Knowing False Negatives: An Adversarial Training Method for Distantly Supervised Relation Extraction

Kailong Hao[†]   **Botao Yu**[†]   **Wei Hu**[†, ‡, *]

[†] State Key Laboratory for Novel Software Technology, Nanjing University, China
[‡] National Institute of Healthcare Data Science, Nanjing University, China
`klhao.nju@gmail.com, btyu.nju@gmail.com, whu@nju.edu.cn`

Abstract

Distantly supervised relation extraction (RE) automatically aligns unstructured text with relation instances in a knowledge base (KB). Due to the incompleteness of current KBs, sentences implying certain relations may be annotated as N/A instances, which causes the so-called false negative (FN) problem. Current RE methods usually overlook this problem, inducing improper biases in both training and testing procedures. To address this issue, we propose a two-stage approach. First, it finds out possible FN samples by heuristically leveraging the memory mechanism of deep neural networks. Then, it aligns those unlabeled data with the training data into a unified feature space by adversarial training to assign pseudo labels and further utilize the information contained in them. Experiments on two wildly-used benchmark datasets demonstrate the effectiveness of our approach.

1 Introduction

Relation extraction (RE), defined as the task of identifying relations from unstructured text given two entity mentions, is a key component in many NLP applications such as knowledge base (KB) population (Ji and Grishman, 2011; De Sa et al., 2016) and question answering (Yu et al., 2017). Supervised RE requires a large amount of human labeled data, which is often labor-intensive and time-consuming. Distant supervision (DS), proposed by (Mintz et al., 2009), deals with this problem by aligning a large corpus with a KB such as Freebase to provide weak supervision. It relies on the assumption that *if two entities participate in a relation in a KB, then any sentence containing this entity pair expresses this relation.*

However, this assumption is too strong and brings in noises, particularly when the training KB is an external information source and not primarily derived from the training corpus. To alleviate

---

Sentences	Relations
[Romano Mussolini] , the Italian jazz pianist , ... , died on Tuesday in [Rome] .	death_place
... in the [Uptown] neighborhood of [New Orleans] , ... , where it was founded .	neighborhood

Table 1: False negative examples excerpted from the NYT10 (Riedel et al., 2010) N/A set.

this problem, Riedel et al. (2010) relax this assumption to *if two entities participate in a relation, **at-least-one** sentence that mentions these two entities expresses this relation.* Bunescu and Mooney (2007) propose multi-instance learning by organizing sentences with the same entity pair into one set, referred to as a *bag*. Following this setting, Riedel et al. (2010); Hoffmann et al. (2011); Surdeanu et al. (2012) propose diverse hand-crafted features. Zeng et al. (2014, 2015) leverage convolutional neural networks (CNN) to learn the representations of instances. Lin et al. (2016) apply an attention mechanism to select informative sentences from a bag. Recently, graph convolutional networks (GCN) have been effectively employed for capturing the syntactic information from dependency trees (Vashishth et al., 2018).

Most of the above works focus on the false positive (FP) problem (which is caused by the strong DS assumption), but totally neglect the false negative (FN) problem, which is also important and induces improper biases in both training and testing procedures. DS treats a sentence as a negative sample (marked as N/A) whose entity pair does not have a known relation in the KB. However, due to the incompleteness of current KBs, sentences implying predefined relations are likely to be mislabeled as N/A. For example, over 70% of people in Freebase do not have birth places (Dong et al., 2014). As shown in Table 1, we have annotated the ground-truth relations for two FN sentences. In fact, the missing facts in KBs yield plenty of FN

---

* Corresponding author

sentences in the automatically annotated datasets.

Generative adversarial networks (GAN) (Goodfellow et al., 2014) is first introduced into RE by (Qin et al., 2018) to learn a sentence-level FP filter. However, they only focus on the FP problem and their model cannot be generalized to other scenarios. Li et al. (2019) attempt to solve the FN problem using entity descriptions from Wikipedia to filter FN samples from the N/A set and further utilize GAN in a semi-supervised manner. However, their method heavily depends on external resources and is not applicable to all situations. Moreover, their filtering heuristic "*two entities mention each other on their Wikipedia pages may imply the predefined relations*" is even stronger than that of (Riedel et al., 2010). It may filter some noisy data but at the same time introducing even more noises. For example, in the sentence "*... to the poetry from [India], China and [Korea] that stretches the book ...*" excerpted from NYT10, India and Korea mention each other on Wikipedia, but this sentence implies no relation.

In this paper, we propose a novel two-stage approach for distantly supervised RE. In the first stage, called **mining**, we find out possible FN samples from the N/A set by heuristically leveraging the memory mechanism of deep neural networks. According to (Arpit et al., 2017), while deep neural networks are capable of memorizing noisy data, they tend to preferentially learn simple patterns first, that is to say, deep neural networks tend to learn and memorize patterns from clean instances within a noisy dataset. We design a Transformer-based (Vaswani et al., 2017) deep filter to mine FN samples from the N/A set. In the second stage, called **aligning**, we formulate the problem into a domain adaptation (DA) paradigm. We exploit a gradient reversal layer (GRL) to align the mined unlabeled data from stage one with the training data into a unified feature space. After aligning, each sentence is assigned with a pseudo label and a confidence score, which provide extra information and attenuate incorrect biases in both training and testing procedures.

In summary, our main contributions are fourfold:

- We propose a simple yet effective method to filter noises in a DS dataset by leveraging the memory mechanism of deep neural networks, without any external resources.
- We formulate distantly supervised RE as a DA paradigm and utilize adversarial training to align unlabeled data with training data into

a unified space, and generate pseudo labels to provide additional supervision.
- We achieve new state-of-the-art on two popular benchmark datasets NYT10 (Riedel et al., 2010) and GIDS (Jat et al., 2018).
- By mining the test set, we show that the FN problem greatly misleads the evaluation of DS models and deserves further study.

