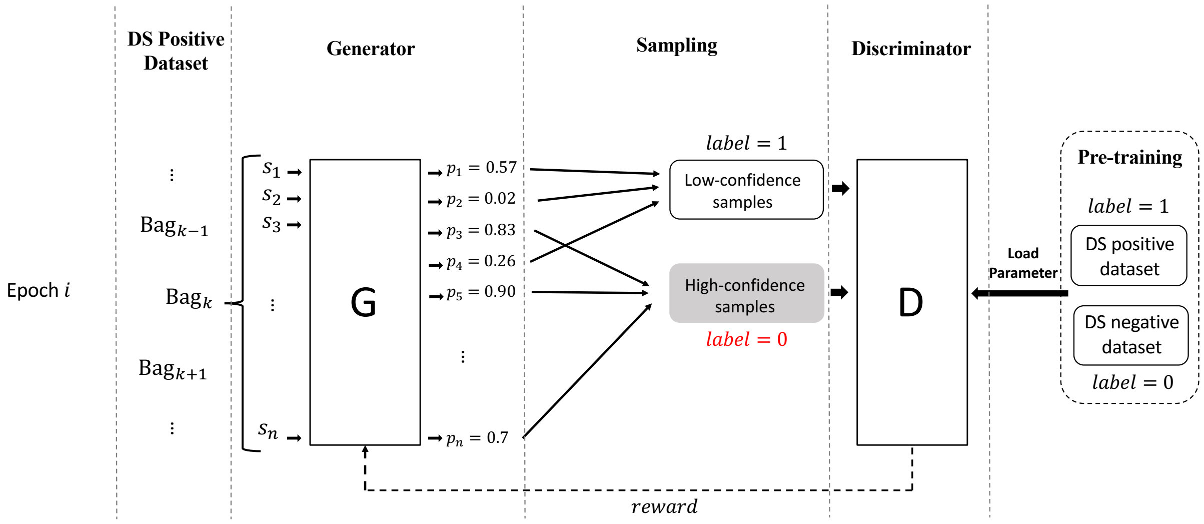

Figure 2: An overview of the DSGAN training pipeline. The generator (denoted by **G**) calculates the probability distribution over a bag of DS positive samples, and then samples according to this probability distribution. The high-confidence samples generated by G are regarded as true positive samples. 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.

This generated dataset T consists of the high-confidence sentences, and is regard as true positive samples by the current generator; however, it will be treated as the negative samples to train the discriminator. In order to challenge the discriminator, the objective of the generator can be formulated as **maximizing** the following probabilities of the generated dataset T:

$$L_G = \sum_{s_j \in T} \log p_D(s_j) \tag{2}$$

Because L_G involves a discrete sampling step, so it cannot be directly optimized by gradient-based algorithm. We adopt a common approach: the policy-gradient-based reinforcement learning. The following section will give the detailed introduction of the setting of reinforcement learning. The parameters of the generator are continually updated until reaching the convergence condition.

Discriminator After the generator has generated the sample subset T , the discriminator treats them as the negative samples; conversely, the rest part $F = B_i - T$ is treated as positive samples. So, the objective of the discriminator can be formulated as **minimizing** the following cross-entropy loss function:

$$\begin{aligned} L_D = -(& \sum_{s_j \in (B_i - T)} \log p_D(s_j) \\ & + \sum_{s_j \in T} \log(1 - p_D(s_j))) \end{aligned} \tag{3}$$

The update of discriminator is identical to the common binary classification problem. Naturally, it can be simply optimized by any gradient-based algorithm.

What needs to be explained is that, unlike the common setting of discriminator in previous works, our discriminator *loads the same pretrained parameter set at the beginning of each epoch* as shown in Figure 2. There are two reasons. First, at the end of our adversarial training, what we need is a robust generator rather than a discriminator. Second, our generator is to sample data rather than generate new data from scratch; Therefore, the discriminator is relatively easy to be collapsed. So we design this new adversarial strategy: the robustest generator is yielded when the discriminator has the largest drop of performance in one epoch. In order to create the equal condition, the bag set B for each epoch is identical, including the sequence and the sentences in each

