Deep Ranking Based Cost-sensitive Multi-label Learning for Distant Supervision Relation Extraction

Hai Ye, Zhunchen Luo and Wenhan Chao†

***Abstract*—Distant supervision relation extraction suffers the problem of *relation overlapping* in which one entity tuple may have multiple relation facts. We believe that relation classes can have latent connections, which we call *class ties*, and can be exploited to enhance relation extraction. However, this property between relation classes has not been fully explored before. In this paper, to exploit class ties between relations to improve relation extraction, we propose a general ranking based multi-label learning framework combined with convolutional neural networks, in which ranking based loss functions with regularization technique are introduced to learn the latent connections between relations. Furthermore, to relieve the *class imbalance* problem in distant supervision relation extraction, we further adopt cost-sensitive learning to rescale the costs from the positive and negative labels. Extensive experiments on a widely used dataset show the effectiveness of our model to exploit class ties and to relieve class imbalance problem. Our model achieves state-of-the-art performance.**

***Index Terms*—Distant supervision relation extraction, class ties, class imbalance, multi-label learning, cost-sensitive learning, deep ranking, convolution neural networks**

I. Introduction

RELATION extraction (RE) aims to classify the relations between two given named entities from natural-language text. Table I shows the process of RE between the entity tuple *(Patsy Ramsey, Atlanta)*. Supervised machine learning methods require numerous labeled data to work well. With the rapid growth of volume of relation types, traditional methods can not keep up with the step for the limitation of labeled data. In order to narrow down the gap of data sparsity, [1] propose *distant supervision (DS)* for relation extraction, which automatically generates training data by aligning a knowledge facts database (ie. Freebase [2]) with texts.

Class ties mean the connections between relations in relation extraction. In general, we conclude that class ties can have two types: weak class ties and strong class ties. Weak class ties mainly involve the co-occurrence of relations such as *place_of_birth* and *place_lived*, *CEO_of* and *founder_of*. On the contrary, strong class ties mean that relations have latent logical entailments. Take the two relations of *capital_of* and *city_of* for example, if one entity tuple has the relation of *capital_of*, it must express the relation fact of *city_of*, because the two relations have the entailment of *capital_of* $\Rightarrow$ *city_of*. Obviously the opposite induction is not correct. Further take the following sentence of

> Jonbenet told me that her mother [Patsy Ramsey]$_{e_1}$ never left [Atlanta]$_{e_2}$ since she was born.

for example. This sentence expresses two relation facts which are *place_of_birth* and *place_lived*. However, the word "born" is a strong bias to extract *place_of_birth*, so it may not be easy to predict the relation of *place_lived*, but if we can incorporate the weak ties between the two relations, extracting *place_of_birth* will provide evidence for prediction of *place_lived*.

TABLE I
TRAINING INSTANCES GENERATED BY FREEBASE.

place_lived (Patsy Ramsey, Atlanta)
place_of_birth (Patsy Ramsey, Atlanta)

	Sentence	Latent Label
#1	*Patsy Ramsey* has been living in *Atlanta* since she was born.	*place_of_birth*
#2	*Patsy Ramsy* always loves *Atlanta* since it is her hometown.	*place_lived*

Exploiting class ties is necessary for DS based relation extraction. In DS scenario, there is a challenge that one entity tuple can have multiple relation facts as shown in Table I, which is called *relation overlapping* [3][4]. However, the relations of one entity tuple can have class ties mentioned above which can be leveraged to enhance relation extraction for it narrowing down potential searching spaces and reducing uncertainties between relations when predicting unknown relations. If one pair of entities has *CEO_of*, it will contain *founder_of* with high possibility.

To exploit class ties between relations, we propose to make joint extraction with considering *pairwise* connections between positive and negative labels inspired by [5][6]. As the two relations with class ties shown in Table I, by joint extraction of two relations, we can maintain the *class ties* (co-occurrence) of them from training samples to be learned by potential models, and then leverage this learned information to extract instances with unknown relations. We introduce a ranking based multi-label learning framework to make joint extraction, to learn to rank the probability for prediction of positive relations higher than negative ones, where we design ranking based loss functions for multi-label learning. Furthermore, inspired

• Hai Ye and Wenhan Chao are with the School of Computer Science and Enginnering, Beihang University, Beijing, China. (email: iamyehai@outlook.com, chaowenhan@buaa.edu.cn)
• Zhunchen Luo is with China Defense Science and Technology Information Center, Beijing, China (email: zhunchenluo@gmail.com.).
• Wenhan Chao is the corresponding author.
• The work was done when Hai Ye interned in Beihang University.

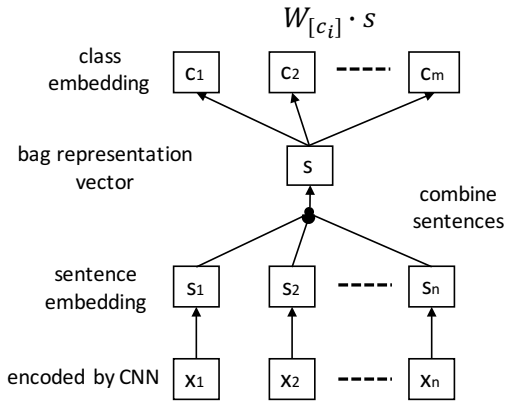

Fig. 1. The main architecture of our model.

by [7][8], we add a regularization term to the loss functions to better learn the connections between relations, and we only regularize the positive relations ignoring relation of NR (does not express any relation) based on the assumption that the connections between relations are only in positive relations not in NR.

Besides, class imbalance is another severe problem which can not be ignored for distant relation extraction. We find that around 70% training data express NR relation type and even more than 90% in test set, so samples with NR counts a much higher proportion comparing to positive samples. This problem will severely affect the model training, causing the model hard to classify positive samples from NR [9]. To relieve this problem, based on the ranking loss functions, we further adopt cost-sensitive learning to rescale the costs from the positive labels and negative ones by increasing losses for positive labels and decreasing losses for NR.

Furthermore, combining information across sentences will be more appropriate for joint extraction which provides more information from other sentences to extract each relation ([10], [11]). In Table I, sentence #1 is the evidence for *place_of_birth*, but it also expresses the meaning of "living in someplace", so it can be aggregated with sentence #2 to extract *place_lived*. Meanwhile, the word of "hometown" in sentence #2 can provide evidence for *place_of_birth* which should be combined with sentence #1 to extract *place_of_birth*.

