Copyright © 2018 Tech Science Press CMC, vol.55, no.1, pp.121-136, 2018

Adversarial Learning for Distant Supervised Relation Extraction

Daojian Zeng[1, 3], Yuan Dai[1, 3], Feng Li[1, 3], R. Simon Sherratt[2] and Jin Wang[3, *]

Abstract: Recently, many researchers have concentrated on using neural networks to learn features for Distant Supervised Relation Extraction (DSRE). These approaches generally use a softmax classifier with cross-entropy loss, which inevitably brings the noise of artificial class *NA* into classification process. To address the shortcoming, the classifier with ranking loss is employed to DSRE. Uniformly randomly selecting a relation or heuristically selecting the highest score among all incorrect relations are two common methods for generating a negative class in the ranking loss function. However, the majority of the generated negative class can be easily discriminated from positive class and will contribute little towards the training. Inspired by Generative Adversarial Networks (GANs), we use a neural network as the negative class generator to assist the training of our desired model, which acts as the discriminator in GANs. Through the alternating optimization of generator and discriminator, the generator is learning to produce more and more discriminable negative classes and the discriminator has to become better as well. This framework is independent of the concrete form of generator and discriminator. In this paper, we use a two layers fully-connected neural network as the generator and the Piecewise Convolutional Neural Networks (PCNNs) as the discriminator. Experiment results show that our proposed GAN-based method is effective and performs better than state-of-the-art methods.

Keywords: Relation extraction, generative adversarial networks, distant supervision, piecewise convolutional neural networks, pair-wise ranking loss.

1 Introduction

There have been many methods proposed for relation extraction. In these methods, the supervised paradigm has been shown to be effective and yields relatively high performance [Kambhatla (2004); Zhou, Su, Zhang et al. (2005)]. However, a large labeled training data is often required for supervision, and manually annotating large labeled training data is a time-consuming and labor-intensive task. In addition, since the

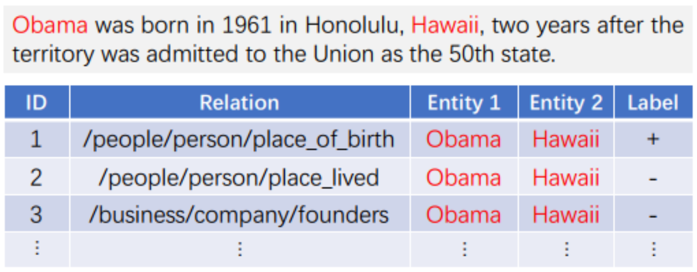
[1] Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, 410114, P. R. China. Email: zengdj@csust.edu.cn.

[2] Department of Biomedical Engineering, University of Reading, UK. Email: r.s.sherratt@reading.ac.uk.

[3] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, 410114, P. R. China. Email: jinwang@csust.edu.cn.

* Corresponding author: Jin Wang. Email: jinwang@csust.edu.cn.

manually labeled training data is often domain dependent, the model tends to be biased toward a specific domain.



Obama was born in 1961 in Honolulu, Hawaii, two years after the territory was admitted to the Union as the 50th state.

ID	Relation	Entity 1	Entity 2	Label
1	/people/person/place_of_birth	Obama	Hawaii	+
2	/people/person/place_lived	Obama	Hawaii	-
3	/business/company/founders	Obama	Hawaii	-
⋮	⋮	⋮	⋮	⋮

Figure 1: An training example

To address the shortcomings of supervised paradigm, distantly supervised paradigm [Mintz, Bills, Snow et al. (2009)] is proposed to automatically generate training data. After obtaining the distant supervised labeled data, traditional methods sometimes applied supervised models to elaborately handcrafted features [Mintz, Bills, Snow et al. (2009); Riedel, Yao and McCallum (2010); Hoffmann, Zhang, Ling et al. (2011); Surdeanu, Tibshirani, Nallapati et al. (2012)]. These features are often derived from off-the-shelf Natural Language Processing (NLP) tools, which inevitably have errors, and have negative impact on the classification accuracy.

With the recent revival of interest in neural networks, many researchers have investigated the possibility of using neural networks to automatically learn features for relation classification [Zeng, Liu, Lai et al. (2014); Zeng, Liu, Chen et al. (2015); Jiang, Wang, Li et al. (2016); Lin, Shen, Liu et al. (2016)]. These approaches generally use a softmax classifier with cross-entropy loss and inevitably bring the noise of artificial class *NA* into classification process. To address the shortcoming, the classifier with ranking loss is employed to relation extraction [Zeng, Zeng and Dai (2017)]. Generally, the ranking loss function needs a negative class to train the model. Randomly selecting a relation or selecting the highest score among all incorrect relations are two common methods for generating negative class.

Unfortunately, these approaches are not ideal, because the sampled relation could be completely unrelated to the two given entities, so the majority of the generated negative class can be easily discriminated from positive class. Thus the quality of selected negative class is often poor and will contribute little towards the training. Cai et al. [Cai and Wang (2017)] and [Jia and Liang (2017)] have found that the quality of the negative class (pattern) is very important in the discriminative architecture. For example, the positive relation between Obama and Hawaii is */people/person/place_of birth* in Fig. 1 (ID 1). Obviously, there can be no */business/company/founders* relation between a Person and a Location and it is a poor negative relation (ID 3). Accordingly, the birthplace of a person is probably also the place where he lives. Thus, */people/person/place_lived* is the high-quality one that can be used to improve the model's discrimination (ID 2).

