Distant Supervision via Prototype-Based Global Representation Learning

Xianpei Han, Le Sun

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
{xianpei, sunle}@nfs.iscas.ac.cn

Abstract

Distant supervision (DS) is a promising technique for relation extraction. Currently, most DS approaches build relation extraction models in local instance feature space, often suffer from the multi-instance problem and the missing label problem. In this paper, we propose a new DS method — *prototype-based global representation learning*, which can effectively resolve the multi-instance problem and the missing label problem by learning informative entity pair representations, and building discriminative extraction models at the entity pair level, rather than at the instance level. Specifically, we propose *a prototype-based embedding algorithm*, which can embed entity pairs into a prototype-based global feature space; we then propose a neural network model, which can classify entity pairs into target relation types by summarizing relevant information from multiple instances. Experimental results show that our method can achieve significant performance improvement over traditional DS methods.

Introduction

Relation extraction (RE) is a fundamental task of text analysis and knowledge acquisition, which aims to identify and categorize relations between pairs of entities in text. For example, a RE system will extract a relation *Founder-of(Jobs, Apple)* from the sentence "*Jobs co-founded Apple in 1976*". Unfortunately, traditional relation extraction approaches (Kambhatla 2004; Zhang et al., 2006; etc.) are mostly supervised, thus require expensive labeled data and often suffer from the labeled data bottleneck when building relation extractors in web or open domain situation.

To resolve the labeled data bottleneck, a promising technique is *distant supervision (DS)*, which tries to build RE systems by exploiting the easily obtained relations/facts in a knowledge base as supervision (e.g., *Yago*[1], *DBPedia*[2] and *Freebase*[3]), rather than using labeled data. Figure 1 shows a classical DS system (Craven and Kumlien, 1999; Wu et al., 2007; Mintz et al., 2009): Firstly, all sentences containing specific entity pairs are collected from a text corpus. Then these instances are heuristically labeled by aligning them with relations in KB. Finally, the heuristically labeled training data is used to build relation extractors.

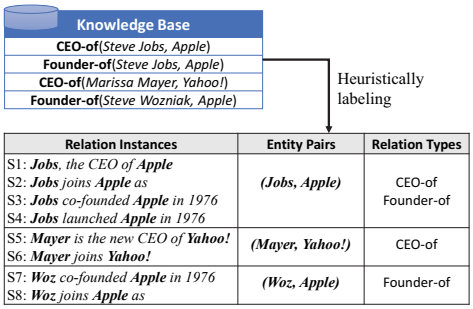

Figure 1.Training data labeling in DS system

Compared with traditional supervised RE systems, DS systems have several challenging problems. Firstly, while the objective of standard supervised RE systems is to classify relation instances (i.e., a sentence mentioned a specific entity pair such as *S1, S2, …, S8* in above), the objective of DS systems is to classify entity pairs, where each entity pair contains a collection of instances (e.g., *(Jobs, Apple)* is represented by four instances *{S1, S2, S3, S4}*) — this problem is referred as *the multi-instance problem*. Secondly, in standard supervision problem, the gold labels of all training instances are given, while in DS only entity pair labels are given, while the labels of all training instances are unknown – we refer to this problem as *the missing label problem*.

Traditionally, most DS approaches build relation extraction models in local instance feature space — this paper refers them as *instance-level models* (Bunescu & Mooney, 2007; Riedel et al., 2010; Yao et al., 2010; Hoffmann et al., 2010; Surdeanu et al., 2012; Ritter et al., 2013; Zeng et al., 2015; Han and Sun, 2016; etc.). This paradigm first learns instance-level classifiers which can classify individual instances into target relation types (e.g., classify *S1* into *CEO-*

[1] http://www.mpi-inf.mpg.de/yago-naga/yago
[2] http://www.dbpedia.org/
[3] http://www.freebase.com/

of or *Founder-of*), then the relation types of an entity pair are determined using the classification scores of its instances, by assuming a relationship between the labels of an entity pair and the labels of its instances (e.g., the commonly used at-least-one assumption). For instance, in Figure 1 the relation types of (*Jobs, Apple Inc.*) will be determined by the classification results of *S1, S2, S3* and *S4*.

