Empower Distantly Supervised Relation Extraction with Collaborative Adversarial Training

Tao Chen[1], **Haochen Shi**[1], **Liyuan Liu**[2], **Siliang Tang**[1*], **Jian Shao**[1], **Zhigang Chen**[3], **Yueting Zhuang**[1]

[1]Zhejiang University [2]University of Illinois at Urbana Champaign [3]iFLYTEK Research
{ttc, hcshi, siliang, jshao, yzhuang}@zju.edu.cn, llychinalz@gmail.com, zgchen@iflytek.com

Abstract

With recent advances in distantly supervised (DS) relation extraction (RE), considerable attention is attracted to leverage multi-instance learning (MIL) to distill high-quality supervision from the noisy DS. Here, we go beyond label noise and identify the key bottleneck of DS-MIL to be its *low data utilization*: as high-quality supervision being refined by MIL, MIL abandons a large amount of training instances, which leads to a low data utilization and hinders model training from having abundant supervision. In this paper, we propose collaborative adversarial training to improve the data utilization, which coordinates virtual adversarial training (VAT) and adversarial training (AT) at different levels. Specifically, since VAT is label-free, we employ the instance-level VAT to recycle instances abandoned by MIL. Besides, we deploy AT at the bag-level to unleash the full potential of the high-quality supervision got by MIL. Our proposed method brings consistent improvements ($\sim$ 5 absolute AUC score) to the previous state of the art, which verifies the importance of the data utilization issue and the effectiveness of our method.

Introduction

Relation extraction (RE) aims at identifying the relation between entities within a specific context and provides essential support for many downstream tasks. As the performance of RE systems is generally limited by the amount of training data, recent RE systems typically resort to distant supervision (DS) to fetch abundant training data by aligning knowledge bases (KBs) and texts. Since this strategy inevitably introduces label noise to model training, how to neutralize the label noise has been viewed as the major problem of DS.

Multi-instance learning (MIL) was introduced to handle label noise (Zeng et al. 2015; Lin et al. 2016) and has received a significant amount of attention. Specifically, MIL clusters training instances into bags. For each bag, MIL demotes its low-quality instances to eliminate label noise and refines high-quality instances as the bag-level representation for model training.

Here, we go beyond label noise and identify the key bottleneck of DS-MIL to be its low data utilization. In order to distill high-quality supervision from DS, MIL only focuses on a few representative instances (with high attention scores) and abandons a large proportion of low-score instances. As in Figure 1, except the situation that one bag only contains one instance (with attention scores of 1.0), most of the instances are assigned with low attention scores (0.0$\sim$0.2) and abandoned during the training process. Specifically, as in Table 2, control experiments show that even if some low-score instances are removed, the newly trained model has a limited performance change. In other words, although DS leads to abundant training instances, MIL fails to unleash the full potential of DS, since it abandons the majority of training instances.

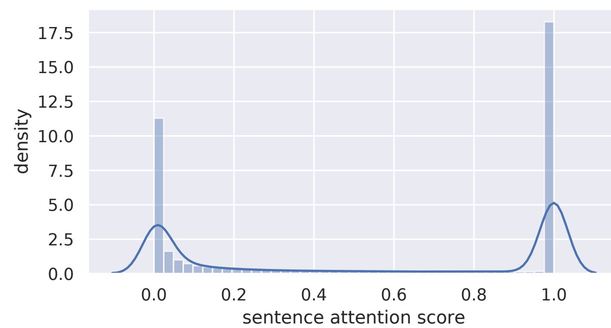

Figure 1: Sentence attention score distribution inside the bag during training process: Most instances are with low scores, instances with high attention scores (not including 1.0) only dominate a small part of data.

Here, we propose MULTICAST (**MULT**i-**I**nstance **C**ollaborative **A**dver**S**arial **T**raining) to improve the data utilization. It coordinates adversarial training (AT) (Goodfellow, Shlens, and Szegedy 2014) and virtual adversarial training (VAT) (Miyato, Dai, and Goodfellow 2016) at different levels. In detail, as the MIL framework intrinsically splits training data into two classes (i.e., high-quality instances for constructing bag-level representations and low-quality instances abandoned by MIL), we use different strategies on them. For low-quality instances, although their associated labels are not very reliable, they can still provide valuable information for label-free regularization objectives. Thus, we apply instance-level virtual adversarial training (IVAT)

*Corresponding author

to exploit entity and context information without using their unreliable label information. For high-quality instances, we try to compensate their loss of quantity (caused by MIL). Specifically, we apply bag-level adversarial training (BAT) to further regularize the constructed representations and unleash the full potential of these high-quality instances.