In this paper, we provide a generic solution to improve the training of ranking based DSRE. Inspired by generative adversarial networks (GANs) [Goodfellow, Pouget-Abadie, Mirza et al. (2014)], we propose a novel adversarial learning framework for this task and use a neural network as the negative label generator to assist the training of our desired model, which acts as the discriminator in GANs. More specifically, we consider a two layers fully connected neural network as the generator to supply better quality negative labels at first. Then we adopt the PCNNs as the discriminator to classify the final relation. Through the alternating optimization of generator and discriminator, the generator is learning to produce more and more discriminable negative classes and the discriminator has to become better as well. Since the generator has a discrete generation step, we cannot directly use the gradient-based approach to back propagate the errors. We then consider a one-step reinforcement learning setting and use a REINFORCE method to achieve this goal [Watkins (1992)].

In sum, the main contributions of this paper lie in three folds:

- We combine GAN with the ranking-based approach and propose a new paradigm for DSRE. To our best knowledge, this is the first attempt to consider adversarial learning for this task.
- We prove that the generator can consistently provide high-quality negative classes which is crucial for the discriminator to improve DSRE.
- Empirically, we perform experiments on a widely used dataset and verify the adversarial learning approach. Experimental results show that our approach obtains state-of-the-art performance on the dataset.

2 Related work

2.1 Relation extraction

Relation extraction is one of the most important topics in NLP. Supervised approaches are the most commonly used methods for relation extraction and yield relatively high performance [Bunescu and Mooney (2005); Zelenko, Aone and Richardella (2003); Zhou, Su, Zhang et al. (2005)]. In the supervised paradigm, relation extraction is considered to be a multi-class classification problem and may suffer from a lack of labeled data for training. To address this issue, Mintz et al. [Mintz, Bills, Snow et al. (2009)] adopts Freebase to perform distant supervision. The algorithm for training data generation is sometimes faced with the wrong label problem. To address this shortcoming, Riedel et al. [Riedel, Yao and McCallum (2010); Hoffmann, Zhang, Ling et al. (2011); Surdeanu, Tibshirani, Nallapati et al. (2012)] develop the relaxed distant supervision assumption for multi-instance learning. Nguyen et al. [Nguyen and Moschitti (2011)] utilize relation definitions and Wikipedia documents to improve their systems.

The methods mentioned above have been shown to be effective for DSRE. However, their performance depends strongly on the quality of the designed features. Recently, deep learning has made great strides in many tasks [Gurusamy and Subramaniam (2017); Yuan, Li, Wu et al. (2017)]. Many researchers attempt to use deep neural network to automatically learning feature for DSRE. Zeng et al. [Zeng, Liu, Lai et al. (2014)] adopts CNNs to embed the semantics of the sentences. Moreover, Santos et al. [Santos, Xiang and Zhou (2015)] proposes a pairwise ranking loss function in the CNNs to reduce the impact of artificial class. These methods build classifier based on sentence-level annotated data, which cannot be directly applied for DSRE since multiple sentences corresponding to a fact may be achieved in the data generating procedure. Therefore, Zeng et al. [Zeng, Liu, Chen et al. (2015)] incorporate multi-instance learning with neural network model, which can build relation extractor based on distant supervision data. Although the method achieves significant improvement in relation extraction, it only selects the most likely sentence for each entity pair in their multi-instance learning paradigm. To address this issue, Lin et al. [Lin, Shen, Liu et al. (2016)] propose sentence level attention over multiple instances in order to utilize all informative sentences. Jiang et al. [Jiang, Wang, Li et al. (2016)] employ cross-sentence max-pooling to select features across different instances and then aggregates the most significant features for each entity pair.

The aforementioned works, especially neural networks, have greatly promoted the development of relation extraction. However, these works do not pay attention to the noise of artificial class *NA*, which are unfortunately very common in DSRE. Zeng et al. [Zeng, Zeng and Dai (2017)] proposed ranking loss and cost-sensitive to address the noise of *NA*. They select the highest score among all incorrect relations as the negative label. This approach is not ideal, because the quality of the selected label is often poor. In this paper, we propose a novel pair-wise ranking loss whose negative samples are provided by a generator of the GAN.

2.2 Generative adversarial networks

GANs [Goodfellow, Pouget-Abadie, Mirza et al. (2014)] was originally proposed for generating samples in a continuous space such as images. A GAN consists of two parts, the generator, and the discriminator. The generator accepts a noise input and outputs an image. The discriminator is a classifier which classifies images as “true” (from the ground truth set) or “fake” (generated by the generator). When training a GAN, the generator and the discriminator play a minimax game, in which the generator tries to generate “real” images to deceive the discriminator, and the discriminator tries to tell them apart from ground truth images. GANs are also capable of generating samples satisfying certain requirements, such as conditional GAN [Mirza and Osindero (2014)]. It is not possible to use GANs in its original form for generating discrete samples like natural language sentences or knowledge graph triples, because the discrete sampling step prevents gradients from propagating back to the generator. SeqGAN [Yu, Zhang, Wang et al. (2017)] is one of the first successful solutions to this problem by using reinforcement learning, which trains the generator using policy gradient. Likewise, our framework relies on policy gradient to train the generator which provides discrete negative labels.

