ARNOR: Attention Regularization based Noise Reduction for Distant Supervision Relation Classification

Wei Jia, Dai Dai, Xinyan Xiao and Hua Wu
Baidu Inc., Beijing, China
{jiawei07, daidai, xiaoxinyan, wu_hua} @baidu.com

Abstract

Distant supervision is widely used in relation classification in order to create large-scale training data by aligning a knowledge base with an unlabeled corpus. However, it also introduces amounts of noisy labels where a contextual sentence actually does not express the labeled relation. In this paper, we propose ARNOR, a novel **A**ttention **R**egularization based **NO**ise **R**eduction framework for distant supervision relation classification. ARNOR assumes that a trustable relation label should be explained by the neural attention model. Specifically, our ARNOR framework iteratively learns an interpretable model and utilizes it to select trustable instances. We first introduce attention regularization to force the model to pay attention to the patterns which explain the relation labels, so as to make the model more interpretable. Then, if the learned model can clearly locate the relation patterns of a candidate instance in training set, we will select it as a trustable instance for further training step. According to the experiments on NYT data, our ARNOR framework achieves significant improvements over state-of-the-art methods in both relation classification performance and noise reduction effect.

1 Introduction

Relation Classification (RC) is a fundamental task in natural language processing (NLP) and is particularly important for knowledge base construction. The goal of RC (Zelenko et al., 2003) is to identify the relation type of a given entity pair in a sentence. Generally, a relation should be explicitly expressed by some clue words. See the first sentence in Figure 1. The phrase "was born in" explains the relation type "place_of_birth" for "Bill Lockyer" and "California". Such indicating words is called *patterns* (Hearst, 1992; Hamon and Nazarenko, 2001).

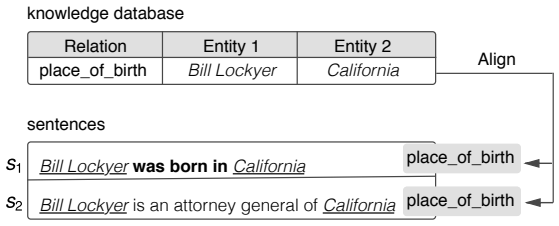

Figure 1: Two relation instances generated by distant supervision. The bold words "was born in" in s_1 is the pattern that explains the relation type "place_of_birth". Hence, this instance is correctly labeled. However, the second instance is noisy due to the lack of corresponding relation pattern.

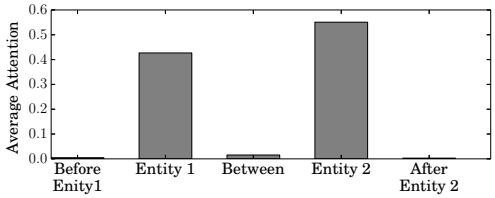

Figure 2: Average attention weights of BiLSTM+ATT model across five parts in the sentences on our test set. This model is trained using noisy data generated by distant supervision. It mainly pays attention to the input entity pair and ignores other words which might express the real relation. It also happens in Figure 1. This result comes from the fact that DS method only depends on entities for labeling data.

In order to cheaply obtain a large amount of labeled RC training data, Distant Supervision (DS) (Mintz et al., 2009) was proposed to automatically generate training data by aligning a knowledge base with an unlabeled corpus. It is built on a weak assumption that if an entity pair have a relationship in a knowledge base, all sentences that contain this pair will express the corresponding relation.

Unfortunately, DS obviously brings plenty of noisy data, which may significantly reduce the

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1399–1408
Florence, Italy, July 28 - August 2, 2019.

performance of an RC model. There may be no explicit relation pattern for identifying the relation. See the second sentence in Figure 1 for example. Mintz et al. (2009) reports that distant supervision may lead to more than 30% noisy instances. On the other hand, based on these noisy data, attention-based neural models often only attend to entity words but fail to attend to patterns (See Figure 2).