We conduct experiments on NYT (Riedel, Yao, and McCallum 2010), the public DSRE benchmark. MULTICAST leads to consistent improvements over the previous state-of-the-art systems. It demonstrates the effectiveness of MULTICAST and validates our intuition that the data utilization issue is the key bottleneck. We further conduct ablation studies to verify that MULTICAST coordinates different modules effectively. The major contributions of this paper are summarized as follows:

- We identify the low data utilization issue as the major bottleneck of DS-MIL.
- We propose MULTICAST to boost data utilization. It coordinates VAT and AT at different levels based on MIL signals (attention scores).
- Controlled experiments verify our intuitions and show that MULTICAST leads to consistent improvements (~ 5 absolute AUC score).

Related Work

In the field of distantly supervised relation extraction, the multi-instance learning framework is introduced to handle the label noise of DS. Recently, MIL has become a common paradigm for DSRE and many efforts have been made for further improvements (Lin et al. 2016; Qin, Xu, and Wang 2018; Ye and Ling 2019; Yuan et al. 2019; Huang and Du 2019; Ye et al. 2019; Shang et al. 2020). In these MIL frameworks, sentences are first encoded by handcrafted features (Mintz et al. 2009; Hoffmann et al. 2011) or neural networks. Then, multiple instances are leveraged to form a bag-level representation, which has better quality. With regard to the strategy for selecting instances inside the bag, a soft attention mechanism (Lin et al. 2016) is widely used for its better performance than the hard selection way.

Based on the multi-instance learning framework, most previous work focus on further improving the strategy to handle label noise. Specifically, Ye and Ling (2019); Yuan et al. (2019) both adopted a relation-aware selective attention mechanism inside the bag, and constructed a superbag which contains a group of bags to alleviate the issue of bag label error. Focusing on transforming the network structure, Huang and Du (2019) utilized recent self-attention mechanism (Vaswani et al. 2017) integrated with convolutional neural networks (CNNs) to obtain a better sentence representation from the noisy inputs, and this work also applied cooperative curriculum learning to constrain student models which can learn from each other. At the same time, few attempts have been made on other aspects of DSRE, i.e., Ye et al. (2019) found that the problem of shifted label distribution influences the performance of DSRE models significantly. Similar to our study, Shang et al. (2020) observe that noisy sentences inside the bag are not useless and developed a way to relabel the noisy data by employing unsupervised deep clustering.

At the same time, adversarial training has been found to be useful for DSRE. Wu, Bamman, and Russell (2017) firstly introduced adversarial training (Goodfellow, Shlens, and Szegedy 2014; Miyato, Dai, and Goodfellow 2016) to relation extraction by generating adversarial noise to the training data. Qin, Xu, and Wang (2018) leverages generative adversarial networks (GANs), i.e., it adopts the trained generator to filter the DS training dataset and redistributes the false positive instances into the negative set, in which way to provide a cleaned dataset for relation classification.

Methodology

In our paper, we identify the low data utilization as the key bottleneck of DS-MIL. As MIL forms accurate bag representations to handle label noise, it abandons a large amount of training instances. Typically, MIL faces the dilemma that label noise reduction sacrifices the data utilization. Here, we go beyond typical DS-MIL and propose collaborative adversarial training to improve the data utilization. The diagram of our method (MULTICAST) is visualized in Figure 2, which contains five components: (1) input representations; (2) sentence encoder; (3) attention-based MIL framework; (4) instance-level virtual adversarial training module; (5) bag-level adversarial training module.

Inputs: Embeddings

For each word t_i in sentence s, we employ word embedding $w_i \subset \mathbb{R}^{d_w}$ to capture its semantic information. Moreover, to encode the sentence in an entity-aware manner, relative position embedding (Zeng et al. 2015) is leveraged to represent the position information in the sentence. Relative distances d_{i1}, d_{i2} of word t_i correspond to the distances between t_i and two entities e_1 and e_2, and can be transferred to position vectors $p_{i1}, p_{i2} \subset \mathbb{R}^{d_p}$ by looking up a position embedding table. This embedding table is initialized randomly and updated during the training process. Concatenating the above two embeddings, each word t_i can obtain its entity-aware representation as $m_i = [w_i; p_{i1}; p_{i2}] \subset \mathbb{R}^d$. Thus the instance representation can be constructed as $X = [m_1; m_2; \ldots; m_l] \subset \mathbb{R}^{l \times d}$, where $d = d_w + 2 \cdot d_p$ and l is the maximum length of the sentences.

Encoder: Piecewise CNN

Convolutional neural networks capture the sentence semantics with sliding windows. In the convolutional layer, the embedding window $X_{t:t+u} = [m_t; m_{t+1} \ldots; m_{t+u-1}] \subset \mathbb{R}^{u \times d}$ interacts with convolution kernels $\{W_1, \ldots, W_p\} \subset \mathbb{R}^{u \times d}$ to extract sentence-level features, where u is the width of kernel and p is the number of kernels.