3 Task definition

DSRE is usually considered as a multi-instance learning problem. In multi-instance learning paradigm, all sentences labeled by a relation triplet constitute a bag and each sentence is called an instance.

Suppose that there are N bags $\{B_1, B_2, \cdots, B_N\}$ in the training set and that the i-th bag B_i contains q_i instances $B_i = \{b_1^i, b_2^i, \cdots, b_{q_i}^i\} (i = 1, \cdots, N)$.The objective of multi-instance learning is to predict the labels of the unseen bags. We need to learn a relation extractor based on the training data and then use it to predict relations for the test set.

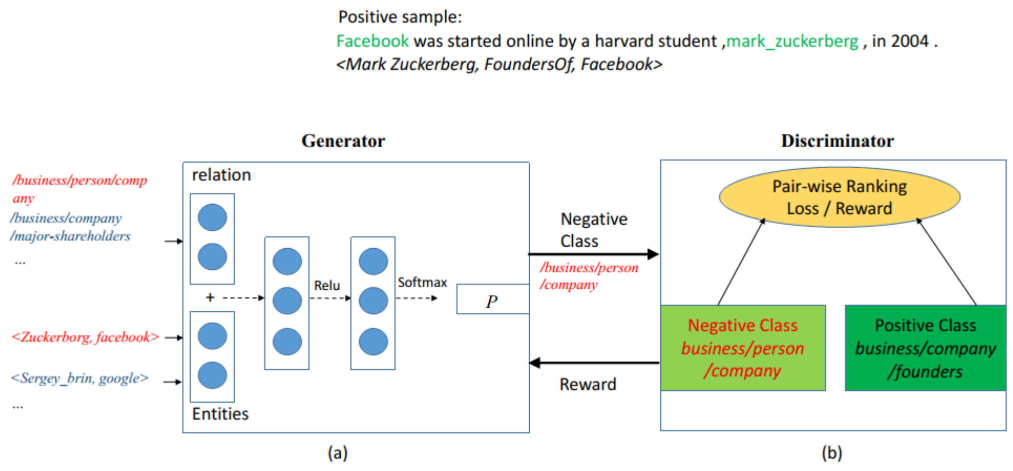

Positive sample:
Facebook was started online by a harvard student ,mark_zuckerberg , in 2004 .
<Mark Zuckerberg, FoundersOf, Facebook>
Generator
Discriminator
/business/person/company
/business/company/major-shareholders
...
<Zuckerborg, facebook>
<Sergey_brin, google>
...
relation
Entities
+
Relu
Softmax
P
Negative Class
/business/person/company
Reward
Pair-wise Ranking Loss / Reward
Negative Class business/person/company
Positive Class business/company/founders
(a)
(b)

Figure 2: GAN-based framework. (a) Generator: calculating a probability distribution over a set of candidate negative relations, then sample one relation from the distribution as the output. (b) Discriminator: receiving the generated negative triple as well as the ground truth triple (in the hexagonal box), and calculating their scores. Generator maximizes the score of the generated negative class by policy gradient, and discriminator minimizes the marginal loss between positive and negative class by gradient descent

Specifically, for a bag $B_j = \{b_1^j, b_2^j, \cdots, b_{q_j}^j\}$ in training set, we need to extract features from the bag (from one or several valid instances) and then use them to train a classifier.

4 Methodology

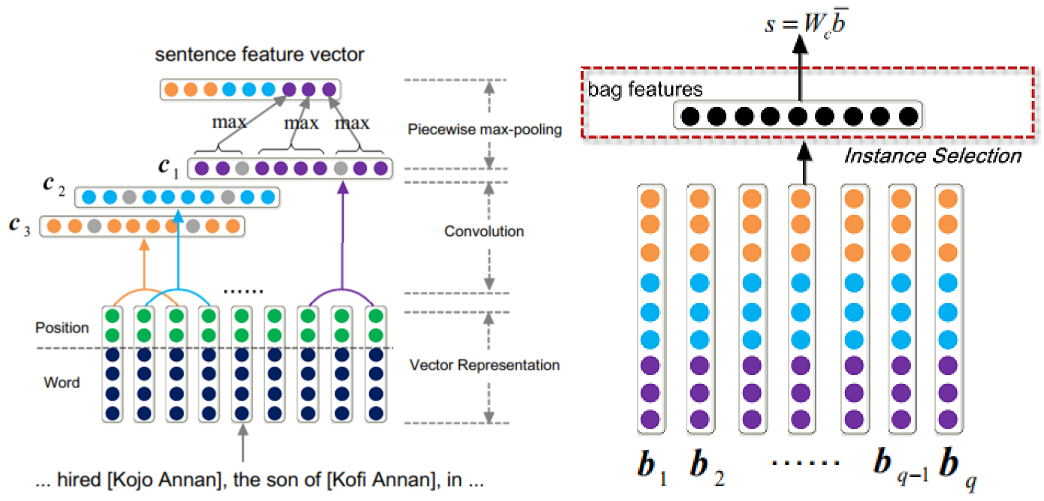
Fig. 2 shows the neural network architecture used in this work. It consists of two parts: Discriminator Network and Generator Network. In this section, we first introduce the discriminator network and generator network used in this paper. Then, we define the objective function for discriminator and generator respectively and explain how to alternatively train the proposed model.

