Distant Supervised Relation Extraction with Cost-Sensitive Loss

Daojian Zeng[1, 2], Yao Xiao[1, 2], Jin Wang[2, *], Yuan Dai[1, 2] and Arun Kumar Sangaiah[3]

Abstract: Recently, many researchers have concentrated on distant supervision relation extraction (DSRE). DSRE has solved the problem of the lack of data for supervised learning, however, the data automatically labeled by DSRE has a serious problem, which is class imbalance. The data from the majority class obviously dominates the dataset, in this case, most neural network classifiers will have a strong bias towards the majority class, so they cannot correctly classify the minority class. Studies have shown that the degree of separability between classes greatly determines the performance of imbalanced data. Therefore, in this paper we propose a novel model, which combines class-to-class separability and cost-sensitive learning to adjust the maximum reachable cost of misclassification, thus improving the performance of imbalanced data sets under distant supervision. Experiments have shown that our method is more effective for DSRE than baseline methods.

Keywords: Relation extraction, distant supervision, class imbalance, class separability, cost-sensitive.

1 Introduction

Relation extraction plays a core role in Natural Language Processing (NLP), and it has always been the focus of many researchers. Supervised methods often achieve good results in relation extraction [Kambhatla (2004); Zhou, Su, Zhang et al. (2005)], But it relies on huge amount of data and entails better separation between classes [Raj, Magg and Wermter (2016)].

To overcome the shortcomings of the lack of labeled training data in the supervised paradigm, distant supervision has been proposed, which can automatically generate training data. Distant supervision can convert massive unstructured data into labeled data by leveraging existing knowledge bases, then the supervised model uses these labeled data to create features [Mintz, Bills, Snow et al. (2009); Hoffmann, Zhang, Ling et al. (2011); Riedel, Yao and McCallum (2010); Surdeanu, Tibshirani, Nallapati et al. (2012)].

[1] Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, 410004, China.

[2] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, 410004, China.

[3] School of Computing Science and Engineering, Vellore Institute of Technology (VIT), Tamil Nadu 637212, India.

[*] Corresponding Author: Jin Wang. Email: jinwang@csust.edu.cn.

CMC. doi:10.32604/cmc.2019.06100 www.techscience.com/cmc

Recently, many researchers combine distant supervision with deep neural network to automatic learn features [Zeng, Liu, Lai et al. (2014); Zeng, Liu, Chen et al. (2015); Lin, Shen, Liu et al. (2016); Jiang, Wang, Li et al. (2016); Zeng, Dai, Li et al. (2018)], which have made a series of progress. Distant supervision assumes that if the sentence in the dataset contains the entity pairs expressing a relation in the knowledge base, then all sentences containing the same entity pairs in the dataset are considered to express this relation. Since this assumption is too absolute, and the data in the real world has its distribution, the method of distant supervision still has the following shortcomings.

First, there is a serious class imbalance problem in the data which automatically labeled by the distant supervision. Secondly, the data automatically generated by distant supervision have a poor class-to-class (C2C) separability.

Currently, there are two mainstream methods to address class imbalance. One is changing the dataset's distribution and another one is adjusting the corresponding algorithm. For the first method, it changes the distribution of data by under-sampling or over-sampling. Specifically, under-sampling removes some instances from the majority class so that the number of samples from the majority class and minority class is close. Since many instances are discarded, the training set is smaller than the original one, it is possible to cause under-fitting. Over-sampling is the opposite of under-sampling. These methods add some instances into minority class to fill the quantity gap of imbalanced classes and then learn. Due to the repeated sampling from minority class, it is prone over-fitting. In this paper, we propose a new algorithm to address the problem of class imbalance. In order to reduce the negative influence of the artificial class noise in distant supervision, we use the ranking loss function. In the conventional ranking loss function, because the conventional cost function will treat all individual errors as equal importance, the classifier tends to classify all instances into the majority class [Murphey, Guo and Feldkamp (2004)]. To avoid this kind of situation, we use cost-sensitive ranking loss function. When misclassifying the instance from minority class, we will give it more punishment than misclassifying a majority instance, so this method is more beneficial to correctly classify the minority class.