Followed by max-pooling layer, the most responsive region of convolutional output $C \subset \mathbb{R}^{l \times p}$ is retained. Instead of just using a unified pooling layer, Zeng et al. (2015) applied max-pooling operation to different pieces of sentence respectively, which has been proved to better capture structured information between two entities. The final feature vector $H \subset \mathbb{R}^{3 \times p}$ can be obtained by concatenating all pooling results of three pieces.

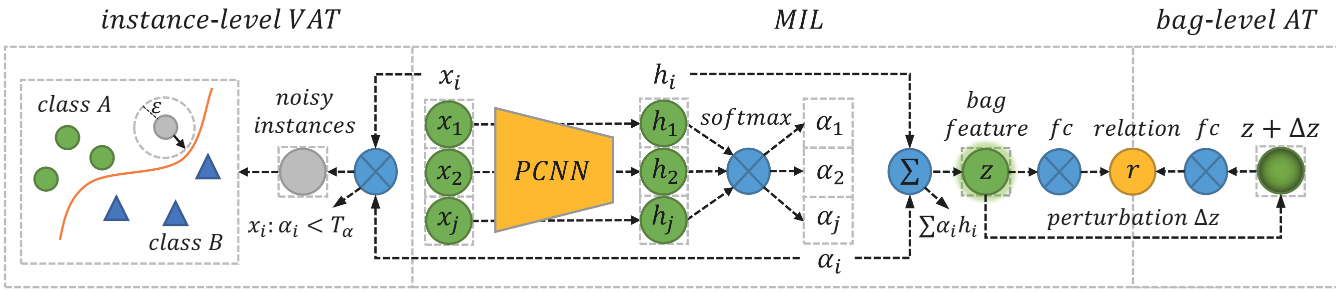

Figure 2: (a) Instances $x_1, x_2 \ldots x_j$ inside the bag firstly encode themselves by piecewise convolutional neural networks, and obtain sentence-level representations $h_1, h_2 \ldots h_j$. Based on the MIL framework, selective attention is adopted to form better bag-level representations $z = \sum_i \alpha_i h_i$ over instances. (b) Inside the bag, those noisy or unrepresentative instances with lower attention score α_i are selected $\{x_i | \alpha_i < T_\alpha\}$ for additional virtual adversarial training. (c) Outside the bag, reliable bag-level representations z are further enhanced via adversarial learning.

MIL: Multi-Instance Learning

For a model parameterized by θ, input representation $x_i \in X$ of each sentence s_i in bag B can be encoded to feature vector $h_i \in H$, then multi-instance learning framework considers all instances inside the bag to get a relatively accurate representation z, which is defined as:

$$z = \sum_i \alpha_i h_i$$

As for the weight α, we adopt a soft attention mechanism as in Lin et al. (2016), where α_i is the normalized attention score calculated by a query-based function f_i which measures how well the sentence representation h_i and the predict relation r matches:

$$\alpha_i = \frac{e^{f_i}}{\sum_j e^{f_j}}$$

where $f_i = h_i A q_r$, A is a weighted diagonal matrix and q_r is the query vector which indicates the representation of relation r (randomly initialized).

Then, based on this bag-level representation, a simple fully-connected layer with activation function *softmax* is added to map the feature vector z to a conditional probability distribution $p(r|Z, \theta)$:

$$p(r|Z,\theta) = \frac{e^{o_r}}{\sum_{i=1}^{n_r} e^{o_i}}$$

where $o = Mz + b$ is the score associated to all relation types, n_r is the total number of relations, M is a projection matrix and b is the bias term.

Finally, we define the objective function of MIL framework using cross-entropy as follows:

$$J(\theta) = -\sum_{i=1} \log p(r_i|z_i, \theta)$$

IVAT: Instance-Level Virtual Adversarial Training

In MIL, the normalized attention score α_i describes how much the instance x_i contribute to the final representation z. A higher value indicates the instance is cleaner or more representative, while a lower value implies the instance is noisy (i.e., its relation label is not reliable). In other words, the attention score is the label quality signal used in MIL.

We refer instances with high attention scores as X_{clean} (i.e., clean instances) and instances with low attention scores as X_{noisy} (noisy instances). As discussed in section Introduction, MIL mainly focuses on X_{clean} and abandons X_{noisy} during the training. To improve the data utilization of MIL, we introduce virtual adversarial training at the instance level to exploit entity and context information from X_{noisy}. Now we proceed to introduce module details.