4.1 Discriminator network

The discriminator in our model is similar to [Zeng, Zeng and Dai (2017)] which is shown in Fig. 3. Different from their model that selects the highest score among all incorrect relations as a negative class, our model uses the negative label generated by the generator which will be described in detail in Section 4.2. In this section, we will first give a detailed description of the discriminator.

4.1.1 Vector representation

The inputs of our discriminator are raw word tokens. When using neural networks, we typically transform word tokens into low-dimensional vectors. In this paper, the “word token” refers to word and entity. In the following, we do not distinguish them and call them “word”. In our method, each input word token is transformed into a vector by looking up pre-trained word embeddings. Moreover, we use Position Features (PFs) [Zeng, Liu, Lai et al. (2014); Zeng, Liu, Chen et al. (2015)] to specify entity pairs, which are also transformed into vectors by looking up position embeddings.



(a)PCNNs Model (b)Ranking Classifier

Figure 3: Discriminator network

4.1.2 Word embeddings

Word embeddings are distributed representations of words that map each word in a text to a ‘k’-dimensional real-valued vector. They have recently been shown to capture both semantic and syntactic information about words very well, setting performance records in several word similarity tasks [Mikolov, Chen, Corrado et al. (2013); Pennington, Socher and Manning (2014)]. Using word embeddings that have been trained a priori has become common practice for enhancing many other NLP tasks [Huang, Ahuja, Downey et al. (2014)]. In the past years, many methods for training word embeddings have been proposed [Bengio, Ducharme, Vincent et al. (2003); Collobert, Weston, Bottou et al. (2011); Mikolov, Chen, Corrado et al. (2013)]. We employ the method [Mikolov, Chen, Corrado et al. (2013)] to train word embeddings and denote it by $\boldsymbol{E}$.

4.1.3 Position embeddings

Zeng et al. [Zeng, Liu, Chen et al. (2015)] have shown the importance of PFs in relation extraction. Similar to their works, we use PFs to specify entity pairs. A PF is defined as the combination of the relative distances from the current word to entity e_1 and entity e_2. We randomly initialize two position embedding matrices $PF_i (i=1,2)$ (for e_1 and e_2), and transform the relative distances into vectors by looking them up.

We concatenate the word representation and position representation as the input of the network (shown in Fig. 3(a)). Assume that the size of word representation is k_w and that of position representation is k_d, then the size of a word vector is $k = k_w + 2k_d$.

4.1.4 Convolution

Assume that $A = (a_{ij})_{m\times n}$ and $B = (b_{ij})_{m\times n}$, then the convolution of *A* and *B* is defined as $A \otimes B = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}b_{ij}$.

We denote the input sentence by $S = \{S_1, S_2, \cdots, S_{|S|}\}$, where s_i is the i-th word, and use $s_i \in \mathbb{R}^k$ to represent its vector. We use $S_{i:j}$ to represent the matrix concatenated by sequence $[s_i : s_{i+1} : \cdots : s_j]$($[x_1 : x_2]$) denotes the horizontal concatenation of x_1 and x_2). We denote the length of filter by w (Fig. 3(a) shows an example of w = 3), then the weight matrix of the filter is $W \in \mathbb{R}^{w\times k}$. Then the convolution operation between the filter and sentence S results in another vector $c \in \mathbb{R}^{|S|-W+1}$:

$$c_j = W \otimes S_{(j-w+1):j} \tag{1}$$

where $1 \le j \le |S| - w + 1$.

In experiments, we use $n(n > 1)$ filters (or feature maps) to capture different features of an instance. Therefore, we also need n weight matrices $W_c = \{W_1, W_2, \cdots, W_n\}$, so that all the convolution operations can be expressed by

$$c_{ij} = W_i \otimes S_{(j-w+1):j} \tag{2}$$

where $1 \le i \le n$ and $1 \le j \le |S| - w + 1$. Through the convolution layer, we obtain the results vectors $C = \{c_1, c_2, \cdots, c_n\}$.

4.1.5 Piecewise max pooling

In order to capture the structural information and fine-grained features, PCNNs divides an instance into three segments according to the given entity pair (two entities cut the sentence into three parts) and do max-pooling operation on each segment. For the result vector c_i of convolution operations, it can be divided into three parts $c_i = \{c_{i,1}, c_{i,2}, c_{i,3}\}$. Then piecewise max-pooling procedure is $p_{ij} = \max(c_{i,j})$ where $1 \le i \le n$ and $j = 1, 2, 3$.