There are mainly three kinds of methods for dealing with such noise problem. First, multi-instance learning (Riedel et al., 2010; Lin et al., 2016; Surdeanu et al., 2012; Zeng et al., 2015) relaxes the DS assumption as at-least-one. In a bag of sentences that mention the same entity pair, it assumes that at least one sentence expresses the relation. Multi-instance learning carries out classification on bag-level and often fails to perform well on sentence-level prediction (Feng et al., 2018b). Secondly, in order to reduce noise for sentence-level prediction, researchers then resort to reinforcement learning or adversarial training to select trustable data (Feng et al., 2018b; Qin et al., 2018a; Han et al., 2018; Xiangrong et al., 2018; Qin et al., 2018b). This line of research selects confident relation labels by matching the predicted label of the learned model with DS-generated label. As the model is also learned from DS data, it might still fail when model predictions and DS-generated labels are both wrong. The third method relies on relation patterns. Pattern-based extraction is widely used in information extraction (Hearst, 1992; Hamon and Nazarenko, 2001). Among them, the generative model (Takamatsu et al., 2012) directly models the labeling process of DS and finds noisy patterns that mistakenly label a relation. Data programming (Ratner et al., 2016, 2017) fuses DS-based labels and manual relation patterns for reducing noise.

In this paper, we propose ARNOR, a novel attention regularization based framework for noise reduction. ARNOR aims to train a neural model which is able to clearly explain the relation patterns through Attention Regularization (AR), and at the same time reduce noise based on an assumption: the clearer the model explain the relation in an instance, the more trustable this instance is. Specifically, our ARNOR framework iteratively learns the interpretable model and selects trustable instances. We first use attention regularization on the neural model to focus on relation patterns (Section 3.4 will introduce the patterns construction). Then, if the learned model can dicover patterns for candidate instances, we will select these candidates as correct labeled data for further training step. These two steps are mutually reinforced. The more interpretable the model is, the better training data is selected, and vice versa.

In addition, most previous DS-based RC models are evaluated approximately on the test set which is split from the training set and thus is also full of noisy data. We argue that this might not be the best choice. Instead, we use a recently released sentence-level test set (Ren et al., 2017) for evaluation. However, there also exist several problems in this test set (see Sec. 4.1). We come up with a revised version that is larger and more precise.

Overall, the contribution is as follows:

1. We propose a novel attention regularization method for reducing the noise in DS. Our method forces the model to clearly explain the relation patterns in terms of attention, and selects trustable instances if they can be explained by the model.

2. Our ARNOR framework achieves significant improvement over state-of-the-art noise reduction methods, in terms of both RC performance and noise reduction effect.

3. We publish a better manually labeled sentence-level test set[1] for evaluating the performance of RC models. This test set contains 1,024 sentences and 4,543 entity pairs, and is carefully annotated to ensure accuracy.

2 Related Work

We deal with DS-based RC in this paper. For RC task, various models are recently proposed based on different neural architectures, such as convolutional neural networks (Zeng et al., 2014, 2015) and recurrent neural network (Zhang et al., 2015; Zhou et al., 2016). To automatically obtain a large training dataset, DS has been proposed (Mintz et al., 2009). However, DS also introduces noisy data, making DS-based RC more challenging.

Previous studies make attempts on kinds of methods to solve the noise problem. The first widely studied method is based on multi-instance

[1]The dataset used in this paper is on https://github.com/PaddlePaddle/models/tree/develop/PaddleNLP/Research/ACL2019-ARNOR