For instances $\{x_1, x_2, \ldots, x_i\}$ inside bag B, we use $\{\alpha_1, \alpha_2, \ldots, \alpha_i\}$ to refer their normalized attention scores (outputs of the selective attention in section MIL). Then, we leverage a hyperparameter T_α to identify instances that are ignored by MIL:

$$X_{noisy} = \{x_i | \alpha_i < T_\alpha\}$$

For instance $x \in X_{noisy}$, we refer its conditional probability distribution output to be $p(y|x,\theta)$. Then, we refer its representation under a small perturbation $||d|| \leq \epsilon_x$ to be $x + d$, and the corresponding model output to be $p\left(y|x+d,\theta\right)$. These two outputs are regularized to be similar, i.e.,

$$l_{\text{ivat}}\left(d,x,\theta\right) := \text{KL}\left[p\left(y|x,\theta\right) \| p\left(y|x+d,\theta\right)\right]$$

where KL is the Kullback–Leibler divergence which measures the similarity of two probability distributions. As to the adversarial perturbation d_{v-adv}, its ideal choice should be the direction maximizing l_{ivat}, i.e.,

$$d_{v-adv} := \arg\max_d \left\{l_{\text{ivat}}\left(d,x,\theta\right); \|d\|_2 \leq \epsilon_x\right\}$$

Following previous work (Miyato et al. 2018), we employ an efficient way to estimate d_{v-adv} under L_2 norm:

$$d_{v-adv} \approx \epsilon_x \frac{g}{||g||_2}$$

where $g = \nabla_r \text{KL}\left[p(y|x,\theta), p(y|x+r,\theta)\right]|_{r=\xi d}$ with $\xi > 0$ and d is a randomly sampled unit vector. For neural networks, this approximation can be performed with K sets of back-propagations.

With such a perturbation d_{v-adv}, our objective is to make the local distributional smoothness (LDS) of the model as high as possible, this is defined as:

$$\text{LDS-X}(\theta) := -\sum_{x \in X_{noisy}} l_{\text{ivat}}(d_{v-adv}, x, \theta)$$

BAT: Bag-Level Adversarial Training

Different from noisy instances, high-quality instances are used to construct the bag-level representation z, which better matches the associated relation and allows MIL to reduce the impact of label noise. Here we leverage adversarial training to unlease the full potential of that high-quality supervision.

Specifically, we add a perturbation d to the bag-level representation z instead of word embedding x. Different from IVAT, we employ the training label instead of the original output to regularize the output under perturbation, *i.e.*,

$$l_{\text{bat}}(d, z, \theta) := -\log p(r|z + d, \theta)$$

Similar to the virtual adversarial perturbation d_{v-adv} in section IVAT, adversarial perturbation d_{adv} is in the direction with maximum model output change, which is further defined as:

$$d_{adv} := \arg\max_{d} \{l_{\text{bat}}(d, z, \theta); \|d\|_2 \leq \epsilon_z\}$$

Generally, a linear approximation (Goodfellow, Shlens, and Szegedy 2014; Miyato, Dai, and Goodfellow 2016) of adversarial perturbation vector d_{adv} under L_2 norm (Fast Gradient Method) is:

$$d_{\text{adv}} \approx \epsilon_z \frac{g}{\|g\|_2}$$

where $g = \nabla_z \log p(r|z, \theta)$, which can be efficiently computed by back-propagation in neural networks. With such a perturbation, our maximization objective is marked as:

$$\text{LDS-Z}(\theta) := \sum_{z} l_{\text{bat}}(d_{adv}, z, \theta)$$

Objective

Considering the original objective of MIL framework mentioned in section MIL, and two regularization terms at instance level and bag level, the overall maximization objective function of our method is:

$$\mathcal{L} = J(\theta) + \beta_1 \,\text{LDS-X}(\theta) + \beta_2 \,\text{LDS-Z}(\theta)$$

where $\beta_1 > 0$ and $\beta_2 > 0$ are the weight coefficients corresponding to the modules IVAT and BAT. Module IVAT uses a hyperparameter T_α to decide extra-learning data ratio. Empirically, the value of β_1 is closely related to the value of parameter T_α (larger $T_\alpha \sim$ larger β_1).

Experiments

Our experiments are designed to verify the effectiveness of the proposed method — MULTICAST.

Dataset

We evaluate our model on the widely used DSRE dataset — NYT (Riedel, Yao, and McCallum 2010), which aligns Freebase (Bollacker et al. 2008) entity relation with New York Times corpus. This dataset uses the corpus from 2005 to 2006 as the training set, and employs the data of 2007 as a test set. In detail, the training set consists of 522,611 sentences, 281,270 entity pairs and 18,252 relation facts, while the testing set contains 172,448 sentences, 96,678 entity pairs, and 1,950 relation facts. For relation labels, this dataset supports 53 different relations including NA which means no relation between an entity pair.

It is worth noting that, some previous work (Wu, Bamman, and Russell 2017; Qin, Xu, and Wang 2018; Ye and Ling 2019) use another dataset which contains 578,288 sentences in the training set. In fact, that dataset is inaccurate because there is considerable training data and test data overlaps. This bug was fixed in March 2018, and all recent work (Huang and Du 2019; Shang et al. 2020) since then adopt the correct dataset as the benchmark. In order to ensure the fairness and scientificity of our experiments, we use the original dataset release in our study and employ the popular relation extraction toolkit OpenNRE (Han et al. 2019).

