Dynamic Label Correction for Distant Supervision Relation Extraction via Semantic Similarity

Xinyu Zhu[0000−0002−6457−0542], Gongshen Liu✉[0000−0001−5194−1570], Bo Su✉, and Jan Pan Nees

Shanghai Jiao Tong University, Shanghai 200240, China
{jasonzxy,lgshen,subo}@sjtu.edu.cn, JanPanNees@yahoo.com

Abstract. It was found that relation extraction (RE) suffered from the lack of data. A widely used solution is to use distant supervision, but it brings many wrong labeled sentences. Previous work performed bag-level training to reduce the effect of noisy data. However, these methods are suboptimal because they cannot handle the situation where all the sentences in a bag are wrong labeled. The best way to reduce noise is to recognize the wrong labels and correct them. In this paper, we propose a novel model focusing on dynamically correcting wrong labels, which can train models at sentence level and improve the quality of the dataset without reducing its quantity. A semantic similarity module and a new label correction algorithm are designed. We combined semantic similarity and classification probability to evaluate the original label, and correct it if it is wrong. The proposed method works as an additional module that can be applied to any classification models. Experiments show that the proposed method can accurately correct wrong labels, both false positive and false negative, and greatly improve the performance of relation classification comparing to state-of-the-art systems.

Keywords: relation extraction · semantic similarity · label correction.

1 Introduction

Relation extraction (RE) aims to obtain semantic relations between two entities from plain text, such as the following examples: *contains*, *lives in*, *capital of*. It is an important task in natural language processing (NLP), particularly in knowledge graph construction, paragraph understanding and question answering.

Traditional RE suffered from the lack of training data. To solve this problem, distant supervision was proposed [9]. If two entities have a relation in a knowledge base, all the sentences that mention the two entities will be labeled as positive instances. If there is no relation between two entities in the knowledge base, it will be marked as a negative instance (NA).

However, the assumption of distant supervision is too strong and it brought lots of wrong instances. Some examples are shown in Figure 1. *Jake Gyllenhaal* is indeed born in *Los Angeles*, but the sentence in Figure 1 does not express

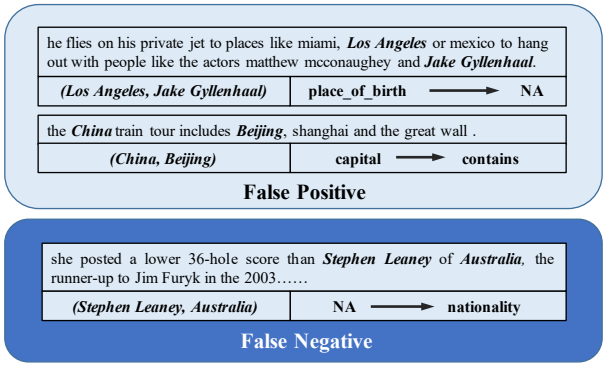

Fig. 1: Some examples of wrong labeled instance in the original training data. False positive instances (above) are caused by the strong assumption of distant supervision. False negative instances (below) are caused by the incomplete knowledge base.

the relation. We call it a false positive instance. There is no related record of *Stephen Leaney* and *Australia* in the knowledge base, but we can infer from the sentence that the nationality of *Stephen Leaney* is *Australia*. We call it a false negative instance.

Previous studies trained data at bag level based on multi-instance learning (MIL) [15,3,19] to deal with the issue of distant supervision. MIL takes the sentences with same entities and relations as a bag so that we can select some good instances for training. Attention mechanism was also applied to help better select instances [6,18,22]. These studies have three limitations: 1) Cannot select correct instances when all the sentences in a bag is wrong labeled. 2) Cannot deal with the false negative instances. 3) Training on bag level cannot make full use of the dataset.