After that, we can concatenate all the vectors $p_i = [p_{i,1}, p_{i,2}, p_{i,3}](i = 1, 2, ..., n)$ to obtain vector $p \in \mathbb{R}^{3n\times 1}$. Fig. 3(a) displays an example of $n = 3$, in which the gray circles are the positions of entities. Finally, we compute the feature vector $b_S = \tanh(p)$ for sentence S.

4.1.6 Classifier

To compute the score for each relation, the feature vector of each instance is fed into a pair-wise ranking loss based classifier. Given the distributed vector representation of an instance b, the network computes the score for a class label t_i by using the dot product:

$$s_{t_i} = w_{t_i} b \tag{3}$$

where $\mathrm{w}_{t_i} \in R^{1\times 3n}$ is the class embedding for class label t_i. All the class embeddings $\mathrm{w}_{t_i} (i = 1,...,T)$ constitute the class embedding matrix $W_T \in \mathbb{R}^{T\times 3n}$ whose rows encode the distributed vector representations of the different class labels. T is equal to the number of possible relation types for the relation extraction system. Note that the number of dimensions in each class embedding must be equal to the size of the distributed vector representation of the input bag $3n$. The class embedding matrix W_T is a parameter to be learned by the network.

4.1.7 Instance selection

Distant supervised relation extraction suffers from wrong label problem [Riedel, Yao and McCallum (2010)]. The core problem that needs to be solved in the multi-instance learning is to get the corresponding bag feature vector from all the instance feature vectors in the bag. In fact, the problem is the instance selection strategy. We employ an instance selection strategy borrowed from Zeng et al. [Zeng, Liu, Chen et al. (2015)]. Different from [Zeng, Liu, Chen et al. (2015)], we randomly select an instance from the bag with NA label since our model do not give score for NA class (see Section 4.1.8.). In addition, we choose the instance which has the highest score for the bag label except for NA. The scores are computed using Eq. (3). Therefore, our instance selection strategy will not be disturbed by the noise in NA. Assume that there is a bag $B_i = \{b_1^i, b_2^i, ..., b_{q_i}^i\}$ that contains q_i instances with feature vectors $\{b_1^i, b_2^i, ..., b_{q_i}^i\}$ and the bag label is r_i ($r_i \neq NA$). The j-th instance b_j^i is selected and the j is constrained as follows:

$$j = \arg\max\{s_{r_i}^1, s_{r_i}^2, ..., s_{r_i}^{q_i}\} \qquad 1 \le j \le q_i \tag{4}$$

where $s_{r_i}^j = \mathrm{w}_{r_i} b_j^i (1 \le j \le q_i)$ is computed using Eq. (3).

4.1.8 Pair-wise ranking loss

The cross-entropy loss brings the noise of artificial class into the classification process. This phenomenon is mainly due to the noise of artificial class NA. To address this shortcoming, we propose a new pairwise ranking loss instead of cross-entropy which is often used for softmax classifier.

In our model, the network can be stated as a tuple $\theta_D = (E, PF_1, PF_2, W_c, W_T)$. Assume that there are N bags in training set $\{B_1, B_2, ..., B_N\}$, and their labels are relations $\{r_1, r_2, ..., r_N\}$. After the instance selection, we get a representative instance and its corresponding feature vector is considered as the bag feature vector $\bar{b}$. The input for each iteration round is a bag feature vector and the class label. In the pairwise loss, the loss function is defined on the basis of pairs of objects whose labels are different. We can get the loss function by selecting a class label that is different from the input one. In this work, we use the negative samples generated by the generator. For example, when the i-th bag with ground truth label r_i^t is fed into the network, we will get a negative class r_i^k which provided by the generator. The pair-wise ranking loss function is defined as follows:

$$L = \sum_{i=1}^{N} \{\log(1+\exp(\lambda(m^+ - s^i_{r_i^t}))) + \log(1+\exp(\lambda(m^- + s^i_{r_i^k})))\} \quad (5)$$

where $r_i^t \neq r_i^k$ and $t, k \in \{1, 2, ..., T\}$. λ is a scaling factor that magnifies the difference between the scores. m^+ and m^- are the margin for correct and incorrect class, respectively. Since it is very difficult to learn patterns for the artificial class NA, softmax classifier often brings noise into the classification process of the natural classes. By using a ranking classifier, we can avoid explicitly leaning patterns for the artificial class. We omit the artificial class NA by setting the first term in the right side of Eq. (5) to zero and do not learn the class embedding for NA. Thus, our model does not give a score for the artificial class NA and the noise in NA is alleviated. At prediction time, an instance is classified as NA only if all actual classes have negative scores. A bag is positively labeled if and only if the output of the network on at least one of its instances is assigned a positive label and we choose the class which has the highest score.