Evaluation Metrics

Following previous literature (Riedel, Yao, and McCallum 2010; Zeng et al. 2015; Lin et al. 2016), we conduct held-out evaluation. Specifically, Precision-Recall curves (PR-curve) are drawn to show the trade-off between model precision and recall, the Area Under Curve (AUC) metric is used to evaluate the overall model performances, and the Precision at N (P@N) metric is also reported to consider the accuracy value for different cut-offs (default using all sentences for each entity pair while testing). Besides, we also conduct human evaluation to further support our claims. We adopt the test set used in the existing literature (Hoffmann et al. 2011), which contains 395 sentences with human annotations.

Baseline Models

We choose six recent methods as baseline models.

- PCNN-ATT (Lin et al. 2016) uses selective attention to reduce the weights of noisy instances.
- PCNN-ATT+ADV (Wu, Bamman, and Russell 2017) adds adversarial noise to DS training data.
- PCNN-ATT+DSGAN (Qin, Xu, and Wang 2018) utilizes GANs to remove potentially inaccurate sentences from the original training dataset.
- PCNN-ATT-RA+BAG-ATT (Ye and Ling 2019) uses intra-bag and inter-bag attentions to deal with the noise at sentence-level and bag-level.
- PCNN-ATT+SELF-ATT+[CCL-CT] (Huang and Du 2019) integrates a self-attention mechanism into the CNN structure and defines two student models for collaborative curriculum learning.
- PCNN-ATT+DC (Shang et al. 2020) employs unsupervised deep clustering to generate reliable labels for noisy sentences.

Method	AUC	P@100	P@200	P@300	P@Mean
PCNN-ATT (Lin et al. 2016)	34.13	73.0	69.0	66.0	69.3
PCNN-ATT+ADV (Wu, Bamman, and Russell 2017)	34.99	80.2	72.1	69.4	73.9
PCNN-ATT-RA+BAG-ATT (Ye and Ling 2019)	35.03	77.0	75.5	72.3	74.9
PCNN-ATT+DSGAN (Qin, Xu, and Wang 2018)	35.19	76.2	70.7	68.4	71.8
PCNN-ATT+SELF-ATT* (Huang and Du 2019)	36.80	81.1	71.6	70.4	74.4
PCNN-ATT+SELF-ATT+CCL-CT* (Huang and Du 2019)	38.10	82.2	79.1	73.1	78.1
PCNN-ATT+MULTICAST (Ours)	**38.78±0.15**	**83.7±1.5**	**79.2±1.0**	**74.2±0.7**	**79.0±0.6**

Table 1: Performances of all compared models. Models marked with * are quoted from original papers, since there are no open-source codes released.

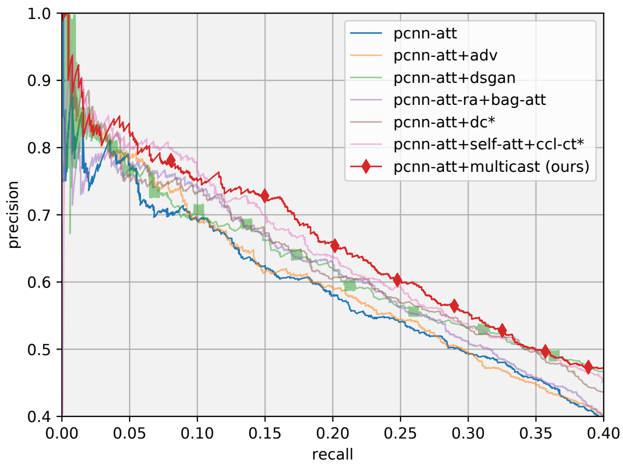

Figure 3: PR-Curve. Models with * directly quote the drawn curves from the corresponding papers.

Overall Comparison

We summarize the model performances of our method and above-mentioned baseline models in Table 1. From the results, we can observe that: (1) With the help of our proposed modules (MULTICAST, *i.e.*, IVAT+BAT), the vanilla baseline model PCNN-ATT achieves the best performance in all five metrics. (2) Compared with the standard baseline model PCNN-ATT, MULTICAST improves the metric AUC (34.13→38.78) by 13.6% and the metric P@Mean (69.3→79.0) by 14.0%.

The overall PR-curve is visualized in Figure 3. From the curve, we can observe that: (1) Compared to the PR-curve of standard baseline model PCNN-ATT, our method shifts up the curve a lot. (2) Our method surpasses current SOTA model in almost all ranges (except when the recall is between 0.05 and 0.10) along the curve.