2 Related Work

Distant supervision. Supervised RE requires a large amount of human-labeled training data, which is labor-intensive and time-consuming. To address this limitation, Mintz et al. (2009) propose DS by heuristically aligning a text corpus to a KB. Riedel et al. (2010) relax DS for multi-instance single-label learning. Subsequently, to handle the overlapping relations between entity pairs, Hoffmann et al. (2011); Surdeanu et al. (2012) propose the multi-instance multi-label learning paradigm. Currently, DS is already a common practice in RE.

Neural relation extraction. The above works strongly rely on the quality of hand-engineered features. Zeng et al. (2014) first propose an end-to-end CNN-based neural network to automatically capture relevant lexical-level and sentence-level features. Zeng et al. (2015); Lin et al. (2016) further improve this through piecewise max pooling and selective attention (Bahdanau et al., 2015). Zhou et al. (2016) propose an attention-based LSTM to capture the most important semantic information in sentences. External knowledge like entity descriptions and type information has been used for RE (Yaghoobzadeh et al., 2017; Vashishth et al., 2018). Pre-trained language models contain a notable amount of semantic information and common-sense knowledge, and several works have applied them to RE (Alt et al., 2019; Xiao et al., 2020).

Adversarial training. Adversarial training is a machine learning technique that improves the networks using an adversarial objective function or deceptive samples. Wu et al. (2017) bring it in RE by adding adversarial noises to the training data. Qin et al. (2018) propose DSGAN to learn a generator that explicitly filters FP instances from the training dataset. Li et al. (2019) propose a semi-distant supervision method by first splitting a dataset through entity descriptions and then using GAN to make full use of unsupervised data. Luo et al. (2020) learn the distribution of true positive

instances through adversarial training and select valid instances via a rank-based model.

Learning with noisy labels. As noisy labels degrade the generalization performance of deep neural networks, learning from noisy labels (a.k.a. robust training) has become an important task in modern deep learning (Song et al., 2021). Arpit et al. (2017) find out that, although deep networks are capable of memorizing noisy data, they tend to learn simple patterns first. Co-teaching (Han et al., 2018) trains two deep neural networks simultaneously and lets them teach each other given every mini-batch. Yan et al. (2016); Nguyen et al. (2020); Li et al. (2020) transform the problem into a semi-supervised learning task by treating possibly false-labeled samples as unlabeled.

3 Methodology

In this section, we introduce our two-stage framework called FAN (False negative Adversarial Networks) in detail. First, we describe how we discover possibly wrong-labeled sentences from the N/A set. Then, we introduce our adversarial DA method, which assigns pseudo labels to unlabeled data with confidence scores.

3.1 Stage I: Mining

We define a distantly supervised dataset $\mathcal{D} = \{s_1, s_2, \ldots, s_N\}$, where each sample s_i is a quadruple, consisting of an input sequence of tokens $t_i = [t_i^1, \ldots, t_i^n]$, $head_i$ and $tail_i$ for head and tail entity positions in sequence t_i, respectively, and the corresponding relation r_i assigned by DS. We split $\mathcal{D}$ into two parts: sentences with predefined relations are divided into the positive set, denoted by $\mathcal{P}$; sentences implying no relations are divided into the negative set, denoted by $\mathcal{N}$, $\mathcal{D} = \mathcal{P} \cup \mathcal{N}$. In this work, we focus on the noises in $\mathcal{N}$, where sentences may be mislabeled by incomplete KBs and useful information is not yet fully discovered.

According to (Arpit et al., 2017), deep neural networks are prone to learn clean samples first, and then gradually learn noisy samples. Following (Malach and Shalev-Shwartz, 2017; Jiang et al., 2018; Han et al., 2018; Nguyen et al., 2020), after proper training, we filter samples in $\mathcal{N}$ with logits larger than threshold θ as possible FN samples. All FN samples form a set $\mathcal{M}$ and remaining samples form a set $\mathcal{N}'$, where $\mathcal{N} = \mathcal{N}' \cup \mathcal{M}$. The original training dataset $\mathcal{D}$ is refined to $\mathcal{D}'$, where $\mathcal{D}' = \mathcal{P} \cup \mathcal{N}'$. The deep noise filter can capture meaningful
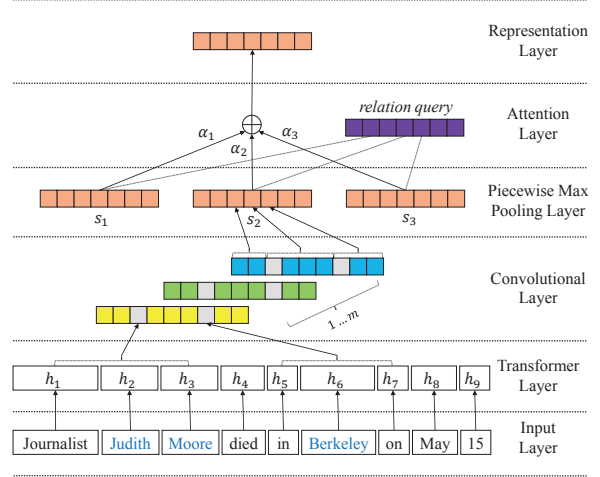

Figure 1: Architecture of encoder.

semantic patterns to differentiate between $\mathcal{P}$ and $\mathcal{N}$. For the samples in $\mathcal{N}$ with logits larger than θ, they may imply predefined relations but only the DS annotations are inaccurate. We argue that $\mathcal{M}$ can be considered as unlabeled data in a semi-supervised way to provide supplementary information.