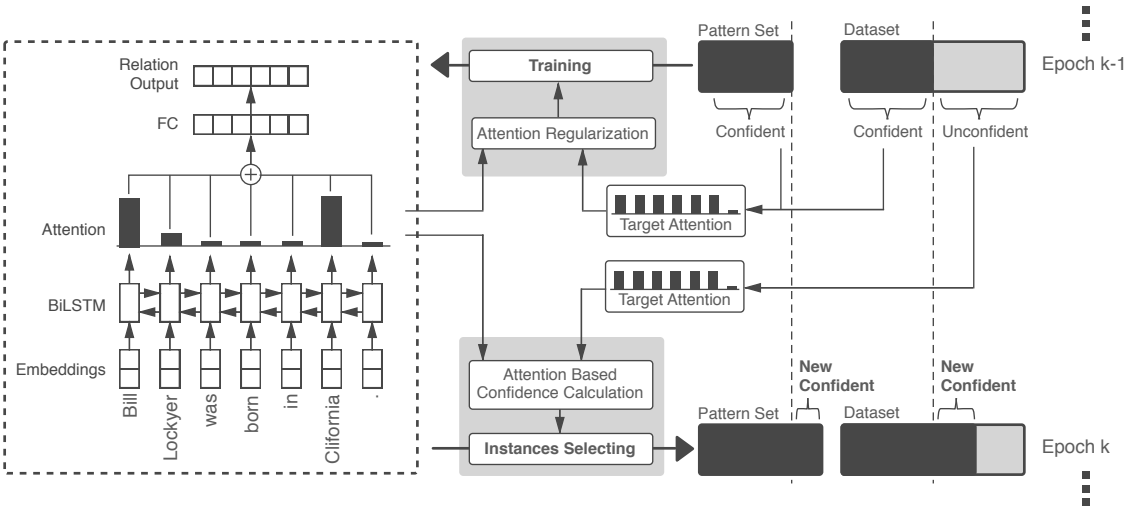

Figure 3: An overview of our ARNOR framework. It is based on a BiLSTM with attention mechanism and utilizes attention regularization to force the model to attend the corresponding relation patterns. Then, an instance selector calculates a confidence score for each training instance to generate a new redistributed training set and a new trustable pattern set. These two steps are run iteratively to form a bootstrap learning procedure.

learning (Riedel et al., 2010; Lin et al., 2016; Surdeanu et al., 2012; Zeng et al., 2015). However, it models noise problem on a bag of instances and is not suitable for sentence-level prediction. The second kind of approach utilizes RL (Feng et al., 2018b; Xiangrong et al., 2018; Qin et al., 2018b) or adversarial training (Qin et al., 2018a; Han et al., 2018) to select trustable instances. The third research line relies on patterns (Hearst, 1992; Hamon and Nazarenko, 2001). Takamatsu et al. (2012) directly models the labeling process of DS to find noisy patterns. Ratner et al. (2016, 2017) proposes to fuse DS-based labels and manual relation patterns for reducing noise. Feng et al. (2018a) presents a pattern extractor based on RL and uses extracted patterns as features for RC.

3 The ARNOR Framework

In this paper, we reduce DS noise and make the model more interpretable according to the observation that a relation should be expressed by its sentence context. Generally, RC classifier should rely on relation patterns to decide the relation type for a pair of entities. Thus, for a training instance, if such an interpretable model cannot attend to the pattern that expresses the relation type, it is possible that this instance is a noise.

Our ARNOR Framework consists of two parts: attention regularization training and instance selection. First, we hope the model is capable of locating relation patterns. Thus, attention regularization is applied to guide the training of the model, forcing it to pay attention to given pattern words. Then, we select instances by checking whether the model can give a clear explanation for the relation label generated by DS. These two steps will be repeated in a bootstrap procedure. We illustrate our method in Figure 3.

3.1 Attention-based BiLSTM Encoder

In order to capture the key feature words for identifying relations, we apply an attention mechanism over a BiLSTM Encoder, which is first introduced in (Zhou et al., 2016) for RC. The model architecture is illustrated on the left side of Figure 3.

Input Embeddings. The input embeddings consist of three parts: word embedding, position embedding, and entity type embedding. Position embedding is first proposed by Zeng et al. (2014) to incorporate position information of input entity pair and has been widely used in the following RC models. We also introduce entity type information by looking up an entity type embedding matrix. The final input embeddings are a concatenation of these embeddings, and are fed to a bidirectional Long Short Term Memory (BiLSTM) with an attention mechanism to generate sentence representation.

Attention-based BiLSTM. Let $\mathbf{H} = \{\mathbf{h}_i\}$ denotes the hidden vectors of BiLSTM encoder. The final sentence representation $\mathbf{u}$ is a weighted sum of

these vectors,

$$\begin{aligned}\mathbf{M} &= \tanh(\mathbf{H}) \\ \mathbf{a} &= \text{softmax}(\mathbf{w}^T\mathbf{M}) \\ \mathbf{u} &= \mathbf{H}\mathbf{a}^T\end{aligned} \tag{1}$$

where $\mathbf{w}^T$ is a trained parameter vector. It is demonstrated that attention mechanism is helpful in capturing important features for classification tasks. However, for noisy data generated by distant supervision, it almost only focuses on entities, but neglects relation patterns which are more informative for RC.