Controlled Experiment

We identify the *low data utilization* issue as the key bottleneck of DS-MIL. To verify that those low-score sentences are not used by the model, we remove these sentences from the training set with different thresholds (e.g. $\alpha_i < 0.1, 0.2$), and use the reduced dataset to re-train PCNN-ATT models and our proposed models.

Dataset Size	Method	AUC
522611 (unfiltered)	PCNN-ATT	34.13
	+MULTICAST	38.93
334194(-36%) (filtered @ 0.1)	PCNN-ATT	33.87(-0.7%)
	+MULTICAST	36.50(-6.2%)
310039(-41%) (filtered @ 0.2)	PCNN-ATT	33.70(-1.3%)
	+MULTICAST	36.24(-6.9%)

Table 2: Model performances of the original dataset and reduced dataset

We summarize model performances on the original dataset and reduced dataset in Table 2. With the significant reduction in the amount of data, the MIL method PCNN-ATT only has subtle performance changes (i.e., yielding $\sim 1\%$ performance loss). It verifies our intuition that MIL abandons these instances and ignores them during training. Besides, our method has a noticeable large performance drop (38.93→36.50) after removing these training instances. It verifies that our proposed method effectively recycles abandoned training instances thus leading to a better data-utilization.

Human Evaluation

We also evaluate our proposed method MULTICAST on the human-annotated dataset and results are listed in Table 3.

From the table we can observe that: (1) Our method can still significantly improve model performance under accurate human evaluation. (2) Compared to other baseline models, our method can generalize better.

Ablation Study

We further conduct ablation studies to verify the effectiveness of our proposed modules respectively.

As to module IVAT, it is designed for utilizing training instances that are abandoned by the MIL framework. Intu-

Method	AUC	F1
PCNN-ATT	38.91	46.98
PCNN-ATT+DSGAN	43.51(+4.60)	47.49(+0.51)
PCNN-ATT+MULTICAST	**46.03(+7.12)**	**50.29(+3.31)**

Table 3: Model performances of human-annotated dataset

Method	AUC	P@100	P@200	P@300	P@Mean
PCNN-ATT (Lin et al. 2016)	34.13	73.0	69.0	66.0	69.3
+BAT	35.10(+0.97)	79.0(+6.0)	77.5(+8.5)	70.7(+4.7)	75.7(+6.4)
+IVAT	37.97(+3.84)	81.2(+8.2)	77.6(+8.6)	73.1(+7.1)	77.3(+8.0)
+IVAT+BAT	**38.93**(+4.80)	**86.2**(+13.2)	**78.6**(+9.6)	**74.1**(+8.1)	**79.6**(+10.3)
PCNN-ATT-RA+BAG-ATT (Ye and Ling 2019)	35.03	77.0	75.5	72.3	74.9
+IVAT*	**38.23**(+3.20)	**87.0**(+10.0)	**82.5**(+7.0)	**75.3**(+3.0)	**81.6**(+6.7)
PCNN-ATT+DSGAN (Qin, Xu, and Wang 2018)	35.19	76.2	70.7	68.4	71.8
+BAT	36.24(+1.05)	79.2(+3.0)	73.1(+2.4)	71.8(+3.4)	74.7(+2.9)
+IVAT	39.21(+4.02)	84.2(+8.0)	77.6(+6.9)	73.4(+5.0)	78.4(+6.6)
+IVAT+BAT	**40.85**(+5.66)	**86.2**(+10.0)	**81.1**(+10.4)	**74.4**(+6.0)	**80.6**(+8.8)

Table 4: Ablation study with three baseline models. The model marked with * does not have bag representation and cannot be integrated with BAT (it employs a two-layer attention mechanism to get relation-aware embedding).

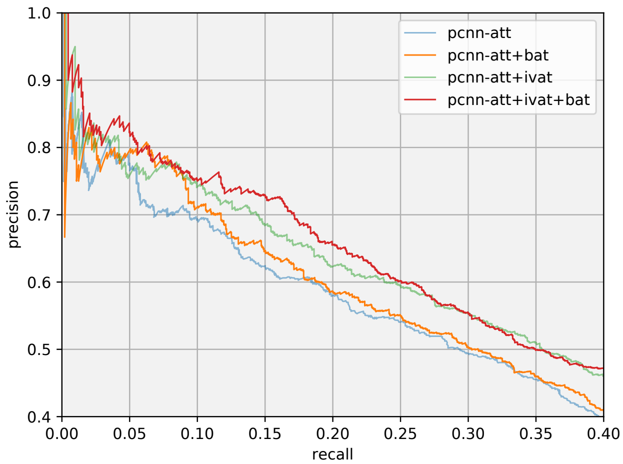

Figure 4: PR-curves of modules IVAT and BAT

itively, improvements from this module are orthogonal to attempts aiming to further improve the supervision quality. Thus, we add this module to three baselines and summarize their performances in Table 4. Module IVAT brings stable and significant improvements to different baseline models in all metrics. For the standard baseline model PCNN-ATT, with the IVAT module alone, its AUC score is already close to the current SOTA model (37.97∼38.10). For another two baseline models, module IVAT also leads to consistent performance improvements. For instance, with IVAT, all metrics of the method PCNN-ATT-RA+BAG-ATT have exceeded the SOTA model (e.g., its P@N score and AUC score reaches 81.6 and 38.23, while the SOTA gets 78.1 and 38.10).