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 $\theta_G$ on dataset ($P$, $N^G$); Pre-trained D with parameters $\theta_D$ on
    dataset ($P$, $N^D$)
Output: Adversarially trained generator G
Load parameters $\theta_G$ for G
Split $P$ into the bag sequence $P = \{B^1, B^2, ..., B^i, ..., B^N\}$
repeat
    Load parameters $\theta_D$ 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 $F =
    B_i - T$
        $G_D \leftarrow -\frac{1}{|P|}\{\nabla_{\theta_D}\sum^T \log(1 - p_D(s_j)) + \nabla_{\theta_D}\sum^F \log p_D(s_j)\}$
        $\theta_D \leftarrow \theta_D - \alpha_D G_D$
        Calculate the reward $r$
        $G_G \leftarrow \frac{1}{|T|}\sum^T r\nabla_{\theta_G}\log p_G(s_j)$
        $\theta_G \leftarrow \theta_G + \alpha_G G_G$
    end for
    Compute the accuracy $ACC_D$ on $N^D$ with the current $\theta_D$
until $ACC_D$ no longer drops
Save $\theta_G$
```

bag B_i.

Optimizing Generator The objective of the generator is similar to the objective of the one-step reinforcement learning problem: Maximizing the expectation of a given function of samples from a parametrized probability distribution. Therefore, we use a policy gradient strategy to update the generator. Corresponding to the terminology of reinforcement learning, s_j is the *state* and $P_G(s_j)$ is the *policy*. In order to better reflect the quality of the generator, we define the reward r from two angles:

- As the common setting in adversarial learning, for the generated sample set, we hope the confidence of being positive samples by the discriminator becomes higher. Therefore, the first component of our reward is formulated as below:

$$r_1 = \frac{1}{|T|}\sum_{s_j \in T} p_D(s_j) - b_1 \tag{4}$$

  the function of b_1 is to reduce variance during reinforcement learning.

- The second component is from the average prediction probability of N^D,

$$\tilde{p} = \frac{1}{|N^D|}\sum_{s_j \in N^D} p_D(s_j) \tag{5}$$

  N^D participates the pre-training process of the discriminator, but not the adversarial training process. When the classification capacity of discriminator declines, the accuracy of being predicted as negative sample on N^D gradually drops; thus, $\tilde{p}$ increases. In other words, the generator becomes better. Therefore, for epoch k, after processing the bag B_i, reward r_2 is calculated as below,

$$\begin{aligned} r_2 &= \eta(\tilde{p}_i^k - b_2) \\ where\ b_2 &= \max\{\tilde{p}_i^m\}, m=1..., k-1 \end{aligned} \tag{6}$$

  b_2 has the same function as b_1.

The gradient of L_G can be formulated as below:

$$\begin{aligned} \nabla_{\theta_D} L_G &= \sum_{s_j \in B_i} \mathbb{E}_{s_j \sim p_G(s_j)} r \nabla_{\theta_G} \log p_G(s_j) \\ &= \frac{1}{|T|}\sum_{s_j \in T} r \nabla_{\theta_G} \log p_G(s_j) \end{aligned} \tag{7}$$

3.3 Cleaning Noisy Dataset with Generator

After our adversarial learning process, we obtain one generator for one relation type; These generators possess the capability of generating true positive samples for the corresponding relation type. Thus, we can adopt the generator to filter the noise samples from distant supervision dataset. Simply and clearly, we utilize the generator as a binary classifier. In order to reach the maximum utilization of data, we develop a strategy: for an entity pair with a set of annotated sentences, if all of these sentences are determined as false negative by our generator, this entity pair will be redistributed into the negative set. Under this strategy, the scale of distant supervision training set keeps unchanged.

4 Experiments

This paper proposes an adversarial learning strategy to detect true positive samples from the noisy distant supervision dataset. Due to the absence of supervised information, we define a generator to heuristically learn to recognize true positive samples through competing with a discriminator. Therefore, our experiments are intended to demonstrate that our DSGAN method possess this capability. To this end, we first briefly introduce the dataset and the evaluation metrics. Empirically, the adversarial learning process, to some extent, has instability; Therefore, we next illustrate the convergence of our adversarial training process. Finally, we demonstrate the efficiency of our generator from two angles: the quality of the generated samples and the performance on the widely-used distant supervision relation extraction task.