3.2 Training with Attention Regularization

Attention Regularization (AR) aims to teach the model to attend to the relation patterns for identifying relations. Given a T-word sentence $\mathbf{s} = \{x_i\}_{i=1}^T$, a pair of entities (e_1, e_2) in the $\mathbf{s}$, a relation label y, and a relation patterns m that explains the relation y of e_1 and e_2. (Section 3.4 will introduce the construction of relation patterns m). We are able to calculate an attention guidance value $\mathbf{a}^m$, according to pattern mention significance function $q(\mathbf{z}|\mathbf{s}, e_1, e_2, m)$ conditional on the input m. Here $\mathbf{z}$ represents the pattern words in a sentence. We hope that the classifier can approximate its attention distribution $\mathbf{a}^s = p(\mathbf{z}|\mathbf{s})$ to $\mathbf{a}^m$, where p represents the classifier network. Intuitively, we apply KL (Kullback–Leibler divergence) as the optimized function, which describes the differences between distributions:

$$KL(\mathbf{a}^m||\mathbf{a}^s) = \sum \mathbf{a}^m \log \frac{\mathbf{a}^m}{\mathbf{a}^s} \tag{2}$$

What is more, the Equation 2 can be further reduced as following:

$$\begin{aligned}loss_a &= \sum \mathbf{a}^m \log \frac{\mathbf{a}^m}{\mathbf{a}^s} \\ &= \sum (\mathbf{a}^m \log \mathbf{a}^m - \mathbf{a}^m \log \mathbf{a}^s)\end{aligned} \tag{3}$$

where $loss_a$ represents the loss of attention regularization. Because $\mathbf{a}^m$ contains fixed values, the equation is equal to

$$loss_a = -\sum \mathbf{a}^m \log \mathbf{a}^s \tag{4}$$

Therefore, we adapt $loss_a$ into classification loss $loss_c$ to regularize attention learning. The final $loss$ is

$$loss = loss_c + \beta loss_a \tag{5}$$

where β is a weight for $loss_a$, which is generally set as 1 in our experiments.

In this paper, we implement a fairly simple function to generate $\mathbf{a}^m$.

$$\begin{aligned}b_i &= \begin{cases}1 & x_i \in \{e_1, e_2, m\} \\ 0 & \text{else}\end{cases} \\ \mathbf{a}^m &= \left\{\frac{b_k}{\sum_{i=1}^T b_i}\right\}_{k=1}^T\end{aligned} \tag{6}$$

Here b denotes that whether x_i belongs to entity words and relation pattern words.

3.3 Instance Selection with Attention from Model

Based on attention mechanism, a trained RC model can tells us the importance of each word for identifying the relation type. For a training instance, if the relation pattern words that the model focuses on do not match the pattern m which explains the relation type, this instance is probably a false positive. Here we still apply KL to measure the probability that an instance is a false positive. Given the attention weights $\mathbf{a}^s$ from the RC model and $\mathbf{a}^m$ calculated by Equation 6, the confidence score c of an instance is normalized by

$$c = \frac{1}{1 + KL(\mathbf{a}^m||\mathbf{a}^s)} \tag{7}$$

The higher c is, the more confident an instance is. We calculate the confidence score for all instances in the training set and select instances whose score is more than a threshold c_t, which is a hyperparameter.

3.4 Bootstrap Learning Procedure

In our ARNOR framework, an important problem is how to acquire relation patterns m in model training and instance selecting step. In the model training step, we need more precise patterns in order to guide the model to attend to important evidence for RC. While in the instance selection step, more various patterns are required so as to select more trustable data as well as to discover more confident relation patterns. Here we will simply define the process of the bootstrap learning steps. In model training, given 1) a pattern extractor E which can extract a relation patterns from an instance, 2) an initial trustable pattern set $\mathbf{M}$ (which might be manually collected or simply counted up from original training dataset $\mathbf{D}$ using E). First,

Algorithm 1 The ARNOR Framework

Require: DS dataset $\mathbf{D}$, a relation classifier C with parameters $\boldsymbol{\theta}$

Collect high frequency patterns from $\mathbf{D}$ into $\mathbf{M}$
Redistribute $\mathbf{D}$ by $\mathbf{M}$
loop
    Train classifier C with $\mathbf{D}$ and $\mathbf{M}$
    Update parameters $\boldsymbol{\theta}$ by Attention Regularization
    Get confident score $\mathbf{c}$ by C for $\mathbf{D}$
    Update $\mathbf{M}$ by high score $\mathbf{c}$ from $\mathbf{D}$
    Redistribute $\mathbf{D}$ by new $\mathbf{M}$
end loop

we redistribute training dataset $\mathbf{D}$ based on $\mathbf{M}$ (described below). Then, the RC model is trained for epochs only using m in $\mathbf{M}$. Next, instance selection is run on $\mathbf{D}$ to select more confident training data. These new trustable instances are fed to E to figure out new trustable patterns and put them into $\mathbf{M}$. We repeat such a bootstrap procedure until the F1 score on dev set does not increase. This bootstrap procedure is detailed in Algorithm 1.