In this work, we propose a unified model that integrates ranking based cost-sensitive multi-label learning with CNN to exploit class ties between relations and further relieve class imbalance problem. Inspired by the effectiveness of deep learning for modeling sentences [12], we use CNN to encode sentences. Similar to [11][13], we use class embeddings to represent relation classes. The whole model architecture is presented in Figure 1. We first use CNN to embed sentences, then we introduce two variant methods to combine the embedded sentences into one bag representation vector aiming to aggregate information across sentences, after that we measure the similarity between bag representation and relation class in real-valued space. Finally, we use the ranking loss functions to make joint extraction.

Our experimental results on dataset of [14] are evident that: (1) Our model is much more effective than the baselines; (2) Leveraging class ties will enhance relation extraction and our model is efficient to learn class ties by joint extraction; (3) A much better model can be trained after relieving class imbalance from NR.

Our contributions in this paper can be encapsulated as follows:

- We propose to leverage class ties to enhance relation extraction. Combined with CNN, an effective deep ranking based multi-label learning model with regularization technique is introduced to exploit class ties.
- We adopt the cost-sensitive learning to relieve the class imbalance problem and experimental results show the effectiveness of our method.
- Our method achieves state-of-the-art performance.

This paper is the extension of [15]. Compared to original work in [15] whose codes can be obtained from https://github.com/oceanypt/DR_RE, this paper has several differences:

Methods: (a) We further fully consider the class imbalance problem. We propose a novel ranking based cost-sensitive loss function combined with multi-label learning. (b) To better learn class ties between relations, we further introduce a regularization term to ranking loss functions.

Experiments: (a) We further do experiments to analyze the effectiveness of our novel cost-sensitive ranking loss functions. (b) The evaluation experiments on the effectiveness of regularization have further be conducted.

Content: (a) We rewrite the description of our methods from the view of multi-label learning and cost-sensitive learning to gain more theoretical justification improvement.

II. Related Work

A. Relation Extraction

Previous methods on relation extraction mainly can be summarized as supervision based and distant supervision based. Supervision based methods will need much labeled data to work well which can not keep up with the rapid growth of volume of relation types. To overcome the problem of data sparsity for supervision based methods, distant supervision relation extraction has been proposed by [1]. However, DS based relation extraction suffers the two problems of *wrong labelling problem* and *overlapping problem*, in which the former means that sentences containing certain entities actually do not express the relation type of the entities or even do not express any relations and the latter indicates that one entity tuple may have multiple relation types. To solve the problem of wrong labelling, [14] introduce multi-instance learning for relation extraction in which the mentions of one certain entity tuple are regarded as one bag and make the model to extract relations on mention bags, however this method can not deal with the relation overlapping problem. Afterwards, [3] and [4] introduce the framework of multi-instance multi-label learning to jointly consider the two problems and improve the performance significantly. Though they also propose to make joint extraction of relations, they only use information from single sentence losing information from other sentences.

[16] try to use *Markov logic* model to capture consistency between relation labels, on the contrary, our model leverages deep ranking to learn class ties automatically.

Recent years, deep learning has achieved remarkable success in computer vision and natural language processing [12]. In supervision relation extraction, deep learning has been applied to automatically learn the features of sentences ([11], [13], [17], [18]). [17] apply convolutional neural networks to model sentences and import position feature for RE, which obtains significant gains in RE performance. Afterwards, [11], [13], [18] further introduce more advanced deep learning models for RE. In distant supervison relation extraction, [19] propose a piecewise convolutional neural network with multi-instance learning for DS based relation extraction, which improves the precision and recall significantly. Afterwards, [11] introduce the mechanism of attention ([20], [21]) to select the sentences to relieve the wrong labelling problem and use all the information across sentences. [22] further propose a multi-lingual neural relation extraction framework considering the information consistency and complementarity among cross-lingual texts. However, the two deep learning based models only make separated extraction thus can not model class ties between relations. Recently, [23] propose to incorporate relation paths for distant supervision relation extraction and [24] introduce to use the description of entities to enhance distant supervision relation extraction. [25] proposes to model the noise caused by wrong labelling problem and show that dynamic transition matrix can effectively characterize the noises.

B. Deep Learning to Rank, Cost-sensitive Learning

Learning to rank (LTR) is an important technique in information retrieval (IR) [26]. The methods to train a LTR model includes pointwise, pairwise and listwise. We apply pairwise LTR in our paper. Deep learning to rank has been widely used in many problems to serve as a classification model. In image retrieval, [27] apply deep semantic ranking for multi-label image retrieval. In text matching, [28] adopt learning to rank combined with deep CNN for short text pairs matching. In traditional supervised relation extraction, [13] design a pairwise loss function based on CNN for single label relation extraction. Based on the advantage of deep learning to rank, we propose pairwise learning to rank (LTR) [26] combined with CNN in our model aiming to jointly extract multiple relations.

Cost-sensitive learning is one of the techniques for class imbalance problem, which assigns higher wrong classification costs to classes with small proportion. For example, [29] propose a regularized softmax to deal with the imbalanced edge label classification. [30] adopt cost-sensitive learning to learn deep feature representations from imbalanced data. The another approach to relieve class imbalance problem is re-sampling [31], [32] including over-sampling and under-sampling, which aims to balance the distributions of data in different labels.

III. Methodology

In this section, we will introduce our model methods. Firstly, we give out the description of the widely used CNN architecture for encoding sentences. Then we will discuss our ranking based multi-label learning framework with regularization technique. After that, we will introduce the cost-sensitive learning to relieve the impact of NR.