In this study, we propose a novel dynamic label correction model to address the limitations mentioned above. Our method consists of two modules: semantic similarity module and relation classification module. Specifically, each relation has a vector representation, we calculate the similarity between input sentence and relations based on semantic features. We do not rely on the original label, but consider semantic similarity and classification score together to determine a new label for each sentence. The label correction will be performed dynamically during the training process. Without noisy instances, we can train our model at sentence level instead of bag level. The contributions of this paper include:

- A novel method for relation extraction is proposed, which uses semantic similarity and classification probability to dynamically correct wrong labels for each sentence. This enables us to train the classifier at sentence level with correct instances.
- The methods proposed in this paper is model-independent, which means it can be applied to any relation extraction model.

– Experiments show that our methods can greatly improve the performance compared with the state-of-art models.

2 Related Work

Relation Extraction is an important work in NLP. Early methods proposed various features to identify different relations, particularly with supervised methods [4,1,17,12]. The methods mentioned above all suffered from the lack of labeled training data. To solve this problem, Mintz et al. [9] proposed distant supervision, Riedel et al. [15] and Hoffmann et al. [3] improved this method.

Recent years, neural networks were widely used in NLP. Various models were applied in RE task, including convolutional network [20,19,16,10,6,18], recurrent neural networks [21] and long short-term memory network [22]. However, the assumption of distant supervision brought many wrong labeled instances, which cannot be solved by previous models.

Some new methods have been proposed to fight against noisy data. Liu et al. [7] infer true labels according to heterogeneous information, such as knowledge base and domain heuristics. Liu et al. [8] set soft labels heuristically to infer the correct labels and train at entity bag level. Lei et al. [5] use extra information expressed in knowledge graph to help improve the performance on noisy data set. Qin et al. [13] use generative adversarial networks to help recognize wrong data. Qin et al. [14] and Feng et al. [2] both try to remove wrong data using reinforcement learning.

Most of the methods above trained models at bag level, which cannot form a mapping at sentence level. [14] and [2] filter wrong data at sentence level. However, they focus on filtering false positive instances but ignore false negative instances, which cannot make full use of the original dataset. To address these issues, we propose a novel framework to dynamically correct wrong labels for each sentence. Our method can deal with both false positive instances and false negative instances, and train at sentence level. What's more, our method do not use extra information from knowledge graph and can be applied easily.

3 Framework

We propose a novel RE framework, which is able to dynamically correct wrong labels during the training process. Without noisy instances, we can train at sentence level instead of bag level, this will make full use of the dataset and achieve better performance.

The proposed framework mainly consists of two parts: semantic similarity calculator and relation classifier. Figure 2 illustrates how the proposed framework works. Features will be extracted from input sentences first by a sentence encoder, such as a convolutional layer in the figure. Afterwards, semantic similarity will be calculated between feature vectors and relation vectors. Classifier will also calculate the probability for each relation. Since our method is model-independent, it can be applied to any classifier. Finally, we propose a label

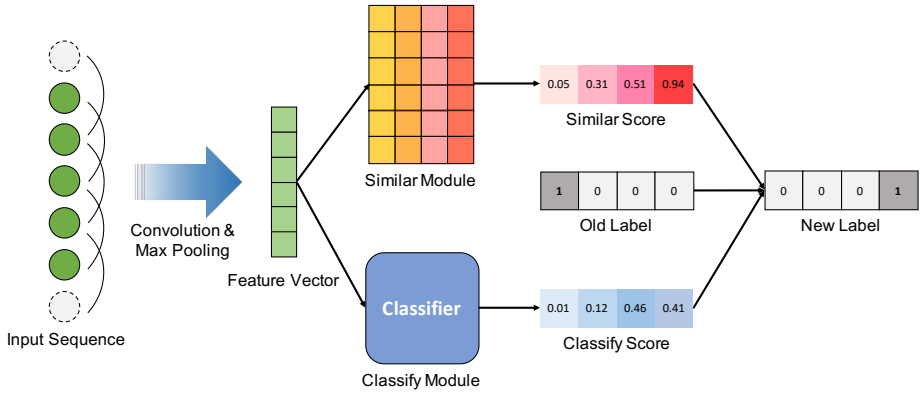

Fig. 2: The architecture of our method for wrong label correction during training process. As an example, according to semantic similarity and original classifier, wrong label is corrected from relation 1 to relation 4.

correction algorithm that combines the information of both parts to give a new correct label for the sentence. In Figure 2, relation 4 and relation 3 achieve the highest scores in semantic similarity and classification probability, respectively. The label is finally corrected from relation 1 to relation 4 according to our algorithm. Each part is described in detail below.