Relation Pattern Extraction. Another problem is how to build a relation pattern extractor E to extract a pattern from an instance. However, we find it is not quite critical. Even though we use a very simple method, we still achieve considerable improvement. It is certain that a more complicated and well-performed extractor will bring additional improvement. This will be one of our future work. Our pattern extractor E simply takes the words between two entities as a relation pattern. For the building of the initial pattern set $\mathbf{M}$, we extract relation patterns from all instances in original training dataset and count them up. $\mathbf{M}$ is initially built by selecting patterns with occurrences. We retain top 10% (maximum 20) patterns for each relation type.

Data Redistribution. After the trustable pattern set $\mathbf{M}$ is built, dataset $\mathbf{D}$ will be redistributed using these patterns. All positive instances that are not matched these patterns will be put into the negative set, revising their relation label to 'None'. We will explain the reason for data redistributing in our experiment section.

NYT	Training	Test
#Sentences	235,253	1,024
#Instances	371,461	4,543
#Positive instances	110,518	671

Table 1: Statistics of the dataset in our experiments.

NYT	Training	Test
#/location/location/contains	60,215	317
#/people/person/nationality	8,349	66
#/location/country/capital	7,959	13
#/people/person/place_lived	7,438	148
#/business/person/company	5,788	84
#/location/nei.../neighborhood_of	5,737	1
#/people/person/place_of_birth	3,279	14
#/people/person/place_of_death	2,002	9
#/business/company/founders	827	11
#/people/person/children	523	8

Table 2: The 10 relation types we retain and statistics of them in the dataset. The distribution of some relation types are distinct in test set because they are much more noisy.

4 Experiments

4.1 Dataset and Evaluation

We evaluate the proposed ARNOR framework on a widely-used public dataset: NYT, which is a news corpus sampled from 294k 1989-2007 New York Times news articles and is first presented in (Riedel et al., 2010). Most previous work commonly generates training instances by aligning entity pairs from Freebase and adopt held-out evaluation to evaluate without costly human annotation. Such an evaluation can only provide an approximate measure due to the noisy test set that is also generated by distant supervision. In contrast, Ren et al. (2017) publishes a training set which is also generated by distant supervision, but a manually-annotated test set that contains 395 sentences from Hoffmann et al. (2011). However, we find that this test set was annotated with only one entity pair for one sentence. Not all of the triplets in these sentences are marked out. In addition, although there are enough test instances (3,880 including "None" type), the number of positive ones is relatively small (only 396). Moreover, the test set only contains half of the relation types of the training set.

To address these issues and evaluate our ARNOR framework more precisely, we annotate and publish a new sentence-level test set (the source address is in section 1) on the basis of the one released by Ren et al. (2017), which also con-

Method	Dev			Test		
	Prec.	Rec.	F1	Prec.	Rec.	F1
CNN (Zeng et al., 2014)	38.32	65.22	48.28	35.75	64.54	46.01
PCNN (Zeng et al., 2015)	36.09	63.66	46.07	36.06	64.86	46.35
BiLSTM	36.71	**66.46**	47.29	35.52	**67.41**	46.53
BiLSTM+ATT	37.59	64.91	47.61	34.93	65.18	45.48
PCNN+SelATT (Lin et al., 2016)	46.01	30.43	36.64	45.41	30.03	36.15
CNN+RL$_1$ (Qin et al., 2018b)	37.71	52.66	43.95	39.41	61.61	48.07
CNN+RL$_2$ (Feng et al., 2018b)	40.00	59.17	47.73	40.23	63.78	49.34
ARNOR (Ours)	**62.45**	58.51	**60.36**	**65.23**	56.79	**60.90**

Table 3: Comparison of our method and other baselines. The first three methods are normal RC model, and the middle three baselines are models for distant supervision RC.

tains annotated named entity types. Firstly, we revise mislabeled instances on the original 395 testing sentences. Then, about 600 sentences are sampled and removed from the original training set. We carefully check their labels and merge them into the test set. We also remove some of the relation types which are overlapping and ambiguous or are too noisy to obtain a non-noise test sample. The details of this dataset and the relation types we used is shown in Table 1 and Table 2.