For module BAT, it aims to make full use of high-quality representations at bag level. As this module still relies on supervised label information (i.e., the more accurate representation is, the more model performance can be enhanced), it has fewer performance improvements than IVAT. Still, this module leads to consistent performance improvements on both baselines that have bag-level representations. Besides, we find the two modules are not exclusive (0.97+3.84∼4.80, 1.05+4.02∼5.66).

To better understand the effects of two modules, we draw their PR-curves in Figure 4. From the figure we can observe that: (1) the IVAT module significantly raises the curve of baseline model in all ranges. (2) the BAT module has a larger benefit with a higher precision score. This observation further verifies our intuitions.

Discussion About AT and VAT

Our method MULTICAST leverage two strategies to coordinate AT and VAT: (1) instead of adding AT/VAT to all instances, MIL attention signals are leveraged to recognize the proper subset to apply these techniques. (2) instead of applying both AT and VAT at both levels, we only apply AT at the bag level and VAT at the instance level.

Effectiveness of Level Selection Classical methods apply AT (Wu, Bamman, and Russell 2017)/VAT to all instances without any selection. In contrast, MULTICAST applies AT and VAT at different levels. To verify the effectiveness of this strategy, we conduct comparison to the conventional methods and summarize the results in Table 5. The gap between adding AT to all instances and adding AT to bag features are marginal. Intuitively, these two methods are very similar to each other, while adding AT at the bag-level is faster (no need to conduct back-propagate to the embedding layer). On the other hand, adding VAT to all more instances (which would also be slower) performs worse than only adding VAT to abandoned instances. It verifies that the context information of high-quality instances are already utilized by the training algorithm, and there is no need to apply VAT on these instances.

Method	Level	AUC
PCNN-ATT	-	34.13
PCNN-ATT+AT	all instances	34.99(+0.86)
	bag features	**35.10(+0.97)**
PCNN-ATT+VAT	all instances	37.35(+3.22)
	noisy instances	**37.97(+3.84)**

Table 5: Discussion of different level selection ways

KB Fact: (*lebron james lived_in akron*) Bag Label: */people/person/place_lived*				
Sentences	Attention Score		Sentence Label	
	w/o BAT	w/ BAT	w/o IVAT	w/ IVAT
an estimated 40,000 ohio state fans came to town, including the **akron** native **lebron james**, giving this quintessential college town ...	0.59	0.71	lived_in	lived_in
bynum is not another **lebron james**, the high school phenomenon from **akron**, ohio, who was the top draft pick in 2003 and immediately ...	0.19	0.13	NA	borned_in
lebron james and his friends used to drive from **akron**, ohio, fill a few of the empty aquamarine seats in cleveland's downtown ...	0.22	0.16	lived_in	NA

Table 6: Case study of how modules IVAT and BAT work

Method	AUC
PCNN-ATT	34.13
+Instance-Level AT+Bag-Level VAT	32.34(-1.79)
+Instance-Level AT+Bag-Level AT	34.16(+0.03)
+Instance-Level VAT+Bag-Level VAT	36.36(+2.23)
+Instance-Level VAT+Bag-Level AT	**38.93(+4.80)**

Table 7: Discussion of different collaboration ways

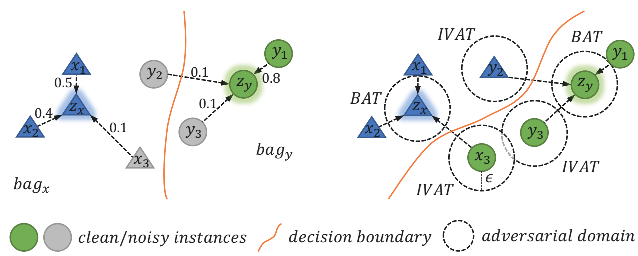

Figure 5: Effect diagram of modules IVAT and BAT

Effectiveness of Collaboration After clarifying the choice of level selection, we proceed to consider the cooperation strategy between AT and VAT (results are summarized in Table 7): (1) For instance-level noisy data, AT may amplify the effects of wrong labels and results in severe *confirmation bias problem* (Tarvainen and Valpola 2017), which makes the model converge too fast and learn nothing extra (34.13∼34.16). (2) For bag-level high-quality features, VAT may weaken the original supervised information provided by MIL framework and complicates model training (38.93→36.36, 34.16→32.34). Comparing Table 5 and Table 7, Instance-Level AT and Bag-Level VAT actually has a negative impact on the model performance (35.10→34.16, 37.35→36.36).