A. Notation

We define the relation classes as $\mathcal{L} = \{1, 2, \cdots, C\}$, entity tuples as $\mathcal{T} = \{t_i\}_{i=1}^{M}$ and mentions[1] as $\mathcal{X} = \{x_i\}_{i=1}^{N}$. Dataset is constructed as follows: for entity tuple $t_i \in \mathcal{T}$ and its relation class set $L_i \subseteq \mathcal{L}$, we collect all the mentions X_i that contain t_i, the dataset we use is $\mathcal{D} = \{(t_i, L_i, X_i)\}_{i=1}^{H}$. Given a data $(t_k, L_k, X_k) \in \{(t_i, L_i, X_i)\}_{i=1}^{H}$, the sentence embeddings of X_k encoded by CNN are defined as $S_k = \{s_i\}_{i=1}^{|X_k|}$ and we use class embeddings $W \in R^{|\mathcal{L}| \times d}$ to represent the relation classes.

B. CNN for sentence embedding

We take the effective piecewise CNN architecture adopted from [11], [19] to encode sentence and we will briefly introduce PCNN in this section. More details of PCNN can be obtained from previous work.

1) Words Representations:

• **Word Embedding** Given a word embedding matrix $V \in \mathbb{R}^{l^w \times d^1}$ where l^w is the size of word dictionary and d^1 is the dimension of word embedding, the words of a mention $x = \{w_1, w_2, \cdots, w_n\}$ will be represented by real-valued vectors from V.

• **Position Embedding** The position embedding of a word measures the distance from the word to entities in a mention. We add position embeddings into words representations by appending position embedding to word embedding for every word. Given a position embedding matrix $P \in \mathbb{R}^{l^p \times d^2}$ where l^p is the number of distances and d^2 is the dimension of position embeddings, the dimension of words representations becomes $d^w = d^1 + d^2 \times 2$.

2) Convolution, Piecewise max-pooling:

After transforming words in x to real-valued vectors, we get the sentence $q \in \mathbb{R}^{n \times d^w}$. The set of kernels K is $\{K_i\}_{i=1}^{d^s}$ where d^s is the number of kernels. Define the window size as d^{win} and given one kernel $K_k \in \mathbb{R}^{d^{win} \times d^w}$, the convolution operation is defined as follows:

$$m_{[i]} = q_{[i:i+d^{win}-1]} \odot K_k + b_{[k]} \tag{1}$$

where m is the vector after conducting convolution along q for $n - d^{win} + 1$ times and $b \in \mathbb{R}^{d^s}$ is the bias vector. For these vectors whose indexes out of range of $[1, n]$, we replace them with zero vectors.

By piecewise max-pooling, when pooling, the sentence is divided into three parts: $m_{[p_0:p_1]}$, $m_{[p_1:p_2]}$ and $m_{[p_2:p_3]}$ (p_1 and p_2 are the positions of entities, p_0 is the beginning of sentence

[1]The sentence containing one certain entity is called mention.

and p_3 is the end of sentence). This piecewise max-pooling is defined as follows:

$$z_{[j]} = max(m_{[p_{j-1}:p_j]}) \tag{2}$$

where $z \in \mathbb{R}^3$ is the result of mention x processed by kernel K_k; $1 \leq j \leq 3$. Given the set of kernels K, following the above steps, the mention x can be embedded to o where $o \in R^{d^s * 3}$.

3) Non-Linear Layer, Regularization: To learn high-level features of mentions, we apply a non-linear layer after pooling layer. After that, a dropout layer is applied to prevent overfitting. We define the final fixed sentence representation as $r \in \mathbb{R}^{d^f}$ $(d^f = d^s * 3)$.

$$s = g(o) \circ h \tag{3}$$

where $g(\cdot)$ is a non-linear function and we use $tanh(\cdot)$ in this paper; h is a Bernoulli random vector with probability p to be 1.

C. Combine Information across Sentences

We propose two options to combine sentences to provide enough information for multi-label learning.

• **AVE** The first option is average method. This method regards all the sentences equally and directly average the values in all dimensions of sentence embedding. This **AVE** function is defined as follows:

$$r = \frac{1}{n} \sum_{s_i \in S_k} s_i \tag{4}$$

where n is the number of sentences and r is the bag representation combining all sentence embeddings. Because it weights the importance of sentences equally, this method may bring much noise data from two aspects: (1) the wrong labelling data; (2) irrelated mentions for one relation class, for all sentences containing the same entity tuple being combined together to construct the bag representation.

• **ATT** The second one is a sentence-level attention algorithm used by [11] to measure the importance of sentences aiming to relieve the wrong labelling problem. For every sentence, **ATT** will calculate a weight by comparing the sentence to one relation. We first calculate the similarity between one sentence embedding and relation class as follows:

$$e_j = a \cdot W_{[c]} \cdot s_j \tag{5}$$

where e_j is the similarity between sentence embedding s_j and relation class c and a is a bias factor. In this paper, we set a as 0.5. Then we apply Softmax to rescale $\mathbf{e}$ ($\mathbf{e} = \{e_i\}_{i=1}^{|X_k|}$) to $[0, 1]$. We get the weight α_j for s_j as follows:

$$\alpha_j = \frac{\exp(e_j)}{\sum_{e_i \in \mathbf{e}} \exp(e_i)} \tag{6}$$

so the function to merge r with **ATT** is as follows:

$$r = \sum_{i=1}^{|X_k|} \alpha_i \cdot s_i \tag{7}$$

D. Learning Class Ties via Ranking based Multi-label Learning with Regularization

Firstly, we have to present the score function to measure the similarity between bag representation r and relation c.

• **Score Function** We use dot function to produce score for r to be predicted as relation c. The score function is as follows:

$$\mathcal{F}(r, c) = W_{[c]} \cdot r \tag{8}$$

There are other options for score function. In [33], they propose a margin based loss function that measures the similarity between r and $W_{[c]}$ by distance. Because score function is not an important issue in our model, we adopt dot function, also used by [13] and [11], as our score function.

Now we start to introduce the ranking loss functions.