4.1 Evaluation and Implementation Details

The Reidel dataset[2] (Riedel et al., 2010) is a commonly-used distant supervision relation extraction dataset. Freebase is a huge knowledge base including billions of triples: the entity pair and the specific relationship between them. Given these triples, the sentences of each entity pair are selected from the New York Times corpus(NYT). Entity mentions of NYT corpus are recognized by the Stanford named entity recognizer (Finkel et al., 2005). There are 52 actual relationships and a special relation NA which indicates there is no relation between head and tail entities. Entity pairs of NA are defined as the entity pairs that appear in the same sentence but are not related according to Freebase.

Hyperparameter	Value
CNN Window c_w, kernel size c_k	3, 100
Word embedding d_e, $\|V\|$	50, 114042
Position embedding d_p	5
Learning rate of G, D	1e-5, 1e-4

Table 1: Hyperparameter settings of the generator and the discriminator.

Due to the absence of the corresponding labeled dataset, there is not a ground-truth test dataset to evaluate the performance of distant supervision relation extraction system. Under this circumstance, the previous work adopt the held-out evaluation to evaluate their systems, which can provide an approximate measure of precision without requiring costly human evaluation. It builds a test set where entity pairs are also extracted from Freebase. Similarly, relation facts that discovered from test articles are automatically compared with those in Freebase. CNN is widely used in relation classification (Santos et al., 2015; Qin et al., 2017), thus the generator and the discriminator are both modeled as a simple CNN with the window size c_w and the kernel size c_k. Word embedding is directly from the released word embedding matrix by Lin et al. (2016)[3]. Position embedding has the same setting with the previous works: the maximum distance of -30 and 30. Some detailed hyperparameter settings are displayed in Table 1.

4.2 Training Process of DSGAN

Because adversarial learning is widely regarded as an effective but unstable technique, here we illustrate some property changes during the training process, in which way to indicate the learning trend of our proposed approach. We use 3 relation types as the examples: */business/person/company*, */people/person/place_lived* and */location/neighborhood/neighborhood_of*. Because they are from three major classes (*bussiness*, *people*, *location*) of Reidel dataset and they all have enough distant-supervised instances. The first row in Figure 3 shows the classification ability change of the discriminator during training. The accuracy is calculated from the negative set N^D. At the beginning of adversarial learning, the

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

[3] https://github.com/thunlp/NRE

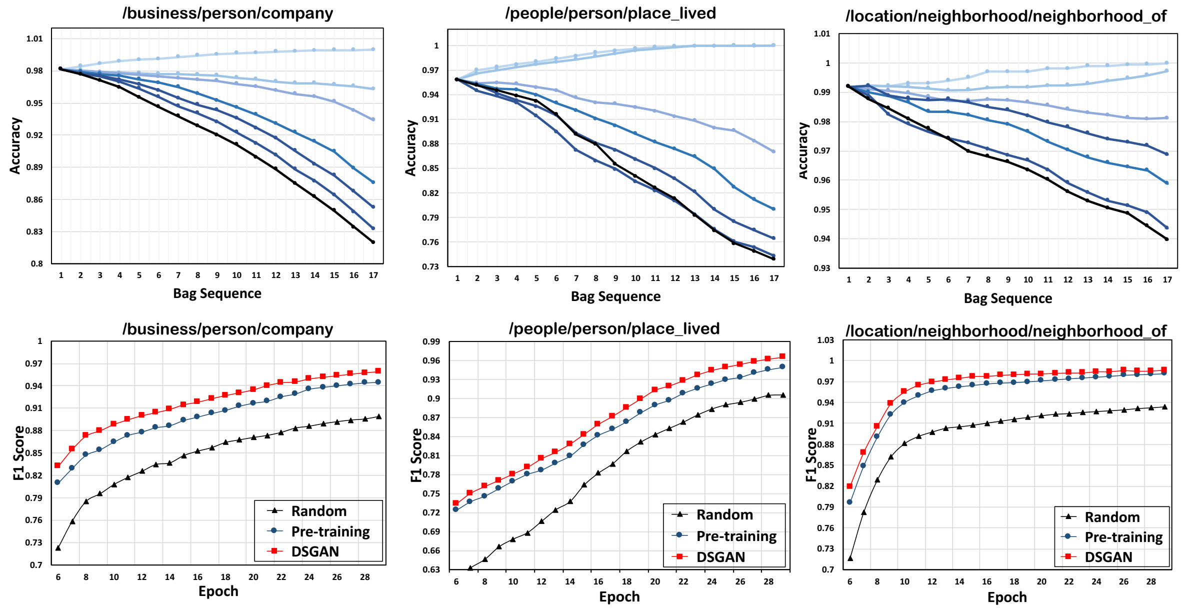

Figure 3: The convergence of the DSGAN training process for 3 relation types and the performance of their corresponding generators. The figures in the first row present the performance change on N^D in some specific epochs during processing the $B = \{B^1, B^2, ...B^N\}$. Each curve stands for one epoch; 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.