3.1 Input Representation

The input of our model is the sequence of words in a sentence. Similar to previous papers, the input representation consists of word embeddings and position embeddings [20].

Given a sentence of N words $s = \{w_1, w_2, \cdots, w_n\}$, each word is converted to a real-value vector by an embedding matrix $W_e \in \mathbb{R}^{d^w \times |V|}$ where d^w is the word embedding dimension and V is a fixed-sized vocabulary. The position embedding is generated by counting the distance between the current word and two entities. Each position embedding dimension is d^p, so the input representation is formed as $R = \{r_1, r_2, \cdots, r_n\}$, where $r_i \in \mathbb{R}^d$, and $d = d^w + 2 \times d^p$.

3.2 Sentence Encoder

Since convolutional neural networks (CNN) is good at dealing with long sentences, we use CNN to extract features from sentences. Given a input representation R, the convolution operation is applied to R with the sliding window of size k. We define the convolution matrix as $W_c \in \mathbb{R}^{d_c \times (k \times d)}$, where d_c is the number of filters. The output of the i-th convolutional layer can be expressed as:

$$p_i = [W_c q + b_c]_i \tag{1}$$

where $q_i = r_{i-k+1:i} (1 \leq i \leq N + k - 1)$ means the concatenation of k word embeddings. As for the boundary of sentences, $\frac{k-1}{2}$ padding tokens are placed at the beginning and the end of the sentence.

Afterwards, we use piecewise max pooling followed PCNN model [19]. The output of convolution layer can be divided into three segments according to the position of two entities. Max pooling operation will be applied to each segment. The final feature vector we obtained can be expressed as:

$$x = [max(p_{i1}), max(p_{i2}), max(p_{i3})] \tag{2}$$

3.3 Semantic Similarity

We believe that each relation has unique semantic features which can be represented by a vector. We define a relation matrix $W_s \in \mathbb{R}^{h \times r}$, where h is the dimension of convolution layer output and r is the number of relations. Each column of W_s can be viewed as the representation of one relation. In our experiments, W_s is initialized by randomly sampling values from a uniform distribution.

Given a feature vector x, the semantic similarity score can be computed by function $S(x, W_s)$. Cosine function is widely used in text semantic similarity, but it only considers the angular difference between vectors. Inspired by Pearson coefficient [11], we propose a improved cosine function in this paper, which consider the average scores of all relations for the same input. The similarity score of vector x is computed as:

$$S_x = \frac{(W_s - \bar{W}_s)(x - \bar{W}_s)}{||W_s - \bar{W}_s|| \cdot ||x - \bar{W}_s||} \tag{3}$$

Semantic similarity is an important part of dynamic label correction. In order to get the best performance, we hope our network can distinguish different relations to the greatest extent. We design a new loss function to help training:

$$J_s = exp(\gamma(m - S_x^+ + S_x^-)) \tag{4}$$

where m is a margin and γ is a scaling factor. The margin gives extra penalization on the difference in scores and the scaling factor helps to magnifies the scores. S_x^+ refers to the similarity between input vector x and the correct relation vector, and S_x^- refers to the highest score among all the wrong relation vectors. By minimizing this loss function, we hope our model can give scores with a difference greater than m between positive label and negative label.