Pairwise ranking aims to learn the score function $\mathcal{F}(r, c)$ that ranks positive classes higher than negative ones. This goal can be summarized as follows:

$$\forall c^+ \in L_k, \forall c^- \in \mathcal{L} - L_k : \mathcal{F}(r, c^+) > \mathcal{F}(r, c^-) + \beta \tag{9}$$

where β is a margin factor which controls the minimum margin between the positive scores and negative scores. Inspired by [13], given c^+ and c^-, we adopt the following function to learn the score function:

$$\begin{aligned} \mathcal{H}(c^+, c^-, r) = &\ln(1 + \exp(\rho[0, \sigma^+ - \mathcal{F}(r, c^+)])) \\ &+ \ln(1 + \exp(\rho[0, \sigma^- + \mathcal{F}(r, c^-)])) \end{aligned} \tag{10}$$

where $[0, \cdot] = \max(0, \cdot)$, ρ is the rescale factor, σ^+ is positive margin and σ^- is negative margin. This loss function is designed to rank positive classes higher than negative ones controlled by the margin of $\sigma^+ - \sigma^-$. In reality, $\mathcal{F}(r, c^+)$ will be higher than σ^+ and $\mathcal{F}(r, c^-)$ will be lower than σ^-. In our work, we set ρ as 2, σ^+ as 2.5 and σ^- as 0.5 adopted from [13]. To simplify the loss functions given in the followings, we use $\rho[0, \sigma^+ - \mathcal{F}(r, c^+)]$ to replace the first term in $\mathcal{H}$ and use $\rho[0, \sigma^- + \mathcal{F}(r, c^-)]$ to replace the second term.

To learn class ties between relations, we extend the formula (10) to make multi-label learning. Followings are the proposed ranking based loss functions:

• **with AVE (Variant-1)** We define the margin-based loss function with option of **AVE** to aggregate sentences as follows:

$$\begin{aligned} G_{[\text{ave}]} = &\sum_{c^+ \in L_k} \rho[0, \sigma^+ - \mathcal{F}(r, c^+)] \\ &+ \rho |L_k| [0, \sigma^- + \mathcal{F}(r, c^-)] \end{aligned} \tag{11}$$

Similar to [34] and [13], we update one negative class at every training round but to balance the loss between positive classes and negative ones, we multiply $|L_k|$ before the right term in function (11) to expand the negative loss. We apply mini-batch based stochastic gradient descent (SGD) to minimize the loss function. The negative class is chosen as the one with highest score among all negative classes [13], i.e.:

$$c^- = \underset{c \in \mathcal{L} - L_k}{\operatorname{argmax}} \mathcal{F}(r, c) \tag{12}$$

TABLE II
THE PROPORTIONS OF NR SAMPLES FROM SEMEVAL-2010 TASK 8 DATASET AND RIEDEL DATASET.

Pro. (%)	Training	Test
SemE.	17.63	16.71
Riedel	72.52	96.26

• **with ATT (Variant-2)** Now we define the loss function for the option of **ATT** to combine sentences as follows:

$$G_{[\text{att}]} = \sum_{c^+ \in L_k} \{\rho[0, \sigma^+ - \mathcal{F}(r^{c^+}, c^+)] + \rho[0, \sigma^- + \mathcal{F}(r^{c^+}, c^-)]\} \quad (13)$$

where r^c means the attention weighted representation r where attention weights are merged by comparing sentence embeddings with relation class c and c^- is chosen by the following function:

$$c^- = \underset{c \in \mathcal{L} - L_k}{\operatorname{argmax}} \mathcal{F}(r^{c^+}, c) \quad (14)$$

which means we update one negative class in every training round. We keep the values of ρ, σ^+ and σ^- same as values in function (11).

According to this loss function, we can see that: for each class $c^+ \in L_k$, it will capture the most related information from sentences to merge r^{c^+}, then rank $\mathcal{F}(r^{c^+}, c^+)$ higher than all negative scores which each is $\mathcal{F}(r^{c^+}, c^-)$ ($c^- \in \mathcal{L} - L_k$). We use the same update algorithm to minimize this loss.

• **Regularization** To learn the class ties between relations, we have proposed the ranking based loss functions above. Inspired by [7][8], we further capture the relation connections by adding an extra regularization term to the loss functions. We only consider the connections between positive labels ignoring NR. The connections can be measured by the followings:

$$W_{ave} = \frac{1}{T} \sum_{c \in \mathcal{L} - c_{\text{NR}}} W_{[c]} \quad (15)$$

where W_{ave} is the average of the positive classes and $T = |\mathcal{L} - c_{\text{NR}}|$. The regularization term can be written as:

$$\Theta(W) = \epsilon \|W_{ave}\|_2 + \eta \frac{1}{T} \sum_{c \in \mathcal{L} - c_{\text{NR}}} \|W_{[c]}\|_2 \quad (16)$$

In this regularization term, the first term is to learn the class ties between relations and the second term is to control the model complexity. η and ϵ are the parameters. In this paper, we set η as 0.001 and ϵ is set as 10^{-6}.

E. Ranking based Cost-sensitive Multi-label Learning

In relation extraction, the dataset will always contain certain negative samples which do not express relations classified as NR (not relation). Table II presents the proportion of NR samples in SemEval-2010 Task 8 dataset[2] [35] and dataset from [14], which shows almost data is about NR in the latter dataset. Data imbalance will severely affect the model training and cause the model only sensitive to classes with high proportion [32], causing a positive sample to be classified as NR. In order to relieve this problem, we adopt cost-sensitive learning to construct the loss function. Based on $G_{[\text{att}]}$, the cost-sensitive loss function which is **Variant-3** is as follows:

$$\begin{aligned} G_{[\text{cost\_att}]} = \sum_{c^* \in L_k} \{ & g(c^*)(\rho[0, \sigma^+ - \mathcal{F}(r^{c^*}, c^*)] \\ & + \rho[0, \sigma^- + \mathcal{F}(r^{c^*}, c^-)] \\ & + \sum_{c^+ \in L_k - c^*} \gamma \rho[0, \sigma^+ - \mathcal{F}(r^{c^*}, c^+)] \\ & + \gamma \mathbf{1}(c^* \neq c_{\text{NR}}) \rho[0, \sigma^- + \mathcal{F}(r^{c^*}, c_{\text{NR}})]\} \end{aligned} \quad (17)$$

where $g(c^*) = \mathbf{1}(c = c_{\text{NR}})\lambda + \mathbf{1}(c \neq c_{\text{NR}})1$; $\mathbf{1}(\cdot)$ is an indicate function. In this loss function, because NR counts a high proportion in the training set, so we add the penalty λ ($\lambda < 1$) to the cost from NR when relation of entity is labeled as NR; if one entity is labeled as positive relation $c* \in L_k$, we will add cost from other positive relation $c^+ \in L_k - c^*$ and at the same time add the extra cost from NR. The default value of γ is 1 and if γ is small enough, this loss function will be similar to loss function (13). We set λ as 0 in this paper. Similar to function (14), we select c^- as follows:

$$c^- = \underset{c \in \mathcal{L} - L_k}{\operatorname{argmax}} \mathcal{F}(r^{c^*}, c) \quad (18)$$

We also add the regularization term $\Theta(W)$ to $G_{[\text{cost\_att}]}$ to better capture the class ties between relations.