Using cost-sensitive ranking loss achieves excellent results in most cases. However, due to some classes have poor C2C separability in the automatically labeled data, these classes cannot be correctly classified. In this case, we use the Silhouette score [Rousseeuw (1987)] as the C2C separability measure, and then adjust the cost of misclassification. Specifically, when the C2C separability is good which means it's easier to correctly classify at this time, so the error should cost more, and vice versa. Generally, researchers will set the cost of misclassification based on the distribution of the data, and the cost remained unchanged during the training. Different from their works, by considering C2C separability, we can adjust the maximum reachable cost of misclassification, so that the cost of misclassification can be automatically learned based on the final problem, thus, our method is more flexible.

2 Related work

Relation extraction automatically identify the semantic relation between entities, which is a very important task in NLP. Generally supervised learning methods yield high performance [Mooney and Bunescu (2006); Zelenko, Aone and Richardella (2003); Zhou, Su, Zhang et al. (2005)]. But supervised relation extraction often faces the challenge of a lack of labeled training data. Mintz et al. [Mintz, Bills, Snow et al. (2009)] uses the freebase, a prevalent knowledge base, to align with rich unstructured data for distant supervision, so that a large amount of labeled data can be obtained. But distant supervision has a problem of the wrong label. In order to address this problem, a relaxed distant supervision assumption was proposed by Riedel et al. [Riedel, Yao and McCallum (2010); Hoffmann, Zhang, Ling et al. (2011); Surdeanu, Tibshirani, Nallapati et al. (2012)] for multi-instance learning. Nguyen et al. [Nguyen and Moschitti (2011)] extends distant supervision by using relations in Wikipedia.

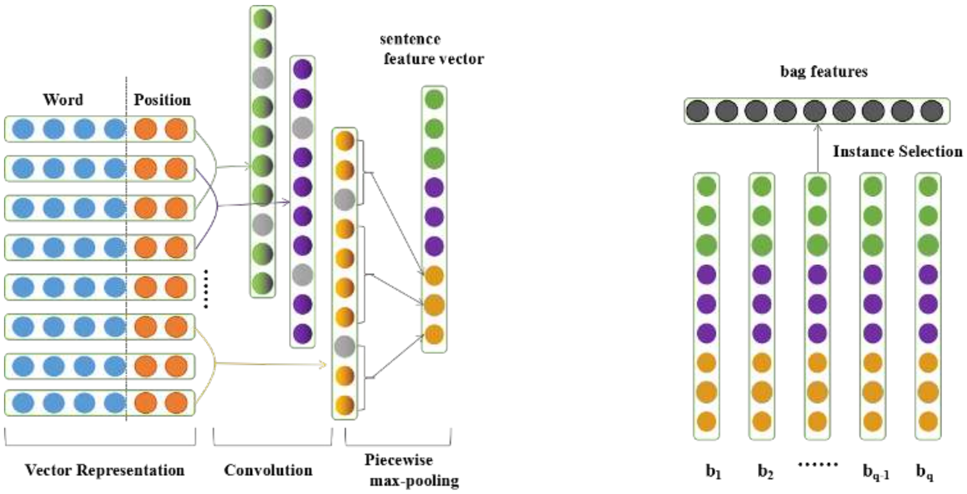
The above methods are effective for DSRE. But they need high quality handcrafted features. Recently, many researchers have attempted to use neural networks for DSRE rather than hand-crafted features. Zeng et al. [Zeng, Liu, Lai et al. (2014)] adopts CNNs to extract sentence-level features and lexical-level features to make full use of the semantic information of sentences. Santos et al. [Santos, Xiang and Zhou (2015)] proposes the pairwise ranking loss function to alleviate the impact of artificial classes. These methods use sentence-level annotated data to train the classifier. However, since a fact may correspond to multiple sentences during data generation, just like data collection of indoor localization [Li, Chen, Gao et al. (2018)], these methods cannot be applied directly in DSRE. Therefore, Zeng et al. [Zeng, Liu, Chen et al. (2015)] proposes piecewise convolutional neural network (PCNN) model, and incorporates multi-instance learning to solve the above problem. Lin et al. [Lin, Shen, Liu et al. (2016)] makes use of the attention mechanism to minimize the negative impact of the wrong label in the DSRE. Jiang et al. [Jiang, Wang, Li et al. (2016)] uses the cross-sentence max-pooling to share information from different sentences. Zeng et al. [Zeng, Zeng and Dai (2017)] Combines ranking loss and cost sensitive to solve the class imbalance problem in DSRE and reduce the impact of the artificial class.