3.4 Label Correction

In previous works, the feature vector is fed into a fully-connected layer and then softmax layer. The output can be seen as the probability score for relations:

$$C_x = softmax(W_r x + b_r) \tag{5}$$

Algorithm 1 Complete Training Process

1. Train similarity model alone in a small clean training set which is labeled manually;
2. Initialize parameters of classification model with random weights;
3. Pre-train the classification model with original labels at bag level.
4. Train the model with dynamic label correction algorithm at sentence level.

where $W_r \in \mathbb{R}^{h \times r}$ and $b_r \in \mathbb{R}^r$. Previous models calculate loss function by comparing relation prediction with the instance label. Due to wrong labels, the model is optimized to the wrong direction.

Our framework introduces the semantic similarity at this stage to correct wrong labels dynamically in each iteration of training. The semantic similarity and classification probability are combined with different weights. The new relation label for input x is computed as:

$$r_{new} = argmax(\lambda C_x + S_x + \beta S_x * L) \tag{6}$$

where $\lambda = max(S_x)$ can be seen as the confidence of classification score. L is a one-hot vector of the original label and β is a constant which control the effect of old label. Afterwards, we can define the cross-entropy loss function of relation prediction based on the new label:

$$J(\theta) = \sum_{i=1}^{n} logp(r_{new}|x;\theta) \tag{7}$$

where $\theta = \{W_{conv}, b_c, W_r, b_r\}$ is the parameter set.

3.5 Model Training

Semantic similarity is the key of our label correction and needs to be trained to a good level in accuracy. The complete training process is described in Algorithm 1. We first randomly choose a few numbers of instances and manually label them. Semantic similarity module is trained with these correct instances. Afterwards, we pre-train the classification model with original labels at bag level. Finally, wrong labels are corrected dynamically with the help of semantic similarity, and the classification model can be trained with new labels at sentence level.

4 Experiments

In this study, a framework is proposed to dynamically correct wrong labels and train model at sentence level. As our framework is model-independent, the experiments focused on the effect of semantic similarity model, the accuracy of label correction and the performance of relation extraction with new labels.

Table 1: Comparison of results between CNN and our model. Distinct is the average difference between positive and negative relations.

Model	Pre	Rec	F1	Distinct
Zeng[20]	-	-	78.9	-
CNN+CE	79.77	80.61	79.91	29.61
CNN+Cos	79.12	81.26	80.18	38.86
CNN+Sim	**79.20**	**84.49**	**81.76**	**59.93**

4.1 Dataset

The proposed method was evaluated on a widely used dataset developed by Riedel [15]. This dataset is generated by aligning relation facts in Freebase with the New York Times (NYT) corpus. Training set contains sentences of 2005-2006, and test set contains sentences of 2007. There are 522611 sentences, 281270 entity pairs and 18252 relational facts in the training data, and 172448 sentences, 96678 entity pairs and 1950 relation facts in the test data. There are 53 relations including a special relation *NA* which indicates no relation between two entities.

The SemEval-2010 Task 8 dataset was also used to evaluate the semantic similarity module, which is also widely used but without noisy data. The dataset contains 8000 training instances and 2717 test instances. There are 9 different relations and a special relation *Other*. Each relation takes into account the directionality between entities, which means that relation *Product-Producer(e1,e2)* is different from the relation *Product-Producer(e2,e1)*.

4.2 Effect of Semantic Similarity

As mentioned before, we design a semantic similarity matrix to better extract the semantic features. To evaluate the effect of our method, we conducted experiments based on the clean data set SemiEval-2010 and compared with the state-of-the-art baseline proposed by [20].

Zeng in Table 1 reports the result in his paper [20], which uses sentence level features and cross entropy loss function. CNN+CE is the model we reproduced using word embedding of size 300. CNN+Cos is the model with cosine functions and CNN+Sim is the model that applies our similarity functions and loss function. Precision, recall and macro-averaged F1 score are calculated. We hope the feature vectors of different relations have a clear distinction, so we added an indicator named Distinct, which calculate the average difference of vector similarity between positive label and negative labels.