4.2 Generator network

The aim of our generator is to provide the discriminator with high-quality negative classes to improve DSRE. In this paper, we devise a two layers fully-connected neural network as the generator. As shown in Fig. 2 (a), the input of the generator is a vector v that combines the embeddings of two entities (e_1 and e_2) and the embedding of the ground truth relation r_t. Nonlinear function ReLU is added after the first layer. The output of the network is as follows:

$$O_G = W_G^2 \,\mathrm{Re}\,lu(W_G^1 v) \quad (6)$$

The softmax function is added to the second layer because they can adequately model the "sampling from a probability distribution process" of discrete GANs. The probability distribution of the relation set $\Re$ is defined as:

$$p_G(r_i \mid e_1, e_2, r_t) = \frac{e^{o_G^i}}{\sum e^{o_G^j}}, r_i \in \Re \tag{7}$$

Finally, the generator samples one relation from the distribution $p_G(r_i \mid e_1, e_2, r_t)$ as the output.

4.3 Generative adversarial training

Intuitively, the discriminator should assign a relatively large score for positive class and small score for the negative class. The objective of the discriminator can be formulated as minimizing the following ranking loss function:

$$L = \sum_{i=1}^{N} \{\log(1+\exp(\lambda(m^+ - s^i_{r_i^t}))) + \log(1+\exp(\lambda(m^- + s^i_{r_i^k})))\}$$

$$r_i^k \sim p_G(r_i \mid e_1, e_2, r_t) \tag{8}$$

The only difference between this loss function and Eq. (5) is that it uses negative class from the generator. In order to encourage the generator to generate useful negative classes, the objective of the generator is to maximize the score for negative classes. The objective function can be formulated as maximizing the following expectation of scores for negative classes:

$$L_G = \sum_{i=1}^{N} \mathrm{E}[s^i_{r_i^k}] \qquad r_i^k \sim p_G(r_i \mid e_1, e_2, r_t) \tag{9}$$

Since L_G involves a discrete sampling step, it cannot be directly optimized by gradient-based algorithms. Following Cai et al. [Cai and Wang (2017)], we use a one-step reinforcement learning to solve this problem. (e_1, e_2, r_t) is the state, $p_G(r_i \mid e_1, e_2, r_t)$ is the policy, (e_1, e_2, r_i) is the action, and $s^i_{r_i^k}$ is the reward. We use the policy gradient to optimize the generator. From Eqs. (8) and (9), we could observe that in order to achieve higher reward, the policy used by the generator would punish the trivial negative classes by lowering down their corresponding probability. In the adversarial training, the generator and the discriminator are alternatively trained towards their respective objectives.

5 Experimental results

In this section, we first introduce the dataset and evaluation metrics, then test several variants via cross-validation to determine the parameters used in our experiments, finally show the experimental results and analysis.

5.1 Dataset and evaluation metrics

We evaluate our method on a widely used dataset[4] that was developed by [Riedel, Yao and McCallum (2010)] and has also been used by Hoffmann et al. [Hoffmann, Zhang, Ling et al. (2011); Surdeanu, Tibshirani, Nallapati et al. (2012); Zeng, Liu, Chen et al. (2015)]. This dataset was generated by aligning Freebase relations with the NYT corpus, with sentences from the years 2005-2006 used as the training corpus and sentences from 2007 used as the testing corpus. Following the previous work [Mintz, Bills, Snow et al. (2009)], we evaluate our approach using held-out evaluation. The held-out evaluation compares the extracted relation instances against Freebase relation data.

Table 1: Parameters used in our experiments

Parameters	Value
Window size	3
Feature maps	230
Word dimension	50
Position dimension	5
Batch size	50
Adadelta parameter	ρ =0.95， ε =1e^(-6)
Constant term	2

5.2 Experimental settings

In this work, we use the Skip-gram model (word2vec)[5] to train the word embeddings on the NYT corpus. The tokens are concatenated using the ## operator when the entity has multiple word tokens. Position features are randomly initialized with a uniform distribution between [-1, 1]. For the convenience of comparing with baseline methods, the PCNNs module uses the same parameter settings as Zeng et al. [Zeng, Liu, Chen et al. (2015)]. We use L2 regularization with regularization parameter β=0.001. We tune the proposed model using three-fold validation to study the effects of two parameters: the constant terms λ used in the loss function. We use a grid search to determine the optimal parameters and manually specify subsets of the parameter spaces: $\lambda \in \{1, 2, \ldots, 10\}$. Tab. 1 shows all parameters used in the experiments.