3.1.1 Sentence Encoder

As for the mining step, we take a mapping function $f(\cdot)$ to map r_i to a binary label. If r_i is N/A, $f(r_i) = 0$; otherwise $f(r_i) = 1$. We use pretrained language model BERT (Devlin et al., 2019) as our embedding module, as it contains a large amount of semantic information and commonsense knowledge. Token sequence $t_i = [t_i^1, \ldots, t_i^n]$ is fed into the pre-trained model and the last hidden representation $\mathbf{h}_i = [\mathbf{h}_i^1, \ldots, \mathbf{h}_i^n]$ is used as our token embedding.

CNN is a widely used architecture for capturing local contextual information (Zeng et al., 2014). The convolution operation involves taking the dot product of the convolutional filter $\mathbf{W}$ with each k-$gram$ in the sequence $\mathbf{h}_i$ to obtain a new representation $\mathbf{p}_i$:

$$p_i^j = \mathbf{W} \cdot [\mathbf{h}_i^{j-k+1} : \ldots : \mathbf{h}_i^j], \qquad (1)$$

where p_i^j is the j-th dimension of $\mathbf{p}_i$. $\mathbf{W} \in \mathbb{R}^{k \times d}$ is the convolutional filter, where k is the kernel size and d is the hidden dimension. $[\mathbf{h}_i^{j-k+1} : \ldots : \mathbf{h}_i^j]$ refers to the concatenation from $\mathbf{h}_i^{j-k+1}$ to $\mathbf{h}_i^j$. In order to capture diverse features and extract the local information at different levels, we make use of multiple filters and varied kernel sizes.

In RE, representation $\mathbf{p}_i$ can be partitioned into three parts according to $head_i$ and $tail_i$, i.e., $\mathbf{p}_i =$

$\{\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \mathbf{p}_{i_3}\}$. To capture the structural information between two entities and obtain fine-grained features, we take piecewise max pooling as (Zeng et al., 2015). For each convolutional filter, we can obtain a three-dimensional vector $\mathbf{q}_i$:

$$\mathbf{q}_i = [\max(\mathbf{p}_{i_1}), \max(\mathbf{p}_{i_2}), \max(\mathbf{p}_{i_3})]. \quad (2)$$

We concatenate vectors from multiple convolutional filters and get the sentence representation $\mathbf{v}_i \in \mathbb{R}^{3m}$, where m is the number of convolutional filters with varied kernel sizes. The architecture of the sentence encoder is shown in Figure 1.

3.1.2 Noise Filter

Through sentence encoding, each sample $s_i = (t_i, head_i, tail_i, r_i)$ is transformed into a fixed-dimensional vector $\mathbf{v}_i \in \mathbb{R}^{3m}$. We feed it into a multilayer perceptron to obtain the probability:

$$o_i = \sigma(\mathbf{W}_f \mathbf{v}_i + b_f), \quad (3)$$

where $\mathbf{W}_f \in \mathbb{R}^{1 \times 3m}$ is the transformation matrix and b_f is the bias. $\sigma(\cdot)$ denotes the sigmoid activation. Output o_i implies the probability of current sentence belonging to $\mathcal{P}$.

We use the binary cross-entropy as our loss to train the deep noise filter in an end-to-end way:

$$L_f = \mathbb{E}_{s_i \sim \mathcal{D}}\big[- f(r_i) \log(\mathrm{P}(s_i)) \\ - (1 - f(r_i)) \log(1 - \mathrm{P}(s_i)) \big]. \quad (4)$$

3.2 Stage II: Aligning

Instances in $\mathcal{M}$ are mislabeled due to the incompleteness of KBs and the distribution may be different from the original training dataset (Ye et al., 2019). A naive way is just dropping those data and using adjusted dataset $\mathcal{D}'$ for training. Doing like this would lose useful information contained in $\mathcal{M}$ and thus is not optimal. Unlabeled data can be annotated by humans, but it is time-consuming and is not applicable to large datasets.

In fact, these unlabeled samples imply predefined relations and can be used together with $\mathcal{D}'$ in a semi-supervised learning paradigm. We formulate this problem as a DA task and the objective is aligning the distributions of $\mathcal{M}$ and $\mathcal{P}$ into a unified feature space. To achieve this objective, we propose a method inspired by GAN. The generator tries to fool the discriminator so that it cannot distinguish the samples in $\mathcal{M}$ and $\mathcal{P}$. On the contrary, the discriminator tries its best to differentiate them. The training procedure forms a classic min-max game by adversarial objective functions. The overall architecture is shown in Figure 2.

3.2.1 Bag Encoder

The sentence encoding layer reuses the architecture in Section 3.1.1. Due to the noisy DS annotations, multi-instance learning is introduced and relation classification is applied on the bag level. One bag $\mathcal{B} = \{s_1, \ldots, s_t\}$ contains t sentences for the same entity pair. Bag representation $\mathbf{g}_i$ is derived from a weighted sum over the individual sentence representations:

$$\mathbf{g}_i = \sum_j \alpha_j \mathbf{v}_j, \quad (5)$$

where α_j is the weight assigned to the corresponding sentence computed through selective attention (Lin et al., 2016). The weight is obtained by computing the similarity between the learned relation query representation $\mathbf{r}_i \in \mathbb{R}^{3m}$ and each sentence:

$$\alpha_i = \frac{\exp(\mathbf{v}_i \cdot \mathbf{r}_i)}{\sum_{j=1}^{t} \exp(\mathbf{v}_j \cdot \mathbf{r}_i)}. \quad (6)$$

3.2.2 Relation Classifier

For a bag in $\mathcal{P}$, its DS label is known. To compute the probability distribution over relations, a linear layer followed by a softmax layer is applied to a bag representation $\mathbf{g}_i \in \mathbb{R}^{3m}$:

$$\mathrm{P}(r_i \,|\, \mathbf{g}_i) = \mathrm{softmax}(\mathbf{W}_c \mathbf{g}_i + \mathbf{b}_c), \quad (7)$$

where $\mathbf{W}_c \in \mathbb{R}^{l \times 3m}$, and l is equal to the number of predefined relations.