We have the following observations from Table 1: (1) CNN+Sim obtains better performance than CNN+CE in both F1 score and Distinct score, indicating that our method has a positive effect on relation classification. (2) The distinction of feature vectors in CNN+Sim has been improved by over 20 percent compared with CNN+Cos, which demonstrates that the similarity function we designed performs better than traditional cosine function.

Table 2: Parameter Settings

Windows size k	3	Filter number d_c	230	Word dimension d^w	50
Word dimension d^w	50	Position dimension d^p	5	Learning rate	0.001
Dropout Probability	0.5	margin m	1.0	scaling factor γ	2

4.3 Effect of Dynamic Label Correction

The key difference between our model and previous models is that we can dynamically correct both false positive and false negative labels. What's more, without noisy data, our model can perform sentence level classification instead of bag level.

Our analysis indicates that although there are 53 relations in Riedel dataset, most of them are belong to *NA*. Only 10 relations have more than 1000 instances. Considering the negative impact of unbalance in training set, we only performed experiments on these 10 relations.

Baselines. We select the following two state-of-the-art methods for comparison. PCNN+MAX [19] assumes at least one instance in the bag can express the correct relation. It chooses the instance that gets the highest score in each bag. PCNN+ATT [6] adopted selective attention mechanism over instances to reduce the weights of noisy instances. Both PCNN+MAX and CNN+ATT are bag level methods.

Parameter Settings. Our model contains the same parameter settings as the reference papers, in order to present and accurate comparative study. Detailed settings are shown in Table 2. For the dynamic label correction module, we set constant β to 0.7 and the pre-train step to 3300. We set batch size to 50 in PCNN+MAX and 160 in PCNN+ATT.

Precision Recall Curve In our method, we deal with the false positive and false negative instances, which are not labeled correctly in original test data set. The experiments should be performed on a clean data set without wrong labels. We randomly selected 1500 sentences from the original test data and manually labeled relation for each instance. Precision recall (P-R) curve is used to evaluate the model performance.

In our experiments, we randomly selected 50 instances of each relation and manually labeled them. The semantic similarity module was pre-trained to a high accuracy of 95.64% with this clean dataset. For PCNN+ATT, our semantic similarity is combined using attention weights in original method. In the training process, we first conducted contrast experiments at bag level, followed by the sentence level experiment. In the test process, each sentence is treated as a bag so that each method can conduct a sentence level prediction.

As shown in Figure 3, both PCNN+MAX and PCNN+ATT achieve much better performance after applying the dynamic label correction. With the help

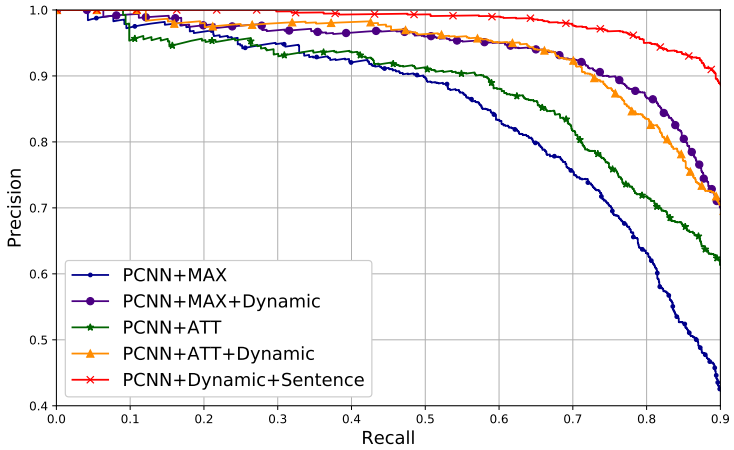

Fig. 3: Precision recall curves of our method and state-of-the-art baselines. PCNN+MAX+Dynamic/PCNN+ATT+Dynamic applied our method to previous models and trained at bag level. PCNN+Dynamic+Sentence used our method and trained at sentence level.

of our method, PCNN+MAX can even outperform the original stronger model PCNN+ATT. When we trained the model at sentence level, it achieves the highest performance. These results indicate that our method can accurately correct the wrong label for each sentence and provide a much cleaner dataset.