We give out the pseudocode of merging $G_{[\text{cost\_att}]}$ in algorithm 1.

Algorithm 1: Ranking based Cost-sensitive Multi-label Learning

input : $\mathcal{L}$, (t_k, L_k, X_k) and S_k;
output: $G_{[\text{cost\_att}]}$;
$G_{[\text{cost\_att}]} \leftarrow 0$;
for $c^* \in L_k$ **do**
  Merge representation r^{c^*} by function (5), (6), (7);
  $G_{[\text{cost\_att}]} \leftarrow g(c^*)(\rho[0, \sigma^+ - \mathcal{F}(r^{c^*}, c^*)])$;
  $c^- \leftarrow \text{argmax}_{c \in \mathcal{L} - L_k} \mathcal{F}(r^{c^*}, c)$;
  $G_{[\text{cost\_att}]} \leftarrow G_{[\text{cost\_att}]} + \rho[0, \sigma^- + \mathcal{F}(r^{c^*}, c^-)]$;
  for $c^+ \in L_k - c^*$ **do**
    $G_{[\text{cost\_att}]} \leftarrow G_{[\text{cost\_att}]} + \gamma\rho[0, \sigma^+ - \mathcal{F}(r^{c^*}, c^+)]$;
  $G_{[\text{cost\_att}]} \leftarrow G_{[\text{cost\_att}]} + \gamma \mathbf{1}(c^* \neq c_{\text{NR}})\rho[0, \sigma^- + \mathcal{F}(r^{c^*}, c_{\text{NR}})]$;
return $G_{[\text{cost\_att}]}$;

IV. EXPERIMENTS

In this section we conduct two sets of experiments, in which the first one is for comparing our method with the baselines and the second one is used to evaluate our model. Without the special statement, we will adhere to the methods and settings mentioned above to conduct the following experiments.

[2]This is a dataset for relation extraction in traditional supervision framework.

TABLE III
HYPER-PARAMETER SETTINGS.

Parameter Name	Symbol	Value
Window size	d^{win}	3
Sentence. emb. dim.	d^f	690
Word. emb. dim.	d^1	50
Position. emb. dim.	d^2	5
Batch size	$\mathcal{B}$	160
Learning rate	μ	0.03
Dropout pos.	p	0.5

A. Dataset and Evaluation Criteria

We conduct our experiments on a widely used dataset, developed by [14] and has been used by [3], [4], [19] and [11]. The dataset aligns Freebase relation facts with the New York Times corpus, in which training mentions are from 2005-2006 corpus and test mentions from 2007. Following [1], we adopt held-out evaluation framework in all experiments. We use all training dataset to train our model and then test the trained model on test dataset to compare the predicted relations to gold relations. Aggregated precision/recall curves are drawn and precision@N (P@N) is reported to illustrate the model performance.

B. Experimental Settings

Word Embeddings. We use a word2vec tool that is gensim[3] to train word embeddings on NYT corpus. Similar to [11], we keep the words that appear more than 100 times to construct word dictionary and use "UNK" to represent the other ones.

Hyper-parameter Settings. Three-fold validation on the training dataset is adopted to tune the parameters following [4]. We use grid search to determine the optimal hyper-parameters. We select word embedding size from $\{50, 100, 150, 200, 250, 300\}$. Batch size is tuned from $\{80, 160, 320, 640\}$. We determine learning rate among $\{0.01, 0.02, 0.03, 0.04\}$. The window size of convolution is tuned from $\{1, 3, 5\}$. We keep other hyper-parameters same as [19]: the number of kernels is 230, position embedding size is 5 and dropout rate is 0.5. Table III shows the detailed parameter settings.

C. Comparisons with Baselines

Baseline. We compare our model with the following baselines:

• **Mintz** [1] is the first original model which incorporates distant supervision for relation extraction.

• **MultiR** [3] is the multi-instance learning based graphical model which aims to address overlapping relation problem.

• **MIML** [4] is a multi-instance multi-label framework which jointly considers the wrong labelling problem and overlapping problem.

• **PCNN+ATT** [11] is the state-of-the-art model in dataset of [14] which applies sentence-level attention to relieve the wrong labelling problem in DS based relation extraction. This model applies piece-wise convolutional neural network [19] to model sentences.

[3]http://radimrehurek.com/gensim/models/word2vec.html

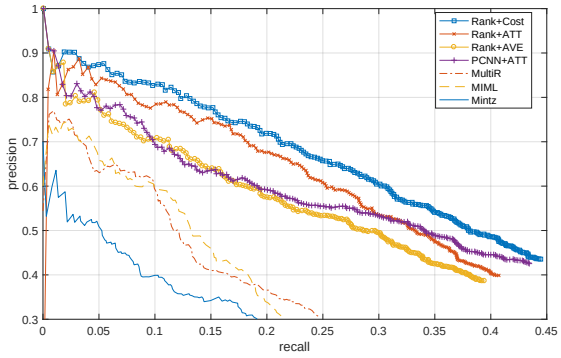

Fig. 2. Performance comparison of our model and the baselines. "Rank+Cost" is using the loss function of $G_{[\text{cost\_att}]}$, "Rank+ATT" is using $G_{[\text{att}]}$ and "Rank+AVE" is using $G_{[\text{ave}]}$.

Results and Discussion. We compare our three variants of loss functions with the baselines and the results are shown in Figure 2. From the results we can see that: (1) Rank+ATT (Variant-1) achieves comparable results with PCNN+ATT; (2) After relieving the impact of NR by adopting cost-sensitive learning, Rank+Cost can significantly outperform PCNN+ATT with much higher precision and slightly higher recall in whole view; (3) Rank+AVE can not keep up with PCNN+ATT because of using average method to merge bag representation leading to incorporate much noise, instead Rank+ATT improves the performance significantly by adopting sentence-level attention.