For evaluation, we evaluate our framework on sentence-level (or instance-level). Sentence-level prediction is more friendly with comprehend sentence tasks, like question answering and semantic parsing (Feng et al., 2018b). Different from commonly-used bag-level evaluation, a sentence-level evaluation compute Precision (Prec.), Recall (Rec.) and F1 metric directly on all of the individual instances in the dataset. We think such an evaluation is more intuitive and suitable for a real-world application.

4.2 Baselines

We compare our ARNOR framework with several strong baselines for noise reduction as follows:

PCNN+SelATT (Lin et al., 2016) is a bag-level RC model. It adopts an attention mechanism over all sentences in a bag and thus can reduce the weight of noise data.

CNN+RL$_2$ (Feng et al., 2018b) is a novel reinforcement learning (RL) based model for RC from noisy data. It jointly trains a CNN model for RC as well as an instance selector to remove unconfident samples.

CNN+RL$_1$ (Qin et al., 2018b) also introduces RL to heuristically recognize false positive instances. Different from Feng et al. (2018b), they redistribute false positives into negative samples instead of removing them.

Meanwhile, to demonstrate the effectiveness of RC after denoising, several non-denoising methods are also used for comparison.

CNN (Zeng et al., 2014) is a widely-used architecture for RE. It introduces position embeddings to represent the location of an input entity pair.

PCNN (Zeng et al., 2015) is a revision of CNN which uses piecewise max-pooling to extract more relation features.

BiLSTM (Zhang et al., 2015) is also commonly used for RE with the help of position embeddings.

BiLSTM+ATT (Zhou et al., 2016) adds an attention mechanism into BiLSTM to capture the most important features for identifying relations. It is the base model used in our ARNOR framework.

4.3 Implementation Details

For our model and other BiLSTM-based baselines, the word embeddings are randomly initialized with 100 dimensions. The position embeddings and entity type embeddings are randomly initialized with 50 dimensions. The size of BiLSTM hidden vector is set to 500. In attention regularization training, parameter β is set to 1. We set the learning rate as 0.001 and utilize Adam for optimization. To better evaluate our models, we averagely split the test dataset into a development set and a testing set. In instance selection step, an appropriate confidence score threshold is set to 0.5 that should be various in other datasets. And we take max 5 new patterns in a loop for each relation type. In bootstrap procedure, we run 10 epochs in the first loop, and 1 epoch in the rest loops until the classification performance on dev set dose not increase. Generally, the bootstrap procedure end

Model	Prec.	Rec.	F1
BiLSTM+ATT	34.93	**65.18**	45.48
+ IDR	**70.95**	40.57	51.63
+ ART	68.70	50.99	58.52
+ BLP	65.23	56.79	**60.90**

Table 4: Evaluation of components in our framework. BiLSTM+ATT is the base model without reducing noise. IDR stands for initial data redistributing using initial confident pattern set. ART denotes attention regularization training for the first loop. BLP stands for bootstrap learning procedure.

Model	Prec.	Rec.	F1
CNN	35.75	**64.54**	46.01
CNN+RL_2	40.23	63.78	49.34
CNN+IDR	**84.87**	39.94	54.32
CNN+IDR+RL_2	83.63	44.27	**57.89**

Table 5: Results of CNN+RL_2 (Feng et al., 2018b) starts with a pre-trained CNN model using initial data redistributing (IDR). CNN+IDR is the model trained on initially redistributed data and CNN+IDR+RL_2 applies RL_2 on pre-trained CNN+IDR model.

in 5 loops. For CNN-based baselines, we use the same embedding settings. The window size of the convolution layer is set to 3 and the number of the filter is set to 230. All the baselines for noise reduction were implemented with the source codes released by their authors.

4.4 Main Results

We compare the results of ARNOR with non-denoising baselines and denoising baselines. As shown in Table 3, ARNOR significantly outperforms all of the baselines in both precision and F1 metric, obtaining about 11% F1 improvement over the state-of-the-art CNN+RL_2. Note that our model achieves a tremendous improvement on precision without too much decline of recall. This demonstrates the proposed framework can effectively reduce the impact of noisy data. Besides, PCNN+SelATT performs the worst among all of the baselines. We think that it is because PCNN+SelATT is a bag-level method and is not suitable for sentence-level evaluation, which is consistent with Feng et al. (2018b).