The above work has greatly promoted the relation extraction task. However, the works [Zeng, Liu, Lai et al. (2014); Zeng, Liu, Chen et al. (2015); Jiang, Wang, Li et al. (2016); Lin, Shen, Liu et al. (2016)] don’t pay attention to the class imbalance problem. In Zeng et al. [Zeng, Zeng and Dai (2017)], a new cost-sensitive loss function is proposed, which replaces the traditional cross-entropy loss, but their costs are predefined and fixed during the training process. When the C2C separability is not good, it does not achieve the expected experimental results. Different from these works, we let the C2C separability as one of the factors that affect the cost of misclassification, and let the cost parameter as an automatically learnable parameter. Our method automatically learns cost parameters based on the final problem, so its relatively more flexible .

3 Methods

The structure of our model is similar to Zeng et al. [Zeng, Zeng and Dai (2017)], which is made up of two parts: the PCNNs feature extractor and the ranking based classifier. As shown in Fig. 1, the PCNNs feature extractor adopts a piecewise max pooling to extract the feature vectors of an instance in a bag. After that, in order to get the most appropriate instance and predict the relation of the instance from the bag, we use the rank-based classifier as shown in Fig. 1(b).

In PCNNs feature extractor, we combine the word embedding which is often used in NLP [Xiang, Yu, Yang et al. (2018)] and position embedding as vector representations. Here we denote the word embedding as *E* and the position features as *PFs*. First, we initialize each word token with its corresponding pre-trained word embedding, and next, we train the word embedding by adopting the method in Mikolov et al. [Mikolov, Chen, Corrado et al. (2013)]. After that, we use the method in Zeng et al. [Zeng, Liu, Lai et al. (2014); Zeng, Liu, Chen et al. (2015)] to obtain the positional features of each word token, and we also transform them into vectors. Finally, after convolution and piecewise max-poling, we can obtain the feature vector of the sentence.



(a) PCNNs Feature Extractor (b) Ranking Based Classifier

Figure 1: The architecture used in this work

Each feature vector of instance is denoted as b. Then, we fed it to the ranking based classifier. The network uses the dot product to calculate the score of the class label t_i:

$$s_{t_i} = w_{t_i} b \tag{1}$$

where w_{t_i} is the embedding of class label t_i. After calculating the score of each instance, we select the instance with highest score in the bag and use the corresponding label as the bag label.

3.1 Cost-sensitive ranking loss

In order to reduce the impact of artificial class and compare more conveniently with baseline methods, we use cost-sensitive ranking loss. Suppose that our training set are composed by *N* bags, the *i-th* bag is represented as B_i, and its label is relation r_i. When the *i-th* bag whose label is $r_i = t_j$ is fed into the network, using Eq. (1), we can get the classification score of the current bag label $s^i_{t_j}$ for class t_j, and the highest score of the negative class in the current bag $s_{t_k}^i$ for class t_k. The cost-sensitive ranking loss is given by:

$$L=\sum_{i=1}^{N}\{\log(1+\exp(\lambda(\mathrm{m}_{\mathrm{t_j}}-\mathrm{s}_{\mathrm{t_j}}^{i})))+\log(1+\exp(\lambda(\mathrm{m}_{\mathrm{t_k}}+\mathrm{s}_{\mathrm{t_k}}^{i})))\} \tag{2}$$

Where t_i represents the class label, and $t_k \neq t_j$ $(j,k \in \{1,\cdots,T\}$, T equals the number of all relation types). λ is a constant term. i indicates the $i-th$ bag is fed into the network. m_{t_j} and m_{t_k} can be obtained by calculate the following Eq. (3), which represents the different margin of the class t_i, that is, a cost sensitive parameter:

$$m_{t_i}=\gamma\times\frac{\log(\#t_i)}{\sum_{j}^{T}\log(\#t_j)} \tag{3}$$

where γ is a constant item, and $\#t_j$ equals the number of samples corresponding to the relation label t_j.

We can observe from Eq. (2) that as the score s_{t_j} increases, the first term on the right side of the equation decreases; and as the score s_{t_k} decreases, the second term on the right decreases. Since the goal of our model is to let the score of correct class t_j greater than m_{t_j} and the score of incorrect class t_k smaller than m_{t_k}. Thus, when misclassify the minor classes, our model give more penalties for it than the major classes.