discriminator performs well on N^D; moreover, N^D is not used during adversarial training. Therefore, the accuracy on N^D is the criterion to reflect the performance of the discriminator. In the early epochs, the generated samples from the generator increases the accuracy, because it has not possessed the ability of challenging the discriminator; however, as the training epoch increases, this accuracy gradually decreases, which means the discriminator becomes weaker. It is because the generator gradually learn to generate more accurate true positive samples in each bag. After the proposed adversarial learning process, the generator is strong enough to collapse the discriminator. Figure 4 gives more intuitive display of the trend of accuracy. Note that there is a critical point of the decline of accuracy for each presented relation types. It is because that the chance we give the generator to challenge the discriminator is just one time scanning of the noisy dataset; this critical point is yielded when the generator has already been robust enough. Thus, we stop the training process when the model reaches this critical point. To sum up, the capability of our generator can steadily increases, which indicates that DSGAN is a robust adversarial learning strategy.

4.3 Quality of Generator

Due to the absence of supervised information, we validate the quality of the generator from another angle. Combining with Figure 1, for one relation type, the true positive samples must have evidently higher relevance (the cluster of purple circles). Therefore, a positive set with more true positive samples is easier to be trained; In other words, the convergence speed is faster and the fitting degree on training set is higher. Based on this , we present the comparison tests in the second row of Figure 3. We build three positive

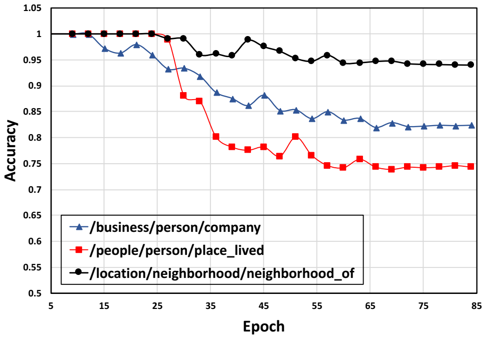

Figure 4: The performance change of the discriminator on N^D during the training process. Each point in the curves records the prediction accuracy on N^D when finishing each epoch. We stop the training process when this accuracy no longer decreases.

datasets from the noisy distant supervision dataset P: the randomly-selected positive set, the positive set base on the pre-trained generator and the positive set base on the DSGAN generator. For the pre-trained generator, the positive set is selected according to the probability of being positive from high to low. These three sets have the same size and are accompanied by the same negative set. Obviously, the positive set from the DSGAN generator yields the best performance, which indicates that our adversarial learning process is able to produce a robust true-positive generator. In addition, the pre-trained generator also has a good performance; however, compared with the DSGAN generator, it cannot provide the boundary between the false positives and the true positives.

4.4 Performance on Distant Supervision Relation Extraction

Based on the proposed adversarial learning process, we obtain a generator that can recognize the true positive samples from the noisy distant supervision dataset. Naturally, the improvement of distant supervision relation extraction can provide a intuitive evaluation of our generator. We adopt the strategy mentioned in Section 3.3 to relocate the dataset. After obtaining this redistributed dataset, we apply it to train the recent state-of-the-art models and observe whether it brings further improvement for these systems. Zeng et al. (2015) and Lin et al. (2016) are both the robust models to solve wrong labeling problem of distant supervision relation extraction. According to the comparison displayed in Figure 5 and Figure 6, all four models (*CNN+ONE*, *CNN+ATT*, *PCNN+ONE* and *PCNN+ATT*) achieve further improvement.

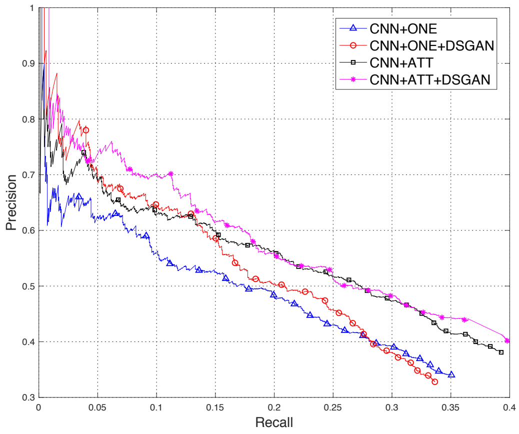

Figure 5: Aggregate PR curves of CNN˙based model.

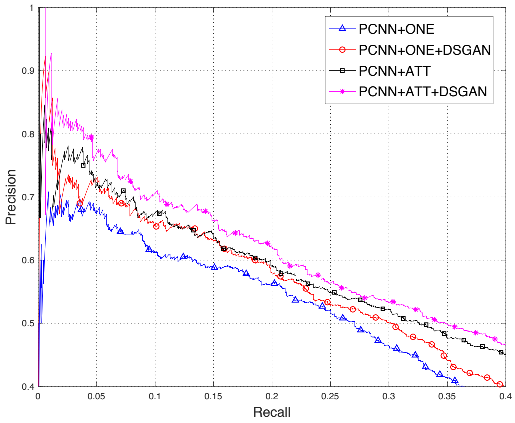

Figure 6: Aggregate PR curves of PCNN˙based model.