D. Impact of Class Ties

In this section, we conduct experiments to reveal the effectiveness of our model to learn class ties with three variant loss functions mentioned above, and the impact of class ties for relation extraction. As mentioned above, we adopt two techniques to model the class ties: multi-label learning with ranking based loss functions and regularization term to better model class ties. In the followings, we will conduct experiments to reveal the two aspects for modelling class ties. We will adopt aggregated P/R curves and precisions@N (100, 200, $\cdots$, 500) to show the model performances.

• **Ranking based Loss Function.** The effectiveness of ranking loss functions to learn class ties lies in the joint extraction of relations to conduct multi-label leaning, so to reveal the impact of ranking loss function to learn class ties, we will compare the joint extraction with separated extraction. Regularization term is added to all variant models.

Experimental results are shown in Figure 3 and Table IV. From the results we can see that: (1) For Rank+ATT and Rank+Cost, joint extraction exhibits better performance than separated extraction, which demonstrates class ties will improve relation extraction and the two methods are effective to learn class ties; (2) For Rank+AVE, surprisingly joint extraction does not keep up with separated extraction. For the second phenomenon, the explanation may lie in the **AVE** method to aggregate sentences will incorporate noise data consistent with the finding in [11]. When make joint extraction, we will combine all sentences containing the same entity tuple

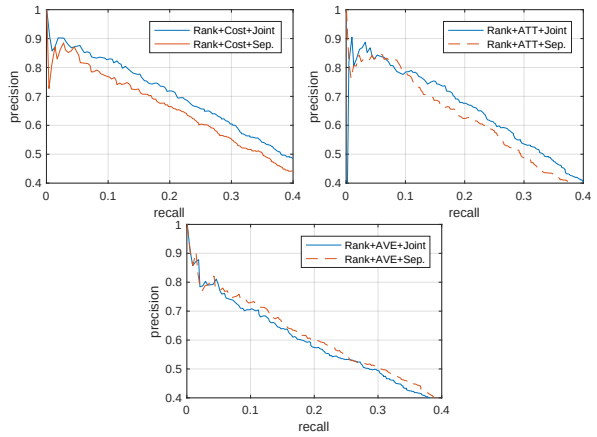

Fig. 3. Results for impact of ranking based loss function with methods of Rank + AVE, Rank + ATT and Rank + Cost

TABLE IV
PRECISIONS FOR TOP 100, 200, 300, 400, 500 AND AVERAGE OF THEM FOR IMPACT OF JOINT EXTRACTION AND CLASS TIES.

P@N(%)	100	200	300	400	500	Ave.
R.+AVE+J.	79.1	73.8	70.4	66.0	63.1	70.5
R.+AVE+S.	**80.2**	**74.9**	**72.2**	**67.8**	**64.0**	**71.8**
R.+ATT+J.	**86.8**	80.6	**78.4**	**75.2**	**71.1**	**78.4**
R.+ATT+S.	82.4	**82.7**	75.3	70.1	66.2	75.3
R.+ExATT+J.	**86.8**	**83.2**	**81.1**	**76.7**	**73.5**	**80.3**
R.+ExATT+S.	85.7	78.5	75.6	72.4	69.0	76.3

no matter which class type is expressed, so it will gender much noise if we only combine them equally.

- **Regularization.** To see the impact of regularization technique for modelling class ties, we compare the methods using regularization with the methods without using regularization. All variant models are in joint extraction setting. The results are shown in Figure 4 and Table V. From the results, we can see that after regularizing the class embeddings, the performance of extraction can be further improved in methods of Rank+Cost and Rank+ATT, which demonstrates the effectiveness of regularization to model class ties. Because simply averaging sentence embeddings will bring much noise, so method of Rank+AVE with regularization actually can not improve the model performance so much.

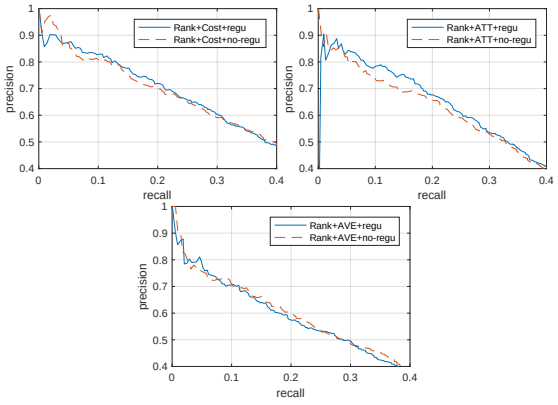

Fig. 4. Results for impact of regularization to model class ties.

TABLE V
PRECISIONS FOR TOP 100, 200, 300, 400, 500 AND AVERAGE OF THEM FOR IMPACT OF REGULARIZATION TO MODEL CLASS TIES

P@N(%)	100	200	300	400	500	Ave.
R.+AVE+no-regu.	78.0	72.3	69.8	**66.5**	**64.0**	70.1
R.+AVE+regu.	**79.1**	**73.8**	**70.4**	66.0	63.1	**70.5**
R.+ATT+no-regu.	84.6	77.5	72.9	69.6	68.0	74.5
R.+ATT+regu.	**86.8**	**80.6**	**78.4**	**75.2**	**71.1**	**78.4**
R.+Cost+no-regu.	85.7	81.7	80.1	75.2	71.3	78.8
R.+Cost+regu.	**86.8**	**83.2**	**81.1**	**76.7**	**73.5**	**80.3**

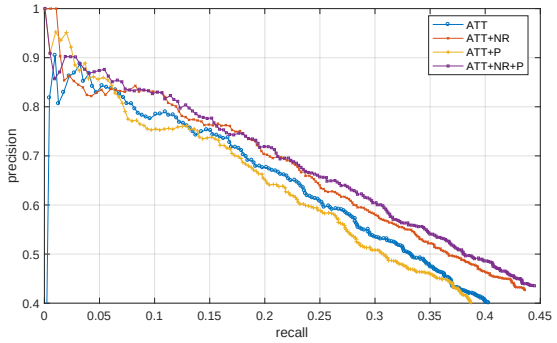

Fig. 5. Results for impact of cost-sensitive learning. "ATT" means the loss function of Variant-2; "ATT+NR" means only considering the cost of NR controlled by λ ignoring the cost controlled by γ based on Variant-2 and λ is set to 0; "ATT+P" means considering the cost controlled by γ based on Variant-2 ignoring the cost of NR and γ is set to 1; "ATT+NR+P" is the loss function of Variant-3 and jointly considers the two kinds of costs mentioned above, λ is set to 0 and γ is 1.