However, one of the drawbacks of this method is that it cannot find the optimum value for m_{t_i}. Experiments show that if the cost is simply set according to the percentage of the classes in the data distribution, the performance improvement is not obvious [Zeng, Zeng and Dai (2017)], especially when poor separability between classes. Thus, we now let m_{t_i} as an adaptive parameter, and experiment show that comparing to static values, the adaptive cost-sensitive parameter can get better performance. In this paper, we use C2C separability to adjust the maximum reachable cost of misclassification, change the originally fixed cost parameter into an optimizable cost parameter, and update the corresponding cost for different classes.

3.2 $\mathbf{m}_{t_i}$ optimization

We will optimize the weight parameters and m_{t_i} simultaneously during the training process, that is, to keep one parameter constant while minimizing the cost relative to the other parameter [Jiang, Wang, Li et al. (2016)]. In our work, we will optimize m_{t_i} in Eq. (2) as follows:

$$m_{t_i}^{*}=\arg\min F(m_{t_i})\ ;\quad F(m_{t_i})=\| T-m_{t_i}\|^{2} \tag{4}$$

T is expressed by the following Eq. (5). H is the ratio of imbalance, which is the maximum reachable cost of misclassification.

$$T=H*\exp(-\frac{F_1}{2}) \tag{5}$$

Through the above steps, we get a m_{t_i} between 1 and H, which is a learnable parameter during the training process, and we call it adaptable m_{t_i}.

3.3 Class-to-class separability

Since effective learning under imbalanced data depends on the degree of separability between classes [Murphey, Guo and Feldkamp (2004)]. For this, our method is, when classes are well separated, if misclassified, more punishment should be given. Conversely, when the C2C separability is poor, classification is difficult to achieve, errors should cost less.

Silhouette score is often used as a measure of C2C separability. The value of Silhouette ranges from -1 to +1, which indicates how close each data point relative to its own cluster. Particularly, when its value is +1, it means that a point is within its own cluster, correspondingly, -1 indicates that the point is completely in the opposite cluster, and when the point is on the boundary of two clusters, the value of Silhouette is 0. The degree of separability of two clusters is calculated by the sum of the Silhouette scores of all points which in these two clusters. So, for the class t_j and class t_k, the separability can be calculated by Eq. (6):

$$S(i) = \frac{K(i) - J(i)}{\max\{J(i), K(i)\}} \tag{6}$$

where, $K(i) = minimun\, d(i, t_k)$, and $d(i, t_k)$ is the average dissimilarity of object $i$in the class t_j to all objects from class t_k; Similarly, $J(i) = minimun\, d(i, t_j)$, where $d(i, t_j)$ is the average dissimilarity of object i in the class t_j to all other objects in this class (As shown in Fig. 2) .

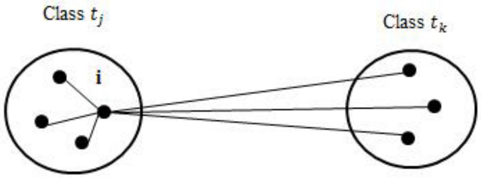


Figure 2: Relation of all elements which included in the computation of $S(i)$

We use S to represent the Silhouette score and give the imbalance ratio as Eq. (7):

$$IR_{t_i} = \gamma \times \frac{\log(\#t_i)}{\sum_{j=1}^{T} \log(\#t_j)} \tag{7}$$

Now, H is defined as the following form:

$$H_{adjusted} = IR_{t_i}(1 + |S|) \tag{8}$$

From Eq. (8) we can observe that if two classes are well separated (in this case $|S| = 1$), the maximum cost at this time can be twice of IR. In this way, we can adjust the maximum reachable cost of misclassification based on separability. The entire optimization process can be seen in the Algorithm 1.

Algorithm 1. Learning optimal parameters

Input: network parameters, training data, cost-sensitive parameters
Output: learned optimal parameters w^* and $m_{t_i}^*$

```
randomly initialize the network parameters,and divide a given training sample into mini-batch
$m_{t_i}^*$ is initialized to 1
while epoch ≠ max-epoch do
    for mini-batch do
        Forward propagation;
            calculate the error by formula (2);
            Calculate the gradient of the error;
            Update
    end
    Calculate the gradient of $m_{t_i}$ by formula (4);
    Update
end
```

4 Experiment

In this section, first, we introduce the dataset and evaluation used in our paper. Then, in order to determine the parameters used in the experiment, we used cross-validation to test several variables. Finally, we show the results of the experiment in charts and analyze them in detail.