During training, we aim to optimize the following cross-entropy loss:

$$L_{cls} = \mathbb{E}_{\mathbf{g}_i \sim \mathcal{P}}\big[- \log \mathrm{P}(r_i \,|\, \mathbf{g}_i) \big]. \quad (8)$$

3.2.3 Gradient Reversal Layer

Pre-trained language models have shown great power in many NLP tasks. They are huge in size and have tremendous ability to fit distributions. Following (Ganin et al., 2016; Chen et al., 2018), we use a GRL after the bag encoder. When forward passing, it works as an identity function, while when back propagating, it reverses gradients to their opposite:

$$\mathrm{GRL} = \mathbb{I}(\cdot)\nabla_{\Theta}, \quad (9)$$

where Θ denotes parameters of the bag encoder. When forward passing, $\mathbb{I}(\cdot) = 1$, and when back propagating, $\mathbb{I}(\cdot) = -1$.
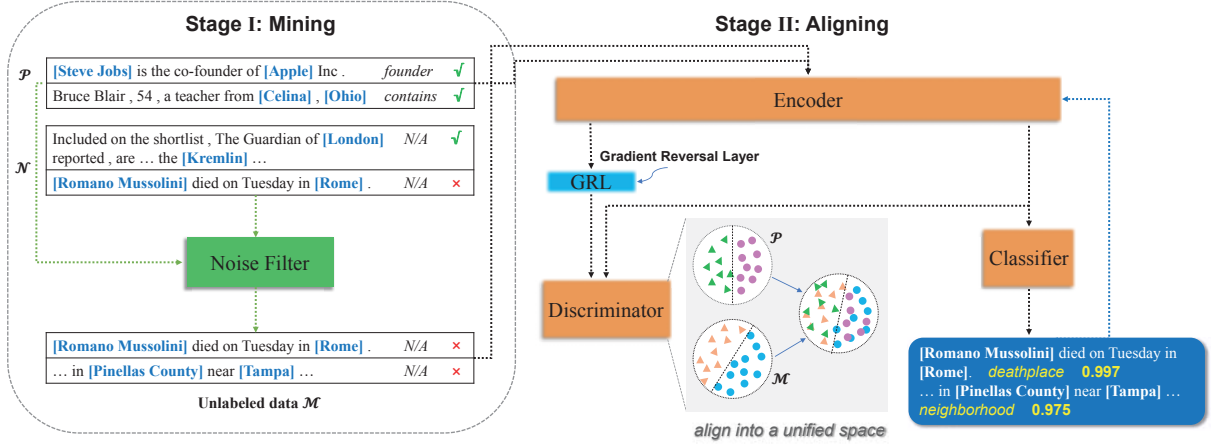
Figure 2: Overview of FAN. Informative unlabeled samples are first mined through the noise filter and then fed into the adversarial module to generate pseudo labels with confidences. The adjusted training data are used for final training.

3.2.4 Discriminator

Given a bag representation $\mathbf{g}_i \in \mathbb{R}^{3m}$, the discriminator first implements an affine transformation and then uses sigmoid function $\sigma(\cdot)$ to obtain the probability distribution:

$$o'_i = \sigma(\mathbf{W}_d \mathbf{g}_i + b_d). \quad (10)$$

3.3 Training Objective

We use adversarial training to generate a unified data distribution. To enforce instances of the same class closer and push away instances from different classes, a contrastive loss is designed for better feature representation.

3.3.1 Adversarial Loss

The bag encoder is optimized to give separate representations for instances in $\mathcal{P}$, so that samples from different classes can be easily distinguished by the relation classifier. In the meantime, it forces the distribution of $\mathcal{M}$ to fit into the distribution of $\mathcal{P}$. The encoder here plays two roles: representation learner and distribution adapter. The classification learning objective is Eq. (8), and the generator objective is

$$L_g = -\mathbb{E}_{s \sim \mathcal{M}} \big[D(G(s)) \big]. \quad (11)$$

On the contrary, the discriminator attempts to distinguish samples from $\mathcal{M}$ with $\mathcal{P}$. Straightforwardly, for generator, the labels of samples in $\mathcal{M}$ are 1; but for discriminator, the labels of samples in $\mathcal{M}$ are 0. The labels of instances in $\mathcal{P}$ are always 1. The discriminator objective is

$$L_d = -\mathbb{E}_{s_i \sim \mathcal{P}} \big[D(s_i) \big] + \mathbb{E}_{s_j \sim \mathcal{M}} \big[D(G(s_j)) \big]. \quad (12)$$

Generator and discriminator improve each other in iterations.

3.3.2 Contrastive Loss

Bag representations are expected to be able to cluster instances with the same relation. We aim to increase the distances between samples with different relations and reduce the variance of distances with the same relation.