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

[5] https://code.google.com/p/word2vec/

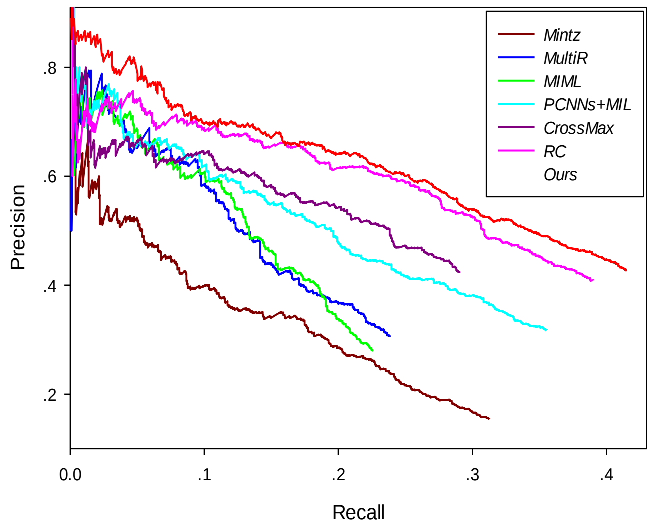


Figure 4: Performance comparison of proposed method and baseline methods

5.3 Baselines

We select some previous works that use handcrafted features as well as the neural network based methods as baselines. *Mintz* is proposed by Mintz et al. [Mintz, Bills, Snow et al. (2009)] which extracts features from all instances; *MultiR* is a multi-instance learning method proposed by Hoffmann et al. [Hoffmann, Zhang, Ling et al. (2011)]; MIML is a multi-instance multi-labels method proposed by Surdeanu et al. [Surdeanu, Tibshirani, Nallapati et al. (2012)]; *PCNNs+MIL* is the method proposed by Zeng et al. [Zeng, Liu, Chen et al. (2015)], which incorporates multi-instance learning with PCNNs to extract bag features; *CrossMax* is proposed by Jiang et al. [Jiang, Wang, Li et al. (2016)], which exploits PCNNs and cross-sentence max-pooling to select features across different instances; *Ranking Loss+Cost-Sensitive (RC)* in the Zeng et al. [Zeng, Zeng and Dai (2017)], is used to address the problem of *NA* noise and class imbalance.

5.4 Comparison with baseline methods

In this section, we show the experimental results and comparisons with baseline methods.

In the following experiments, we use *Ours* to refer to the proposed model that use the GAN-based framework.

The held-out evaluation provides an approximate measure of precision without requiring costly human evaluation. Half of the Freebase relations are used for testing. The relation instances discovered from the test articles are automatically compared with those in Freebase. For the convenience of comparing with baseline methods, the prediction results are sorted by the classification scores and a precision-recall curve is created for the positive classes.

Table 2: Precision, recall and F1 score of some relations

Relations	Ours w/o GAN			Ours		
	P	R	F1	P	R	F1
/location/location/contains	35.45	**43.44**	39.04	**37.62**	43.25	40.24
/people/person/place_lived	12.87	18.67	15.24	**16.49**	**19.53**	**17.88**
/people/person/nationality	**47.79**	24.53	32.42	47.62	**27.22**	**34.64**
/business/person/company	35.41	**52.48**	42.29	**38.80**	52.31	**44.55**
/people/person/place_of_birth	11.45	16.82	13.62	**16.86**	**18.95**	**17.84**
/people/deceased_person/place_of_death	26.31	22.22	24.09	**27.52**	**26.03**	**26.75**
/location/neighborhood/neighborhood_of	37.50	29.34	32.92	**39.06**	**32.70**	**35.60**
/business/company/founders	47.05	28.76	35.70	**48.30**	**29.52**	**36.64**

Fig. 4 shows the precision-recall curves of our approach and all the baselines. We can observe that our model outperforms all the baseline systems and improves the results significantly. It is worth emphasizing that the best of all baseline methods achieve a recall level of 38%. In contrast, our model is much better than the previous approach and enhances the recall to approximately 41%. The significant improvement can be contributed to the magic of our new pair-wise ranking loss function. The classifier uses pair-wise ranking loss which avoids explicitly learning the patterns for *NA*. Thus, our model will not trend to classify the samples as *NA* compared to softmax classifier and recalls more positive samples.

Furthermore, our model achieves a large improvement in precision especially at higher recall levels. From Fig. 4, we can see that our model achieves a precision of 43% when the recall is 41%. In contrast, when *PCNNS*, *CrossMax* and *RC* achieve such precision, the recalls are decreased to approximately 24%, 29% and 37%, respectively. Thus, our approach is advantageous from the point of view of precision. Also note that our model also shows advantages in the precision when the recall is very low compare with *RC*. This phenomenon is mainly due to the fact that when using adversarial training, the generator can consistently provide high-quality negative classes. Therefore, we can conclude that our model outperforms all the baseline systems and improves the results significantly in terms of both precision and recall.

5.4.1 Effects of adversarial training

In order to validate the effects of adversarial training, we compute the confusion matrix and analyze the detail of some relations in Tab. 2. From Tab. 2, we can see that: (1) It achieves better results in the majority of relations when using the adversarial training fra-mework; (2) The F1 score have a significant improvement in */people/person/place_lived* an-d */people/person/place_of_birth* compared with Zeng et al. [Zeng, Zeng and Dai (2017)], mainly due to the fact that the generator supply high-qulity negative classes. Nonetheless, the GA-N-based framework helps to improve the performance in this case.

The precision-recall curves with and without GAN are illustrated in Fig. 5, from which we can also observe that it brings better performance when using the GAN in DSRE. After removing the GAN, our model degrades to *PCNNS+MIL+Traditional Ranking loss*.

In order to further validate the effects of our model, the PCNNS+MIL result is illustrated in Fig. 5. As we expected, our method can get better performance compared with *PCNNS+MIL*. The superiority of our approach indicates that incorporate the GAN framework in DSRE can effectively improve DSRE.