4.1 Dataset and evaluation metrics

The dataset[3] used in this paper has been widely used in distant supervision relation extraction, it was developed by Pennington et al. [Pennington, Socher and Manning (2014)] and used by Santos et al. [Santos, Xiang and Zhou (2015); Riedel, Yao and McCallum (2010); Zelenko, Aone and Richardella (2003)]. It generated by aligning the NYT corpus with Freebase. We use corpus from 2005-2006 as the training corpus and corpus from 2007 as the test corpus.

The goal of our methods is to improve the overall precision but not affect the precision of the majority and minority classes. In order to compare with baseline methods and test the performance of our method, we evaluate the models via precision, recall and F1-score.

4.2 Experiment settings

We pretrained skip-gram to generate word embedding. If the entity has multiple word tokens, then we use the ## operator to connect the tokens. We randomly initialized the Position Features to a uniform distribution between [-1,1]. Parameters used in PCNNs model are set as the same as Zeng et al. [Zeng, Zeng and Dai (2017)]. All parameters of our model are in Tab. 1.

Table 1: All parameters used in our experiments.

Parameters	*Value*
Feature maps	$n = 230$
Window size	$w = 3$
Word dimension	$d_w = 50$
Constant term	$\lambda = 2, \gamma = 50$
Position dimension	$d_p = 5$
Adadelta parameter	$\rho = 0.95, \varepsilon = 1e^{-6}$
Mini-batch size	$b_s = 50$

4.3 Baseline

In our baseline methods, there are three methods that use handcrafted features, and the others use convolutional neural networks to extract features. *Mintz* extract features from all sentences, which proposed by Mintz et al. [Mintz, Bills, Snow et al. (2009)]; *MultiR* is proposed by Hoffmann et al. [Hoffmann, Zhang, Ling et al. (2011)], which treats DSRE as a multi-instance learning task; the *MIML* method used in [Surdeanu, Tibshirani, Nallapati et al. (2012)] is a multi-instance and multi-label method for relation extraction; *PCNNs+MIL* is proposed by [Zeng, Liu, Chen et al. (2015)], which extract bag features by using PCNNs and multi-instance learning; *CrossMax* selects features across different instances by incorporating cross-sentence max-pooling and PCNNs, it's proposed by Jiang et al. [Jiang, Wang, Li et al. (2016)]; *R-L* was proposed by [Zeng, Zeng and Dai (2017)], which uses cost sensitivity learning to solve the class imbalance problems.

4.4 Comparison with baseline methods

In this part, we present the results of our experiments in charts, and perform some analysis based on these results. In the following charts, we use ours to represent the method that use C2C separability.

We use the class separability scores *S* to adjust the maximum reachable cost of a misclassification, thereby changing the cost-sensitive ranking loss from a fixed cost into an adaptive cost. The precision/recall curve of the method proposed in this work and baseline methods as shown in Fig. 3. We can observe that our method gain the highest precision at all recall levels, and it can achieve a maximum recall level of approximately 39%. PCNNsMIL can achieve a recall level of 36%, but their precision is too low. R-L can get about 38% recall level, but its precision is lower than the method in this paper. Taking into account the precision and recall at the same time, our method can achieve better results.

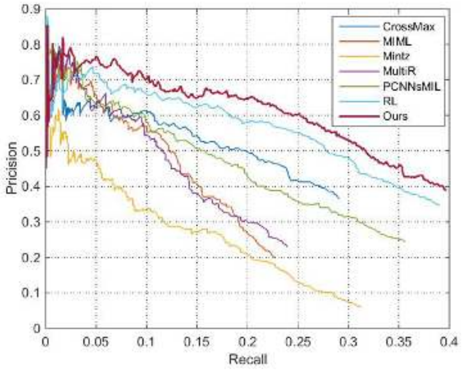


Figure 3: P-R curve for comparison of our method and baseline

4.5 Effect of class separability score on cost-sensitive ranking loss

Since our model degrades to *R-L* after removing the metrics of the C2C separability, in order to verify the impact of C2C separability to cost-sensitive ranking loss, we calculated F1-score for some relations to make a comparison between the R-L baseline with fixed cost and our method with adaptable cost. The results are in Tab. 2.