Case Study and Visualization

In order to better understand how BAT and IVAT work, we conduct case studies and visualization.

In our method MULTICAST, both IVAT and BAT improve model performance, but the ways they work are entirely different. We select a typical bag to illustrate their roles respectively: (1) For the bag (see Table 6) with KB fact (*lebron james lived_in akron*), it consists of three different sentences. Module IVAT pays its attention to these low-score (0.19, 0.22) sentences. With the help of the IVAT module, these sentences are allowed to rethink their probability distributions without considering their noisy labels. For example, although the $3rd$ sentence (*lebron james and his friends used to drive from akron* ...) mentions the entity pair (*lebron*, *james*), it actually fails to express the relation *live_in*. With the help of IVAT, this instance succeeds to realize the error and find its true label to be NA. (2) Meanwhile, the BAT module focuses on accurate bag features formed by high-quality instances. In this bag, the final representation is mainly composed of the $1st$ sentence (... *including the akron native lebron james*), which is representative enough to express current bag label *lived_in*. After the adversarial enhancement at bag level, the model is more confident in the high-quality instance with higher attention score (the representation of the $1st$ instance is near to the bag-level representation).

Moreover, we draw a diagram to better illustrate their mechanisms. On the left side of Figure 5, original DS-MIL only uses bag features z_x and z_y for training and obtains a decision boundary without considering noisy instances like y_2. Thus, the resulting model may not be trained with abundant instances and have issues like shifted label distribution (Ye et al. 2019). On the right side, IVAT helps instances x_3 and y_2 find their correct labels. It works with BAT to smooth model outputs in their respective adversarial domains, which prompts the model to generate a better classification boundary.

From the above diagram we can also see that, IVAT acts on those noisy instances (x_3, y_2, y_3), which are far away from the targets of module BAT — bag features z_x, z_y. Therefore, the adversarial domains of modules BAT and IVAT only have limited overlap, which provides an intuitive explanation for why the effects of two modules are orthogonal (see Table 4, 0.97+3.84∼4.80, 1.05+4.02∼5.66).

Conclusion

In this paper, we propose Multi-Instance Collaborative Adversarial Training (MULTICAST) to alleviate the problem of low data utilization under MIL framework. Experiments have shown the effectiveness of our method with stable and significant improvements over several different baseline models, including current SOTA systems.

Acknowledgments

This work has been supported in part by National Key Research and Development Program of China (2018AAA010010), Zhejiang NSF (LR21F020004), Zhejiang University iFLYTEK Joint Research Center, Funds from City Cloud Technology (China) Co. Ltd., Zhejiang University-Tongdun Technology Joint Laboratory of Artificial Intelligence, Chinese Knowledge Center of Engineering Science and Technology (CKCEST), Hikvision-Zhejiang University Joint Research Center, the Fundamental Research Funds for the Central Universities, and Engineering Research Center of Digital Library, Ministry of Education.

References

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* .

Han, X.; Gao, T.; Yao, Y.; Ye, D.; Liu, Z.; and Sun, M. 2019. OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, 169–174.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 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*, 541–550. Association for Computational Linguistics.

Huang, Y.; and Du, J. 2019. Self-Attention Enhanced CNNs and Collaborative Curriculum Learning for Distantly Supervised Relation Extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 389–398.

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

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 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*, 1003–1011. Association for Computational Linguistics.

Miyato, T.; Dai, A. M.; and Goodfellow, I. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725* .

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* 41(8): 1979–1993.

Qin, P.; Xu, W.; and Wang, W. Y. 2018. DSGAN: Generative Adversarial Training for Distant Supervision Relation Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 496–505.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 148–163. Springer.

Shang, Y.; Huang, H. Y.; Mao, X.; Sun, X.; and Wei, W. 2020. Are Noisy Sentences Useless for Distant Supervised Relation Extraction? In *AAAI*, 8799–8806.

Tarvainen, A.; and Valpola, H. 2017. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR abs/1703.01780. arXiv* 1703: 01780.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wu, Y.; Bamman, D.; and Russell, S. 2017. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1778–1783.

Ye, Q.; Liu, L.; Zhang, M.; and Ren, X. 2019. Looking Beyond Label Noise: Shifted Label Distribution Matters in Distantly Supervised Relation Extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3832–3841.

Ye, Z.-X.; and Ling, Z.-H. 2019. Distant Supervision Relation Extraction with Intra-Bag and Inter-Bag Attentions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2810–2819.

Yuan, Y.; Liu, L.; Tang, S.; Zhang, Z.; Zhuang, Y.; Pu, S.; Wu, F.; and Ren, X. 2019. Cross-relation cross-bag attention for distantly-supervised relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 419–426.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1753–1762.