NA Evaluation In Riedel dataset, 166003 test sentences belong to *NA* relation, accounting for the majority of test set. However, many of them are false negative instances, which means there is actually a relation between two entities but it is missing in Freebase. To better illustrate the performance of our method, we performed relation prediction for sentences labeled *NA*. We conducted manual evaluation for the top 100, top 200, and top 300 sentences which were predicted to have a certain relation.

Table 3 shows the Top-N prediction accuracy. We can observe that: (1)Many *NA* sentences are indeed false negative instances, which proves the rationality of our method. (2)After applied our method, the accuracy of relation prediction has been greatly improved, at both bag level and sentence level. (3)The model trained at sentence level with our dynamic label correction achieve the highest accuracy.

4.4 Accuracy of Dynamic Label Correction

To illustrate the effect of dynamic label correction, we recorded sentences of corrected labels during each iteration. Subsequently, we selected top 100, top 200, and top 300 sentences for manual verification.

Table 3: Top-N prediction accuracy for sentences labeled NA. +Dynamic(BL) trained the model at bag level, while +Dynamic(SL) trained at sentence level. Results are ranked in descending order according to predict scores.

Model	100	200	300	Avg
PCNN+MAX	56	53.5	52	53.83
+Dynamic(BL)	72	66.5	61.67	66.72
+Dynamic(SL)	**76**	**71**	**64.33**	**70.44**

Table 4: Top-N accuracy of label correction during training process. Sentences are ranked in descending order according to new label score.

Model	100	200	300	Avg
PCNN+MAX	96	93.5	92.33	93.94
PCNN+ATT	97	95	94	95.33

Results in Table 4 contains the following observations: (1) Both models keep the correction accuracy at high level, which shows that our method can make a stable improvement in the training process. (2) The correction accuracy in PCNN+ATT is 97 compared to 96 with PCNN+MAX. As stated in referenced papers, attention mechanism performs better in sentence features extraction, which is also helpful for our correction algorithm. It's reasonable that PCNN+ATT can get a higher accuracy. (3) Due to the high accuracy in label correction, we can obtain a much cleaner dataset. That is why we can train model at sentence level and achieve the better performance.

4.5 Case Study

Table 5 shows some examples of dynamic label correction. Case 1, Case 2 and Case 3 are examples of false positive instances, false negative instances and similar relation instances, respectively. In each case, we present the classification scores calculated by PCNN+MAX model and the final scores after combining our semantic similarity, respectively.

In case 1, the model trained with noisy labels gives relation *place_of_birth* a higher score. After applying our method, relation *NA* achieves the highest score and the wrong label is corrected accurately. In case 2, our method can give the model higher confidence against the wrong labels. In case 3, although *Armenia* does contain *Yerevan*, the sentence further expresses the relation of capital. The original model gives similar scores for these two relations. Our method allows them to have a much clearer distinction.

These cases clearly indicate that our method can deal with both false positive and false negative instances. Similar relations can also be distinguished accurately by our method, such as *capital* and *contains*, *place lived* and *place of birth*.

Table 5: Some examples of dynamic label correction. *Wrong Rel* refers to the wrong labels in original dataset, and *Correct Rel* refers to the new relation corrected by our method.

	Case Study	Wrong Rel	Correct Rel
Case 1	The two pitchers ⋯ are ***Oakland***'s Barry Zito and Florida's ***Dontrelle Willis***.	/person/place_of_birth	NA
Model Score	PCNN+MAX	0.45	0.31
	+Dynamic Label Correction	0.2135	**1.15**
Case 2	⋯ the unpredictable ***Xavier Malisse*** of ***Belgium*** before coming on strong ⋯	NA	/location/contains
Model Score	PCNN+MAX	0.08	0.91
	+Dynamic Label Correction	-0.75	**1.74**
Case 3	Mattheson's scores turned up in ***Yerevan***, the capital of ***Armenia***, and were ⋯	/location/contains	/country/capital
Model Score	PCNN+MAX	0.26	0.38
	+Dynamic Label Correction	0.36	**1.22**