From Tab. 2, we can see the advantages of incorporating C2C separability metrics. Especially in relation label *people/person/place_lived* and *people/person/place_of_birth*, F1-score is lower when using the *R-L* baseline, because of the poor separability of these two relation classes. In our approach, since the metrics of C2C separability are considered, the classification performance of these two classes is greatly improved. In summary, incorporating class separability metrics to cost-sensitive ranking loss improves the performance effectively.

Table 2: F1-score for some relations to verify the impact of class separability

Relations	*R-L*	*Ours*
/location/location/contains	39.79	**40.20**
/people/person/place_lived	18.05	**21.60**
/people/person/nationality	32.80	**32.87**
/business/person/company	42.14	**42.96**
/people/person/place_of_birth	16.65	**20.72**
/people/deceased/person/place_of_death	**25.47**	25.34
/location/neighborhood/neighborhood_of	34.86	**35.80**
/business/company/founders	35.49	**35.83**

5 Conclusions

We concentrate on the class imbalance problem in DSRE. We use the Silhouette score to measure C2C separability and incorporate this measure to cost-sensitive ranking loss to adjust the maximum applicable cost. Through extensive experiments, the result shows that our method has more significant effect on improving the experimental results. The problem of class imbalance in DSRE can be effectively solved by incorporating the C2C separability measure into the cost-sensitive ranking loss. In future work, we want to further study the impact and difference of other loss functions and cost-sensitive strategies in DSRE.

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

References

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

Jiang, X. T.; Wang, Q.; Li, P.; Wang, B. (2016): Relation extraction with multi-instance multi-label convolutional neural networks. *International Conference on Computational Linguistics*, pp. 1471-1480.

Kambhatla, N. (2004): Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. *Association for Computational Linguistics*, pp. 22.

Lin, Y. K.; Shen, S. Q.; Liu, Z. Y.; Luan, H. B.; Sun, M. S. (2016): Neural relation extraction with selective attention over instances. *Association for Computational Linguistics*, vol. 1, pp. 2124-2133.

Li, W.; Chen, Z.; Gao, X.; Liu, W.; Wang, J. (2018): Multi-model framework for indoor localization under mobile edge computing environment. *IEEE Internet of Things Journal*.

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

Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. (2009): Distant supervision for relation extraction without labeled data. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, vol. 2, pp. 1003-1011.

Mooney, R. J.; Bunescu, R. C. (2006): Subsequence kernels for relation extraction. *International Conference on Neural Information Processing Systems*, vol. 19, pp. 171-178.

Murphey, Y. L.; Guo, H.; Feldkamp, L. A. (2004): Neural learning from unbalanced data. *Applied Intelligence*, vol. 21, no. 2, pp. 117-128.

Nguyen, T. V. T.; Moschitti, A. (2011): End-to-end relation extraction using distant supervision from external semantic repositories. *Association for Computational Linguistics*, vol. 2, pp. 277-282.

Pennington, J.; Socher, R.; Manning, C. (2014): Glove: global vectors for word representation. *Conference on Empirical Methods in Natural Language Processing*, pp. 1532-1543.

Raj, V.; Magg, S.; Wermter, S. (2016): Towards effective classification of imbalanced data with convolutional neural networks. *Artificial Neural Networks in Pattern Recognition*, pp. 150-162.

Riedel, S.; Yao, L.; McCallum, A. (2010): Modeling relations and their mentions without labeled text. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 148-163.

Rousseeuw, P. J. (1987): Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65.

Santos, C. N. D.; Xiang, B.; Zhou, B. (2015): Classifying relations by ranking with convolutional neural networks. *Association for Computational Linguistics*, vol. 1, pp. 132-137.

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

Xiang, L. Y.; Yu, J. M.; Yang, C. F.; Zeng, D. J.; Shen, X. B. (2018): A word-embedding-based steganalysis method for linguistic steganography via synonym substitution. *Institute of Electrical and Electronics Engineers Access*, vol. 6, pp. 64131-64141.

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

Zeng, D. J.; Dai, Y.; Li, F.; Sherratt, R. S.; Wang, J. (2018): Adversarial learning for distant supervised relation extraction. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 121-136.

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

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

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

Zhou, G. D.; Su, J.; Zhang, J.; Zhang, M. (2005): Exploring various knowledge in relation extraction. *Association for Computational Linguistics*, pp. 427-434.