Simply put, given two instances, their similarity score should be high if they belong to the same relation and low otherwise. We use the contrastive loss (Neculoiu et al., 2016) for this objective. Following (Zhou et al., 2020), given a bag representation $\mathbf{g}$, we divide all the other instances with the same relation type as $\mathcal{Q}_+$ and the ones with different types as $\mathcal{Q}_-$. The contrastive loss can be formulated as

$$L_{ctra} = \mathbb{E}_{\mathbf{g} \sim \mathcal{P}} \Big[\max_{\mathbf{g}_i \in \mathcal{Q}_+} \text{dist}_+(\mathbf{g}, \mathbf{g}_i) - \min_{\mathbf{g}_j \in \mathcal{Q}_-} \text{dist}_-(\mathbf{g}, \mathbf{g}_j) \Big], \quad (13)$$

where the measurement of distance is defined as follows:

$$\text{dist}_+(\mathbf{g}, \mathbf{g}_i) = \big(\max(\tau - \cos(\mathbf{g}, \mathbf{g}_i), 0) \big)^2,$$
$$\text{dist}_-(\mathbf{g}, \mathbf{g}_j) = 1 - \big(\max(\cos(\mathbf{g}, \mathbf{g}_j), 0) \big)^2, \quad (14)$$

where τ is a hyperparameter for avoiding collapse of the representations of bags, $\cos(\cdot)$ denotes the cosine function.

3.3.3 Overall Loss

The adversarial training procedure is modeled as multi-task learning and trained in an end-to-end

way. The overall objective function is

$$L = L_{cls} + \alpha \cdot L_g + \beta \cdot L_d + \gamma \cdot L_{ctra}, \quad (15)$$

where α, β, γ are hyperparameters.

4 Experiment Setup

We conduct the experiments on two widely used benchmark datasets: NYT10 and GIDS (Google-IIsc Distant Supervision dataset). The source code is publicly available[1].

4.1 Datasets

The statistics of two datasets are listed in Table 2. We briefly describe them below:

- **NYT10** is developed by (Riedel et al., 2010) through aligning Freebase with the New York Times corpus. News from year 2005 to 2006 are used for creating the training set and from year 2007 for the test set. The entity mentions are annotated using Stanford NER (Finkel et al., 2005) and linked to Freebase. The dataset has been broadly used for RE (Hoffmann et al., 2011; Surdeanu et al., 2012; Vashishth et al., 2018; Alt et al., 2019).
- **GIDS** is built by extending the Google RE corpus[2] with additional instances for each entity pair (Jat et al., 2018). It assures that the *at-least-one* assumption of multi-instance learning holds, which makes the automatic evaluation more accurate and reliable.

4.2 Comparative Models

To evaluate the proposed FAN, we compare it with the following seven representative models:

- **PCNN-ONE** (Zeng et al., 2015): A CNN-based neural RE model using piecewise max pooling for better sentence representations.
- **PCNN-ATT** (Lin et al., 2016): A model using selective attention to choose useful information across different sentences in a bag.
- **BGWA** (Jat et al., 2018): An attention-based neural model which formulates various word attention and entity attention mechanisms to help a RE model focus on the right context.
- **RESIDE** (Vashishth et al., 2018): A GCN-based model which uses side information from a KB to improve performance.

Datasets	Splits	Sentences	Entity pairs
NYT10 (Relations: 53)	Train	455,771	233,064
	Dev	114,317	58,635
	Test	172,448	96,678
GIDS (Relations: 5)	Train	11,297	6,498
	Dev	1,864	1,082
	Test	5,663	3,247

Table 2: Statistics of NYT10 and GIDS.

- **DISTRE** (Alt et al., 2019): A GPT-based model capturing semantic and syntactic information along with commonsense knowledge.
- **DS-GAN** (Li et al., 2019): A GAN-based model which treats distantly supervised RE as a semi-supervised learning process.
- **DS-VAE** (Christopoulou et al., 2021): A variational auto-encoder (VAE) based model which biases the latent space of sentences and is trained jointly with a relation classifier.

4.3 Criteria

Following the conventions (Zeng et al., 2015; Lin et al., 2016; Vashishth et al., 2018; Alt et al., 2019), we use the held-out evaluation. For each model, we compute the precision-recall (PR) curves and report the area under curve (AUC) scores for straightforward comparison. We also report P@N, which measures the percentage of correct classifications in the top-N most confident predictions. Additionally, micro-F1 is measured at different points along the PR-curve and the best is reported.

For NYT10, we compare with all models listed above. We reuse the results reported in the original papers for BGWA, RESIDE, DISTRE and DS-VAE, and implement PCNN-ONE, PCNN-ATT, DS-GAN by ourselves. Because the source code of DS-VAE is not released yet, we cannot obtain its PR-curve. For GIDS, we only compare with PCNN-ONE, PCNN-ATT, BGWA and RESIDE, as other models are not applicable to this dataset.