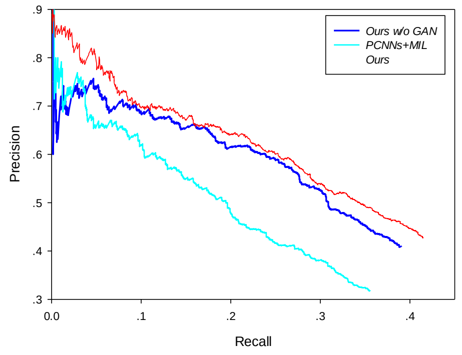


Figure 5: Effects of generative adversarial training

6 Conclusions

In this paper, we exploit a novel GAN-based framework for DSRE. In the traditional ranking based classifier, the majority of the generated negative class can be easily discriminated from positive class. To address the shortcoming, we design a neural network as the negative class generator to supply high-quality negative classes. The generator is learning to produce more and more discriminable negative classes, while the discriminator has to become better as well. We perform experiments on a widely used dataset and verify the adversarial learning approach. Experiment results show that our approach obtains state-of-the-art performance on the dataset.

Acknowledgments: This research work is supported by the National Natural Science Foundation of China (NO. 61772454, 6171101570, 61602059). Hunan Provincial Natural Science Foundation of China (No. 2017JJ3334), the Research Foundation of Education Bureau of Hunan Province, China (No. 16C0045), and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR). Professor Jin Wang is the corresponding author.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. (2003): A neural probabilistic language model. *Journal of Machine Learning Research*, vol. 3, pp. 1137-1155.

Bunescu, R. C.; Mooney, R. J. (2005): Subsequence kernels for relation extraction. *Advances in Neural Information Processing Systems*, pp. 171-178.

Cai, L.; Wang, W. (2017): Kbgan: Adversarial learning for knowledge graph embeddings. *Computation and Language*.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, Michael; Kavukcuoglu, K. et al. (2011): Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, vol. 12, pp. 2493-2537.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D. et al. (2014): Generative adversarial nets. *Advances in neural information processing systems*, pp. 2672-2680.

Gurusamy, R.; Subramaniam, V. (2017): A machine learning approach for mri brain tumor classification. *Computers, Materials & Continua*, vol. 53, no. 2, pp. 91-108.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; Weld, D. S. (2011): Knowledge-based weak supervision for information extraction of overlapping relations. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 541-550.

Huang, F.; Ahuja, A.; Downey, D.; Yang, Y.; Guo, Y. et al. (2014): Learning representations for weakly supervised natural language processing tasks. *Computational Linguistics*, vol. 40, no. 1, pp. 85-120.

Jia, R.; Liang, P. (2017): Adversarial examples for evaluating reading comprehension systems. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021-2031.

Jiang, X.; Wang, Q.; Li, P.; Wang, B. (2016): Relation extraction with multi-instance multilabel convolutional neural networks. *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, pp. 1471-1480.

Kambhatla, N. (2004): Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, pp. 22.

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; Sun, M. (2016): Neural relation extraction with selective attention over instances. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 2124-2133.

Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. (2013): Efficient estimation of word representations in vector space. *Computation and Language*.

Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. (2009): Distant supervision for relation extraction without labeled data. *ACL'09 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*, vol. 2, pp. 1003-1011.

Mirza, M.; Osindero, S. (2014): Conditional generative adversarial nets. *Learning*, pp. 1-7.

Nguyen, T. V. T.; Moschitti, A. (2011): End-to-end relation extraction using distant supervision from external semantic repositories. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 277-282.

Pennington, J.; Socher, R.; Manning, C. D. (2014): Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746-1751.

Riedel, S.; Yao, L.; McCallum, A. (2010): Modeling relations and their mentions without labeled text. *Proceedings of ECML PKDD*, pp. 148-163.

Santos, C. N. D.; Xiang, B.; Zhou, B. (2015): Classifying relations by ranking with convolutional neural networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, vol. 1, pp. 626-634.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; Manning, C. D. (2012): Multi-instance multi-label learning for relation extraction. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 455-465.

Williams, R. J. (1992): Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement Learning*, pp. 5-32.

Yu, L.; Zhang, W.; Wang, J. Yu, Y. (2017): Seqgan: Sequence generative adversarial nets with policy gradient. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 2852-2858.

Yuan, C.; Li, X.; Wu, Q. J.; Li, J.; Sun, X. (2017): Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis. *Computers, Materials & Continua*, vol. 53, no. 4, pp. 357-372.

Zelenko, D.; Aone, C.; Richardella, A. (2003): Kernel methods for relation extraction. *Journal of Machine Learning Research*, vol. 3, pp. 1083-1106.

Zeng, D.; Liu, K.; Chen, Y.; Zhao, J. (2015): Distant supervision for relation extraction via piecewise convolutional neural networks. *Conference on Empirical Methods in Natural Language Processing*, pp. 1753-1762.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. (2014): Relation classification via convolutional deep neural network. *Proceedings of COLING 2014, 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2335-2344.

Zeng, D.; Zeng, J.; Dai, Y. (2017): Using cost-sensitive ranking loss to improve distant supervised relation extraction. *Chinese Computational Linguistics and Natural Language Processing Based on Naturally*, pp. 184-196.

Zhou, G.; Su, J.; Zhang, J.; Zhang, M. (2005): Exploring various knowledge in relation extraction. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 427-434.