Noise Reduction	Prec.	Rec.	F1
CNN+RL_2	40.58	**96.31**	57.10
ARNOR	**76.37**	68.13	**72.02**

Table 6: Comparison of effectiveness on noise reduction. We randomly sample 200 sentences (529 instances) from the training set. After manually checking, 213 of them are not noise. We use these samples to evaluate the capability of reducing noise.

5 Analysis and Discussion

5.1 Effects of components

In order to find which component contributes to our framework, we evaluate our model by adding each of the components. The results are shown in Table 4. BiLSTM+ATT is the baseline model that is trained by original noisy data. After using the initial redistributed dataset, which is generated by the method described in the above section, the BiLSTM+ATT model achieves about 6% improvement in F1. And the precision sharply increases by about 26%. This demonstrates that the DS dataset contains a large proportion of noise. Even such a simple filtering noise method can effectively improve model performance. However, this simple method seriously affects recall. On the one hand, amounts of true positives with long-tail patterns will be mistakenly regarded as false negatives. And we guess some relation patterns in training data are too rare to make the model learn to attend them. Therefore, after we add attention regularization to the model, the recall increases by about 10% with only 2% decline in precision. As a result, our model achieves another 7% F1 improvement. We believe this is the power of guiding the model to understand which words are more crucial for identifying relations. After we obtain an initial model trained by attention regularization, we continue the bootstrap learning procedure and finally achieve 2.4% F1 improvement. In this procedure, ARNOR will collect more confident long-tail patterns to improve the recall of the model.

5.2 Start with small clean or large noisy data

In the previous section, we have found that the initial redistributed dataset (with small but clean positive data) helps the model improve a lot. On the contrary, the previous neural network-based model for distant supervision RC, including all baselines in this paper, usually starts with the original dataset which is large but noisy. Which is the

	Entity 1: **AOL** Entity 2: **Jim Kimsey** Relation: **/business/company/founders**	Entity 1: **Kent Snyder** Entity 2: **Senomyx** Relation: **/business/person/company**
BiLSTM+ATT	Jim Kimsey , a founder of AOL ; Jack Valenti , former head … 0.36 0.19	… said Senomyx 's chief executive , Kent Snyder . 0.30 0.03 0.31
ARNOR	Jim Kimsey , a founder of AOL ; Jack Valenti , former head … 0.12 0.13 0.14 0.12	… said Senomyx 's chief executive , Kent Snyder . 0.15 0.15 0.13 0.13 0.11

Table 7: Here is attention cases with a heat map. These cases have shown our model's ability to locating relation indicators. Based on attention supervision, our model can concentrate on relation patterns and entities.

/people/person/children		
	#Occ	Pattern
High Frequency	7	$\mathbf{e_2}$, the son of $\mathbf{e_1}$
	4	$\mathbf{e_2}$, daughter of $\mathbf{e_1}$
Long Tail	1	$\mathbf{e_1}$'s youngest son , $\mathbf{e_2}$
	1	$\mathbf{e_2}$, the son of Secretary General $\mathbf{e_1}$
	1	$\mathbf{e_2}$, a daughter of Representative $\mathbf{e_1}$
/business/person/company		
	#Occ	Pattern
High Frequency	74	$\mathbf{e_2}$ secretary general , $\mathbf{e_1}$
	68	$\mathbf{e_1}$, the chairman of $\mathbf{e_2}$
	67	$\mathbf{e_1}$, chief executive of $\mathbf{e_2}$
Long Tail	4	$\mathbf{e_1}$, the secretary general of the $\mathbf{e_2}$
	3	$\mathbf{e_1}$, the chief executive of the $\mathbf{e_2}$
	3	$\mathbf{e_1}$, the oil minister of $\mathbf{e_2}$
	2	$\mathbf{e_1}$, the former chief executive of $\mathbf{e_2}$
	2	$\mathbf{e_1}$, the vice chairman of $\mathbf{e_2}$

Table 8: Pattern set cases. This table has shown some high frequency and top long tail patterns discovered by our model in pattern bootstrap.

better choice? In order to figure it out, we use the same initial redistributed dataset to pre-train the CNN which is used in the CNN+RL$_2$ and then apply RL$_2$ procedure for noise reduction on the original noisy dataset. We report the results in Table 5. The pre-trained PCNN also achieves a significant improvement, and after further denoising by RL$_2$, CNN+RL$_2$ finally obtain 57.89% in F1, which is still 3% lower than the performance of our model. Therefore, we consider that starting the model with a small but clean dataset might be a choice for noise reduction.