5 Conclusion

In this paper, we propose a novel model to deal with the wrong labels in RE task. Previous work tried to reduce the effect of wrong instances or just remove them. Our method focuses on correcting the wrong labels, which can fundamentally improve the quality of the dataset without reducing its quantity. We introduce the semantic similarity to help dynamically recognize and correct wrong labels during the training process. As a result, we can achieve better performance by training at sentence level instead of bag level. It is worth mentioning that our method is model-independent, which means that our method can work as an additional module and be applied to any classification model. Extensive experiments demonstrate that our method can accurately correct wrong labels and greatly improve the performance of state-of-the-art RE models.

Acknowledgements

This research work has been funded by the National Natural Science Foundation of China (Grant No.61772337, U1736207), and the National Key Research and Development Program of China NO. 2016QY03D0604.

References

1. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: Proceedings of HLT/EMNLP. pp. 724–731. Association for Computational Linguistics (2005)
2. Feng, J., Huang, M., Zhao, L., Yang, Y., Zhu, X.: Reinforcement learning for relation classification from noisy data. In: Proceedings of AAAI (2018)
3. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th ACL: Human Language Technologies-Volume 1. pp. 541–550 (2011)

4. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. p. 22 (2004)
5. Lei, K., Chen, D., Li, Y., Du, N., Yang, M., Fan, W., Shen, Y.: Cooperative denoising for distantly supervised relation extraction. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 426–436 (2018)
6. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th ACL (Volume 1: Long Papers). vol. 1, pp. 2124–2133 (2016)
7. Liu, L., Ren, X., Zhu, Q., Zhi, S., Gui, H., Ji, H., Han, J.: Heterogeneous supervision for relation extraction: A representation learning approach. arXiv preprint arXiv:1707.00166 (2017)
8. Liu, T., Wang, K., Chang, B., Sui, Z.: A soft-label method for noise-tolerant distantly supervised relation extraction. In: EMNLP. pp. 1790–1795 (2017)
9. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: 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. pp. 1003–1011. Association for Computational Linguistics (2009)
10. Nguyen, T.H., Grishman, R.: Relation extraction: Perspective from convolutional neural networks. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. pp. 39–48 (2015)
11. Pearson, K.: Note on regression and inheritance in the case of two parents. Proceedings of the Royal Society of London **58**, 240–242 (1895)
12. Qian, L., Zhou, G., Kong, F., Zhu, Q., Qian, P.: Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. pp. 697–704. Association for Computational Linguistics (2008)
13. Qin, P., Xu, W., Wang, W.Y.: Dsgan: Generative adversarial training for distant supervision relation extraction. arXiv preprint arXiv:1805.09929 (2018)
14. Qin, P., Xu, W., Wang, W.Y.: Robust distant supervision relation extraction via deep reinforcement learning. arXiv preprint arXiv:1805.09927 (2018)
15. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 148–163. Springer (2010)
16. Santos, C.N.d., Xiang, B., Zhou, B.: Classifying relations by ranking with convolutional neural networks. arXiv preprint arXiv:1504.06580 (2015)
17. Suchanek, F.M., Ifrim, G., Weikum, G.: Combining linguistic and statistical analysis to extract relations from web documents. In: the 12th ACM SIGKDD. pp. 712–717. ACM (2006)
18. Wang, L., Cao, Z., De Melo, G., Liu, Z.: Relation classification via multi-level attention cnns. In: the 54th ACL (long papers). vol. 1, pp. 1298–1307 (2016)
19. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: EMNLP. pp. 1753–1762 (2015)
20. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING: Technical Papers. pp. 2335–2344 (2014)
21. Zhang, D., Wang, D.: Relation classification via recurrent neural network. arXiv preprint arXiv:1508.01006 (2015)
22. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th ACL (Volume 2: Short Papers). vol. 2, pp. 207–212 (2016)