5 Results

5.1 Overall Results

The PR-curves on NYT10 and GIDS are shown in Figure 3. On both datasets, FAN achieves the best results. On NYT10, we get visibly higher recall, especially when precision is higher than 75.0, which indicates that our model can find more informative samples along with correct labels. It makes sense because FAN knows more information by digging
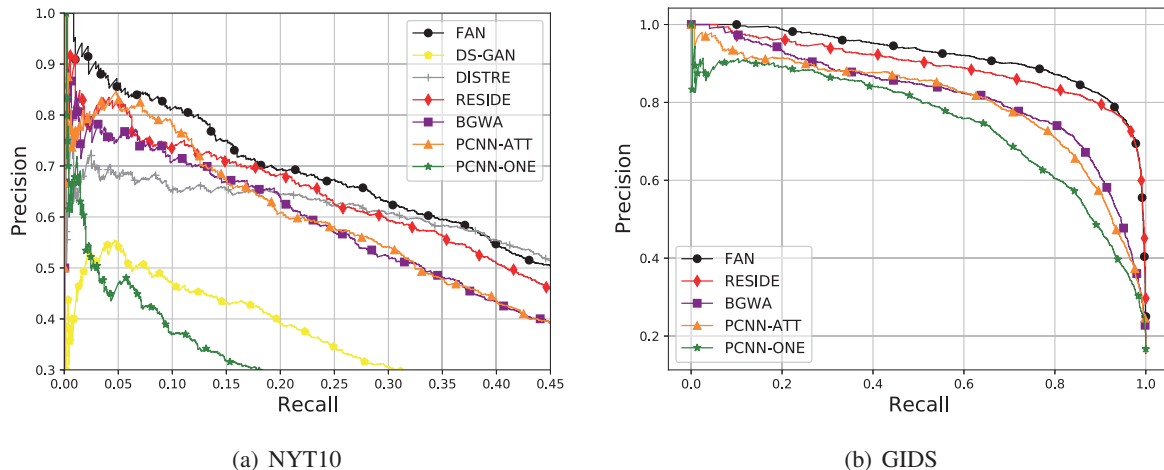
---

[1]https://github.com/nju-websoft/FAN
[2]https://research.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html

(a) NYT10

(b) GIDS

Figure 3: PR-curve of different models. FAN achieves the state-of-the-art on both datasets.

	NYT10	GIDS
PCNN-ONE	15.7	74.4
PCNN-ATT	37.6	79.9
BGWA	34.0	81.5
RESIDE	41.5	89.1
DISTRE	42.2	-
DS-GAN	20.3	-
DS-VAE	42.9	-
FAN	**45.5**	**90.3**

Table 3: AUC scores on NYT10 and GIDS.

	P@100	P@200	P@300	Micro-F1
PCNN-ONE	49.5	44.7	44.8	24.3
PCNN-ATT	82.8	79.9	75.0	42.0
BGWA	82.0	75.0	72.0	42.1
RESIDE	81.8	75.4	74.3	45.7
DISTRE	68.0	67.0	65.3	48.6
DS-GAN	60.6	57.3	51.8	30.7
DS-VAE	83.0	75.5	73.0	-
FAN	**85.8**	**83.4**	**79.9**	**48.7**

Table 4: P@N and micro-F1 on NYT10.

	P@100	P@200	P@300	Micro-F1
PCNN-ONE	88.1	90.5	90.0	70.0
PCNN-ATT	97.0	93.5	91.4	75.6
BGWA	99.0	**98.0**	96.0	77.3
RESIDE	**100.0**	97.5	**97.0**	84.6
FAN	98.3	97.6	96.8	**85.9**

Table 5: P@N and micro-F1 on GIDS.

informative samples from the N/A set and weakens improper biases in the training procedure. On NYT10, our AUC is **45.5**, improving 3.3 on the basis of the second best model. On GIDS, our AUC is **90.3**, improving 1.2 compared with the second best. Please refer to Table 3 for details.

Table 4 shows P@N in top ranked samples and micro-F1 values on NYT10. Our model improves 3.8 on average for P@N, indicating that our model does not reduce the precision while improving the recall. Table 5 shows the results on GIDS. From the table, we observe that FAN achieves the highest score in micro-F1. The scores of P@N are also comparable to RESIDE and BGWA.

5.2 Mining Results

In the mining step, we filter FN samples from $\mathcal{N}$ with logits larger than threshold θ. As a result, we discover 4,556 FN samples from NYT10, which refer to 3,733 entity pairs; and 238 FN samples from GIDS, which refer to 225 entity pairs.

To evaluate the quality of $\mathcal{M}$, we choose five relations assigned by FAN with most samples, each

selecting 100 sentences with highest confidence scores. Three well-trained NLP annotators are asked to annotate sentences in a binary way, to see whether the assigned pseudo labels are correct. The results are shown in Table 6. The average precision is 87.0, improving around 17.0 compared with the original NYT10 dataset (Riedel et al., 2010). It verifies both the quality of the mined data and the effectiveness of the aligning step. The "nationality" relation gets relatively lower precision because for some sentences about sports, there are usually more than one person and one country mentioned, the model gets confused in this scenario.

Relations	P@100
/location/location/contains	92.0
/people/person/place_lived	87.0
/people/person/nationality	74.0
/business/person/company	94.0
/location/administrative_division/country	88.0
Average	87.0

Table 6: P@100 for top 5 relations in $\mathcal{M}$.

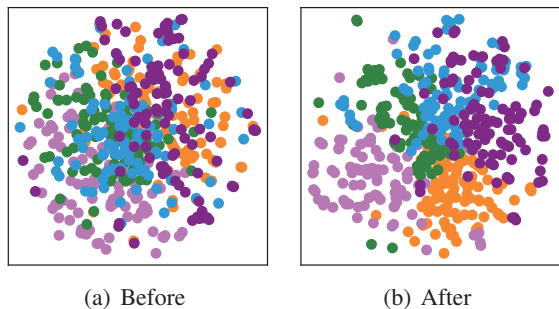(a) Before   (b) After

Figure 4: T-SNE visualization of representations before and after aligning for $\mathcal{M}$.

5.3 Adversarial Domain Adaptation

The label distributions may shift between $\mathcal{M}$ and $\mathcal{P}$. The generator aligns the two distributions into a unified space. We use bag representations obtained through the bag encoder as the input of T-SNE to perform dimension reduction and obtain two-dimensional representations. As seen from Figure 4, the feature distributions before aligning are overlapped and the classification boundary is not clear. After aligning, the samples are better clustered.