E. Impact of Cost-sensitive Learning

In this section, we conduct experiments to reveal the effectiveness of cost-sensitive learning to relieve the impact of NR for model training and model performance. For the loss function of $G_{[\text{cost\_att}]}$, we have two parts for cost-sensitive learning: the first is the one penalized by γ, and the second is the NR cost penalized by λ. Based on loss function of Variant-3, we respectively relieve the cost controlled by γ and the cost of NR controlled by λ to see the impact of cost-sensitive learning. We will adopt aggregated P/R curves and precisions@N (100, 200, $\cdots$, 500) to show the model performances.

The results are shown in Figure 5 and Table VI. From the results, we can see that considering the cost controlled by γ can sightly improve the performance in low recall range and considering the cost of NR controlled by λ can boost the performance significantly. Considering both of the two kinds

TABLE VI
PRECISIONS FOR TOP 100, 200, 300, 400, 500 AND AVERAGE OF THEM FOR IMPACT OF COST-SENSITIVE LEARNING.

P@N(%)	100	200	300	400	500	Ave.
ATT	86.8	80.6	78.4	75.2	71.1	78.4
ATT+NR	82.4	84.3	80.1	76.2	73.5	79.3
ATT+P	85.7	77.5	75.6	73.7	69.9	76.5
ATT+NR+P	86.8	83.2	81.1	76.7	73.5	80.3

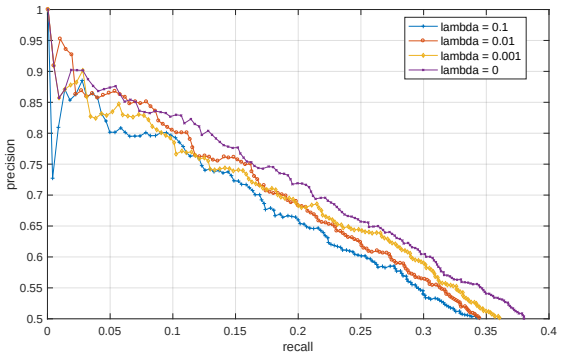

Fig. 6. Effect of λ for model performance based on the loss function of Variant-3.

TABLE VII
PRECISIONS FOR TOP 100, 200, 300, 400, 500 AND AVERAGE OF THEM FOR IMPACT OF COST-SENSITIVE LEARNING.

P@N(%)	100	200	300	400	500	Ave.
$\lambda = 0$	86.8	83.2	81.1	76.7	73.5	80.3
$\lambda = 0.001$	82.4	82.2	77.0	73.9	71.1	77.3
$\lambda = 0.01$	85.7	84.3	77.7	75.7	70.5	78.8
$\lambda = 0.1$	85.7	80.1	76.3	73.1	68.6	76.8

of cost can achieve the best performance. From these results, we can see that relieving impact NR is really important to improve the performance of extraction.

F. Impact of NR

From the discussion above, we can know that NR can have much significant impact for model performance, so in this section, we conduct more experiments to reveal the impact of NR cost controlled by λ for model performance.

• **Effect of λ Penalty.** We conduct experiments on the choice of λ. Based on the loss function of Variant-3, we select λ from $\{0, 0.001, 0.01, 0.1\}$ to see how much effect of NR can gender to the performance. We also adopt aggregated P/R curves and precisions@N (100, 200, $\cdots$, 500) to show the model performances. Models are set with joint extraction and regularization. The results are shown in Figure 6 and Table VII. From the results we can find that when λ becomes larger, the model performance will decrease because NR will have more negative impact on model performance, so in order to achieve better model performance, the value of λ should be set smaller.

• **Effect of NR for Model Convergence.** Then we further evaluate the impact of NR for convergence behavior of our model in model training. Also with the three variant loss functions, in each iteration, we record the maximal value of F-measure [4] to represent the model performance at current epoch. Models are with setting of joint extraction but no regularization. Model parameters are tuned for 15 times and the convergence curves are shown in Figure 7. From the result, we can find out: "+NR" converges quicker than "-NR" and arrives to the final score at the around 11 or 12 epoch. In general, "-NR" converges more smoothly and will achieve better performance than "+NR" in the end.

[4] $F = 2 * P * R/(P + R)$

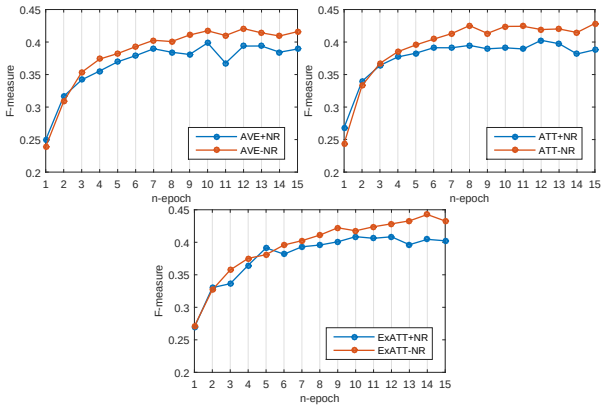

Fig. 7. Impact of NR for model convergence. "+NR" means not relieving NR impact with λ of 1; "-NR" is opposite with λ of 0.

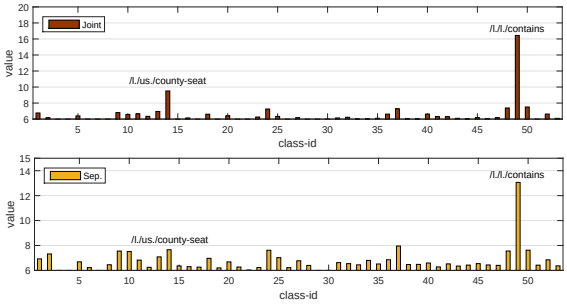

Fig. 8. The output scores for every relation with method of Rank + ATT. The top is under joint extraction setting; the bottom is under separated extraction.