5.3 Effects of Noise Reduction

The instance selector in our ARNOR framework calculates a confidence score for each instance in the training set by checking whether the attention weights matches a given pattern. Then we utilize this confidence score to reduce noise. In order to verify the capability of reducing noise, we randomly sample 200 sentences to annotate whether they are noise and use them to evaluate the accuracy of noise reduction. We compare the results with CNN+RL$_2$ in Table 6. The ARNOR significantly outperforms CNN+RL$_2$ on percision and obtains a 14.92% F1 improvement.

5.4 Case Study

Our ARNOR is able to make the RC model more interpretable through attention regularization training. To verify this point, we select some instances from the test set and visualize their attention weights for a case study. As shown in Table 7, BiLSTM+ATT which is trained on original noisy data only focuses on the entity pairs, and makes wrong predictions on these cases. This is probably because the model does not learn the key evidence for RC. While ARNOR can perfectly capture the important features and correctly predict the relation.

In addition, we also check the confident patterns which are discovered in bootstrap learning. As presented in Table 8, the high-frequency patterns can be easily obtained by initially building of confident pattern set, and after bootstrap learning, we can discover more long-tail patterns, most of which are representative and meaningful. More importantly, some of these additional patterns are not similar in literal terms, demonstrating the model might learn the semantic correlation among related feature words.

6 Conclusion

We propose ARNOR, an attention regularization-based noise reduction framework for distant supervision relation classification. We find relation pattern is an important feature but is rarely captured by the previous model trained on noisy data. Thus, we design attention regulation to help the model learn the locating of relation patterns. With a more interpretable model, we then conduct noise reduction by evaluating how well the model explains the relation of an instance. A bootstrap learning procedure is built to iteratively improve the model, training data and trustable pattern set. With a very simple pattern extractor, we outperform several strong RL-based baselines, achieving

significant improvements on both relation classification and noise reduction. In addition, we publish a better manually labeled test set for sentence-level evaluation.

In the future, we hope to improve our work by the utilization of better model-based pattern extractor, and resorting to latent variable model (Kim et al., 2018) for jointly modeling instance selector. What is more, we also hope to verify the effectiveness of our method on more tasks, including open information extraction and event extraction, and also overlapping relation extraction models (Dai et al., 2019).

Acknowledgments

This work was supported by the Natural Science Foundation of China (No. 61533018).

References

Dai Dai, Xinyan Xiao, Yajuan Lyu, Qiaoqiao She, Shan Dou, and Haifeng Wang. 2019. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019), Honolulu, USA, January 27, 2019*.

Jun Feng, Minlie Huang, Yijie Zhang, Yang Yang, and Xiaoyan Zhu. 2018a. Relation mention extraction from noisy data with hierarchical reinforcement learning. *arXiv preprint arXiv:1811.01237*.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018b. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*.

Thierry Hamon and Adeline Nazarenko. 2001. Detection of synonymy links between terms: experiment and results. *Recent advances in computational terminology*, 2:185–208.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distant supervision for relation extraction via instance-level adversarial training. *arXiv preprint arXiv:1805.10959*.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Yoon Kim, Sam Wiseman, and Alexander M Rush. 2018. A tutorial on deep latent variable models of natural language. *arXiv preprint arXiv:1812.06834*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2124–2133.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018a. Dsgan: Generative adversarial training for distant supervision relation extraction. *arXiv preprint arXiv:1805.09929*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018b. Robust distant supervision relation extraction via deep reinforcement learning. *arXiv preprint arXiv:1805.09927*.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Advances in neural information processing systems*, pages 3567–3575.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024. International World Wide Web Conferences Steering Committee.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 721–729. Association for Computational Linguistics.

Zeng Xiangrong, Liu Kang, He Shizhu, Zhao Jun, et al. 2018. Large scaled relation extraction with reinforcement learning.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212.