5.4 Ablation Study

We conduct an ablation study to verify the effectiveness of submodules in FAN. Table 7 shows the comparison results. (1) BERT has a great influence on the results because it introduces valuable linguistic knowledge and commonsense knowledge to RE. If replacing it by GloVe, both AUC and micro-F1 drop significantly. (2) By removing GRL and unlabeled data, using only the refined training set $\mathcal{D}'$, both AUC and micro-F1 drops. This indicates that the information contained in unlabeled data is helpful for distantly supervised RE, and FAN can leverage it reasonably. (3) Contrastive loss can help the learning procedure. Different classes can be better clustered, thus reducing errors on the classification boundary.

	AUC	Micro-F1
FAN	45.5	48.7
w/o BERT	39.8	44.4
w/o GRL	43.7	47.1
w/o contrastive	44.7	48.0

Table 7: Results of ablation study.
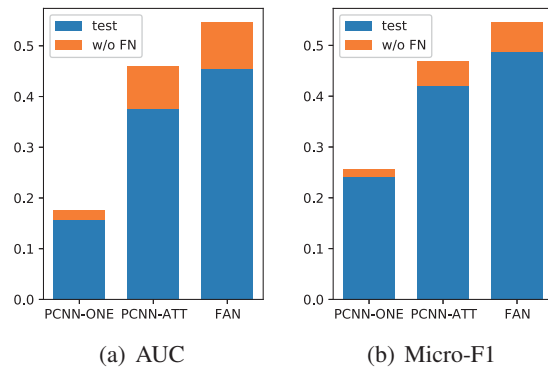

(a) AUC   (b) Micro-F1

Figure 5: Improvements of AUC and micro-F1 after removing FN in the test set.

5.5 False Negatives in the Test Set

We also investigate the impact of FN on the test set during evaluation. As what we do in the training phase, we train a deep filter to mine FN from the test N/A set. As a result, we obtain 6,468 sentences, containing 4,951 entity pairs. By simply removing these data, we obtain huge improvements on AUC by around 20% higher than the original to 54.6, and 12% on micro-F1 to 54.5. See Figure 5. Similar phenomena occur on different baseline models.

This indicates that the FN samples in the N/A set greatly affect the evaluation procedure and bring in improper biases for model selection. For comparison, we randomly remove the same number of sentences from the test N/A set, as a result, AUC increases 0.4 and micro-F1 increases 0.9. This result is rational because randomly removing 6,468 samples is negligible in comparison with 166,004 N/A samples in the test set.

5.6 Case Study

In Table 8, we show several cases of mined data which are excerpted from NYT10.

1. The first sentence is correctly assigned with label "/people/person/place_lived" by FAN, but it is missed by DS-GAN. Because one person is unusual to be mentioned on a Wikipedia page of a location, samples with relations between people and location are greatly omitted.

Among those former clients are two wealthy sons of **[Robert C. McNair]** , a **[Houston]** oilman and the owner of the Houston Texans of the National Football League .
Predicted & gold standard: **/people/person/place_lived**

... the mayor said , defending his advocacy of a new stadium on the Far West Side of **[Manhattan]** for the 2012 Olympics and the **[New York Jets]** .
Predicted & gold standard: **/sports/sports_team/location**

The good news : Marta Caiza of **[Corona]** , **[Queens]** , won a new car .
Predicted: **/location/location/contains**
Gold standard: **N/A**

Table 8: Case study of mined data. **correct prediction** and **wrong prediction** are colored accordingly.

2. For less-frequent relations such as "/sports/sports_team/location", FAN can still identify it to enlarge the training data and weaken the imbalance between relations.

3. In the third sentence, Queens contains Corona, but not reversely. FAN incorrectly assigns "/location/location/contains" from Corona to Queens. In fact, differentiating relation directions is a hard task and needs further study.

6 Conclusion and Future Work

In this paper, we propose FAN, a two-stage method using adversarial DA to handle the FN problem in distantly supervised RE. We mine FN samples using the memory mechanism of deep neural networks. We use GRL to align unlabeled data with training data and generate pseudo labels to correct improper biases in both training and testing procedures. Our experiments show the superiority of FAN against many comparative models. In future work, we plan to use the teacher-student model to deal with FP and FN simultaneously.

Acknowledgements

References

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. In *ACL*, pages 1388–1398.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *ICML*, pages 233–242.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *ACL*, pages 576–583.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2021. Distantly supervised relation extraction with sentence reconstruction and knowledge base priors. *CoRR*, abs/2104.08225.

Christopher De Sa, Alex Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. 2016. DeepDive: Declarative knowledge base construction. *SIGMOD Record*, 45(1):60–67.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge Vault: A web-scale approach to probabilistic knowledge fusion. In *ACM SIGKDD*, pages 601–610.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*, pages 363–370.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8535–8545.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. In *EMNLP-IJCNLP*, pages 169–174.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL-HLT*, pages 541–550.

Sharmistha Jat, Siddhesh Khandelwal, and Partha P. Talukdar. 2018. Improving distantly supervised relation extraction using word and entity based attention. *CoRR*, abs/1804.06987.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *ACL-HLT*, page 1148–1158.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2309–2318.

Junnan Li, Richard Socher, and Steven C.H. Hoi. 2020. DivideMix: Learning with noisy labels as semi-supervised learning. In *ICLR*.

Pengshuai Li, Xinsong Zhang, Weijia Jia, and Hai Zhao. 2019. GAN driven semi-distant supervision for relation extraction. In *NAACL-HLT*, pages 3026–3035.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*, pages 2124–2133.