Even though Zeng et al. (2015) and Lin et al. (2016) are designed to alleviate the influence of false positive samples, both of them merely focus on the noise filtering in the sentence bag of entity pairs. Zeng et al. (2015) combine at-least-one multi-instance learning with deep neural network to extract only one active sentence to represent the target entity pair; Lin et al. (2016) assign soft attention weights to the representations of all sentences of one entity pair, then employ the weighted sum of these representations to predict the relation between the target entity pair. However, from our manual inspection of Riedel dataset (Riedel et al., 2010), we found another false positive case that all the sentences of a specific entity pair are wrong; but the aforementioned methods overlook

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.

this case, while the proposed method can solve this problem. Our DSGAN pipeline is independent of the relation prediction of entity pairs, so we can adopt our generator as the true-positive indicator to filter the noisy distant supervision dataset before relation extraction, which explains the origin of these further improvements in Figure 5 and Figure 6. In order to give more intuitive comparison, in Table 2, we present the AUC value of each PR curve, which reflects the area size under these curves. The larger value of AUC reflects the better performance. Also, as can be seen from the result of t-test evaluation, all the p-values are less than 5e-02, so the improvements are obvious.

5 Conclusion

Distant supervision has become a standard method in relation extraction. However, while it brings the convenience, it also introduces noise in distantly labeled sentences. In this work, we propose the first generative adversarial training method for robust distant supervision relation extraction. More specifically, our framework has two components: a generator that generates true positives, and a discriminator that tries to classify positive and negative data samples. With adversarial training, our goal is to gradually decrease the performance of the discriminator, while the generator improves the performance for predicting true positives when reaching equilibrium. Our approach is model-agnostic, and thus can be applied to any distant supervision model. Empirically, we show that our method can significantly improve the performances of many competitive baselines on the widely used New York Time dataset.

Acknowledge

This work was supported by National Natural Science Foundation of China (61702047), Beijing Natural Science Foundation (4174098), the Fundamental Research Funds for the Central Universities (2017RC02) and National Natural Science Foundation of China (61703234)

References

Razvan Bunescu and Raymond J Mooney. 2005. Subsequence kernels for relation extraction. In *NIPS*, pages 171–178.

Liwei Cai and William Yang Wang. 2017. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

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: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic. 2016. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*.

Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao, et al. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *AAAI*, pages 3060–3066.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL (1)*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *ACL (2)*, pages 365–371.

Pengda Qin, Weiran Xu, and Jun Guo. 2017. Designing an adaptive attention mechanism for relation classification. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 4356–4362. IEEE.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. 2013. A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 73–78. ACM.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*.

Yatian Shen and Xuanjing Huang. 2016. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*.

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. Association for Computational Linguistics.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 721–729. Association for Computational Linguistics.

Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 515–524. ACM.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning-the good, the bad and the ugly. *arXiv preprint arXiv:1703.04394*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal*, pages 17–21.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.