G. Case Study

Joint vs. Sep. Extraction (Class Ties). We randomly select an entity tuple *(Cuyahoga County, Cleveland)* from test set to see its scores for every relation class with the method of Rank + ATT under the setting of relieving impact of NR with joint extraction and separated extraction but no regularization. This entity tuple have two relations: */location/./county_seat* and */location/./contains*, which derive from the same root class and they have weak class ties for they all relating to topic of "location". We rescale the scores by adding value 10. The results are shown in Figure 8, from which we can see that: under joint extraction setting, the two gold relations have the highest scores among the other relations but under separated extraction setting, only */location/./contains* can be distinguished from the negative relations, which demonstrates that joint extraction is better than separated extraction by capturing the class ties between relations.

V. CONCLUSION AND FUTURE WORKS

In this work, we propose a ranking based cost-sensitive multi-label learning for distant relation extraction aiming to leverage class ties to enhance relation extraction and relieving class imbalance problem. We adopt multi-label learning and regularization technic to lean the latent connections between relations, furthermore we introduce a cost-sensitive learning approach to relieve the class imbalance problem. Experimental results show that our method can effectively leverage the

connections to enhance relation extraction and significantly relieve the class imbalance problem.

In the future, we will focus on two aspects: (1) Our method in this paper considers pairwise intersections between labels, so to better exploit class ties, we will extend our method to exploit all other labels' influences on each relation for relation extraction, transferring *second-order* to *high-order* [36]; (2) We will focus on other problems by leveraging class ties between labels, specially on multi-label learning problems [37] such as multi-category text categorization [38] and multi-label image categorization [39].

ACKNOWLEDGMENT

This work was supported by the National High-tech Research and Development Program (863 Program) (No. 2014AA015105) and National Natural Science Foundation of China (No. 61602490).

REFERENCES

[1] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of ACL-IJCNLP*. Association for Computational Linguistics, 2009, pp. 1003–1011.

[2] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of KDD*, 2008, pp. 1247–1250.

[3] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proceedings of ACL-HLT*. Association for Computational Linguistics, 2011, pp. 541–550.

[4] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in *Proceedings of EMNLP*. Association for Computational Linguistics, 2012, pp. 455–465.

[5] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine learning*, vol. 73, no. 2, pp. 133–153, 2008.

[6] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[7] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artificial Intelligence*, vol. 176, no. 1, pp. 2291–2320, 2012.

[8] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 615–637, 2005.

[9] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[10] H. Zheng, Z. Li, S. Wang, Z. Yan, and J. Zhou, "Aggregating inter-sentence information to enhance relation extraction," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[11] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proceedings of ACL*, vol. 1, 2016, pp. 2124–2133.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] C. N. d. Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," in *Proceeding of ACL*, 2015, pp. 626–634.

[14] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Proceedings of ECML-PKDD*. Springer, 2010, pp. 148–163.

[15] H. Ye, W. Chao, Z. Luo, and Z. Li, "Jointly extracting relations with class ties via effective deep ranking," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017.

[16] X. Han and L. Sun, "Global distant supervision for relation extraction," in *Proceedings of AAAI*, 2016, pp. 2950–2956.

[17] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao *et al.*, "Relation classification via convolutional deep neural network." in *Proceeding of COLING*, 2014, pp. 2335–2344.

[18] M. G. Yu Mo and M. Dredze, "Factor-based compositional embedding models," in *NIPS Workshop on Learning Semantics*, 2014, pp. 95–101.

[19] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proceedings of EMNLP*, 2015, pp. 17–21.

[20] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of EMNLP*, 2015, pp. 1412–1421.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[22] Y. Lin, Z. Liu, and M. Sun, "Neural relation extraction with multi-lingual attention," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 34–43.

[23] W. Zeng, Y. Lin, Z. Liu, and M. Sun, "Incorporating relation paths in neural relation extraction," *arXiv preprint arXiv:1609.07479*, 2016.

[24] G. Ji, K. Liu, S. He, and J. Zhao, "Distant supervision for relation extraction with sentence-level attention and entity descriptions." in *AAAI*, 2017, pp. 3060–3066.

[25] B. Luo, Y. Feng, Z. Wang, Z. Zhu, S. Huang, R. Yan, and D. Zhao, "Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 430–439.

[26] T.-Y. Liu, "Learning to rank for information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[27] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proceedings of CVPR*, 2015, pp. 1556–1564.

[28] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 373–382.

[29] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3982–3991.

[30] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Cost sensitive learning of deep feature representations from imbalanced data," *arXiv preprint arXiv:1508.03422*, 2015.

[31] C. Huang, Y. Li, C. Change Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5375–5384.

[32] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[33] L. Wang, Z. Cao, G. de Melo, and Z. Liu, "Relation classification via multi-level attention cnns," in *Proceedings of ACL, Volume 1: Long Papers*, 2016.

[34] J. Weston, S. Bengio, and N. Usunier, "WSABIE: scaling up to large vocabulary image annotation," in *Proceedings of IJCAI*, 2011, pp. 2764–2770.

[35] K. Erk and C. Strapparava, Eds., *Proceedings of SemEval*. The Association for Computer Linguistics, 2010.

[36] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[37] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artificial Intelligence*, vol. 176, no. 1, pp. 2291–2320, 2012.

[38] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, "Learning hierarchical multi-category text classification models," in *Proceeding of ICML*. ACM, 2005, pp. 744–751.

[39] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, and Z. Wang, "Joint multi-label multi-instance learning for image classification," in *CVPR*. IEEE, 2008, pp. 1–8.

Hai Ye received his Bachelor Degree from the Department of Computer Science and Engineering of Beihang University in 2017. His research interest lies in natural language processing and machine learning.

Zhunchen Luo now is working as a software engineer in China Defense Science and Technology Information Center. He received his Phd degree from National University of Defense Technology in 2013 and he was a full visiting student in the University of Edinburgh from 2011 to 2013. His research interest lies in natural language processing, information retrieval and social media.

Wenhan Chao now is the lecture in the Department of Computer Science and Engineering of Beihang University. He obtained his Phd degree from National University of Defense Technology in 2008. His research interest lies in machine translation and data mining.