Guoqing Luo, Jiaxin Pan, and Min Peng. 2020. RDSGAN: Rank-based distant supervision relation extraction with generative adversarial framework. *CoRR*, abs/2009.14722.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling "when to update" from "how to update". In *NIPS*, pages 961–971.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP*, pages 1003–1011.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with Siamese recurrent networks. In *RepL4NLP*, pages 148–157.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. 2020. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. DSGAN: Generative adversarial training for distant supervision relation extraction. In *ACL*, pages 496–505.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML PKDD*, pages 148–163.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2021. Learning from noisy labels with deep neural networks: A survey. *CoRR*, abs/2007.08199.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP*, pages 455–465.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. RESIDE: Improving distantly-supervised neural relation extraction using side information. In *EMNLP*, pages 1257–1266.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, page 6000–6010.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *EMNLP*, pages 1778–1783.

Ya Xiao, Chengxiang Tan, Zhijie Fan, Qian Xu, and Wenye Zhu. 2020. Joint entity and relation extraction with a hybrid transformer and reinforcement learning based model. In *AAAI*, pages 9314–9321.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. Noise mitigation for neural entity typing and relation extraction. In *EACL*, pages 1183–1194.

Yan Yan, Zhongwen Xu, Ivor W. Tsang, Guodong Long, and Yi Yang. 2016. Robust semi-supervised learning through label aggregation. In *AAAI*, pages 2244–2250.

Qinyuan Ye, Liyuan Liu, Maosen Zhang, and Xiang Ren. 2019. Looking beyond label noise: Shifted label distribution matters in distantly supervised relation extraction. In *EMNLP-IJCNLP*, pages 3841–3850.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *ACL*, pages 571–581.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, pages 1753–1762.

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

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *ACL*, pages 207–212.

Wenxuan Zhou, Hongtao Lin, Bill Yuchen Lin, Ziqi Wang, Junyi Du, Leonardo Neves, and Xiang Ren. 2020. NERO: A neural rule grounding framework for label-efficient relation extraction. In *WWW*, page 2166–2176.

A Experiment Setup

In this section, we provide more details of our experiments. We implement FAN with PyTorch 1.6 and train it on a server with an Intel Xeon Gold 5117 CPU, 120GB memory, two NVIDIA Tesla V100 GPU cards and Ubuntu 18.04 LTS.

The parameters of FAN are initialized with Xavier (Glorot and Bengio, 2010) using a fixed initialization seed. We train FAN by SGD optimizer with mini-batch of size 160. In the convolutional module, we set kernel size in $\{2, 3, 4, 5\}$ and filter size to 230. For details, please refer to Table 9.

Hyperparameters	Values
Batch size	160
Learning rate	0.1
Optimizer	SGD
Filter size	230
Kernel size	$\{2, 3, 4, 5\}$
Weight decay	0.00001
α	0.01
β	0.01
γ	0.0001
θ	0.5
τ	1.0
Epoch	50
Dim. of word embeddings	768
Dim. of position embeddings	50
Dropout rate	0.5

Table 9: Hyperparameters of FAN.

B Dataset Availability

The NYT10 dataset (Riedel et al., 2010) is available at http://iesl.cs.umass.edu/riedel/ecml/. We use the version adapted by OpenNRE (Han et al., 2019), which has removed the overlapped samples between the training set and the test set. The data is available at https://github.com/thunlp/OpenNRE. The GIDS dataset is available at https://github.com/SharmisthaJat/RE-DS-Word-Attention-Models.

C False Negatives in N/A

In this section, we give several examples of FN samples mined from the training N/A set. Those samples are assigned with pseudo-labels and confidence scores by FAN. They are diverse and contain many relation types, which verifies that the noises in N/A is not negligible and deserves further study.

On the other hand , a recent lake study by researchers at **[Middlebury College]** , in **[Vermont]** , found nothing similar .
Pseudo-label: **/location/location/contains** 0.999

I need all the help I can get , said **[Chet Culver]** , the Democratic secretary of state in **[Iowa]** and a likely contender for governor in 2006 .
Pseudo-label: **/people/person/place_lived** 0.997

Meb Keflezighi , the American who won the silver medal in the men 's marathon in last year 's Athens Olympics , and **[Liu Xiang]** of **[China]** , who won the men 's 110-meter hurdles in Athens , were entered yesterday in track and field competitions in New York .
Pseudo-label: **/people/person/nationality** 0.999

If they think we 're the weakest link , they 're truly underestimating this little airline , said **[John Denison]** , the new chief executive at **[ATA Airlines]** , which sought bankruptcy protection on Oct. 26 .
Pseudo-label: **/business/person/company** 0.999

The campaign against this poor little church outside **[Hangzhou]** , the capital of Zhejiang Province in eastern **[China]** , is part of a national wave of repression against independent , or underground , churches that are not registered with the government and do not recognize the authority of state-appointed spiritual leaders .
Pseudo-label: **/location/administrative_division/country** 0.999

For investors seeking the high returns that are no longer possible in the mature European and North American real estate markets , India and China are hot , said Prakash Gurbaxani , the chief executive of TSI Ventures in Bangalore , a joint venture of Tishman Speyer Properties of New York and India 's largest privately owned bank , **[ICICI Bank]** , based in **[Mumbai]** .
Pseudo-label: **/business/company/place_founded** 0.971

[Peter Benenson] was born in **[London]** on July 31 , 1921 , the son of a British army colonel .
Pseudo-label: **/people/person/place_of_birth** 0.823

Table 10: Examples of FN.