The instance-level model based DS approaches, however, often suffer from the multi-instance problem and the missing label problem. Firstly, due to the multi-instance problem, a system should summarize relevant evidence from multiple instances. For example, to extract *Founder-of(Jobs, Apple)* in Figure 1, a system should summarize relevant evidence "*X co-found Y*" from *S3* and "*X launched Y*" from *S4*, meanwhile should identify "*X, the CEO of Y*" from *S1* and "*X join Y*" from *S2* as irrelevant evidence. Unfortunately, the discriminative learning process of the instance-level models occurs at the instance level, but not at the entity pair level. This fact prevents instance-level models from leveraging information beyond individual instances. Secondly, the multi-instance problem and the missing label problem make it a challenging problem to learn accurate classifiers in local instance feature space. For example, Figure 2(a) plots the instances in Figure 1, we can see that the optimal classification hyperplane is hard to learn due to the overlap of instances from different entity pairs and the missing labels of all instances.

In this paper, we propose a new DS method – *prototype-based global representation learning*, which can effectively solve the multi-instance problem and the missing label problem by learning informative entity pair representations. Specifically, we propose an effective global representation learning algorithm – *prototype-based embedding*, which will represent each entity pair by its similarities to a set of informative and discriminative relation prototypes. For example, in Figure 2(b) the entity pairs *(Jobs, Apple)*, *(Mayer, Yahoo!)* and *(Woz, Apple)* are embedded into a three-dimensional global feature space, with its dimensions corresponding to prototypes "*X co-found Y*", "*X launch Y*" and "*X is CEO of Y*". Based on the global representations of entity pairs, we propose a neural network, which can categorize these entity pairs into relation types by jointly embedding entity pairs and learning classifiers in global feature space. Compared with traditional DS methods, our method has the following advantages:

1) By learning global representations of entity pairs, our method can effectively solve the missing label problem and the multi-instance problem. Concretely, we solve the multi-instance problem by representing each entity pair as an individual point in global feature space (Figure 2(b)), rather than representing it as a set of points in instance feature space (Figure 2(a)). Furthermore, because the relation types of all training entity pairs are given in KB, there will be no missing label problem in entity pair level classification.

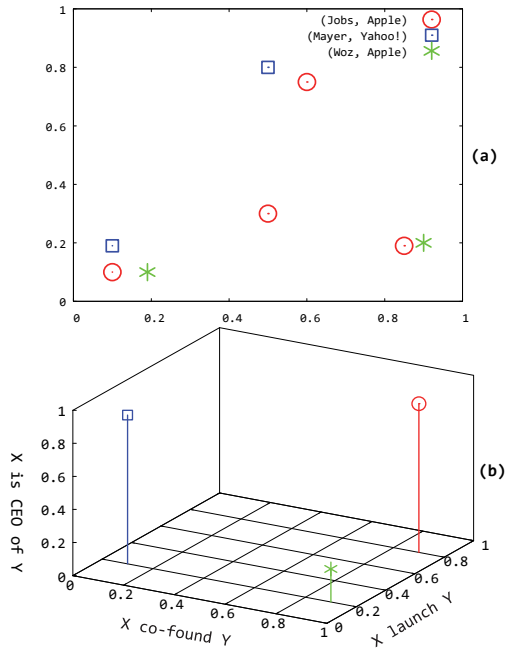

Figure 2. The representation of entity pairs: (a) The multi-instance representation; and (b) The global representation via prototype-based embedding

2) Compared with instance-level models, our model can better summarize information from multiple instances. Furthermore, the discriminative learning process of our model occurs at the entity pair level, which makes the learning problem of DS systems less challenging. For instance, in Figure 2, it is obviously easier to determine the optimal classification hyperplane in the global feature space than in the local instance feature space.

3) The prototype-based embedding algorithm can learn accurate global representation for entity pairs. Currently, to the best of our knowledge, there is only one global representation based DS method — Mintz et al. (2009), which represents an entity pair by simply combining together all features of its instances. Such a naïve global representation strategy, unfortunately, will introduce a lot of noisy information in global representation, e.g., the irrelevant information from "*Jobs join Apple*", "*Jobs left Apple*", "*Apple after Jobs*" will be used to extract *Founder-of(Jobs, Apple)*. By contrast, our method can filter out irrelevant instances via its similarities to a set of relation prototypes.

We conduct experiments on a standard data set. Experimental results show that our method can significantly improve the performance of DS systems.

This paper is organized as follows. Firstly we describe the embedding algorithm. Secondly we introduce our relation extraction system. Then we discuss experiments and review related work. Finally we conclude this paper.

Global Representation Learning via Prototype-based Embedding

In this section, we describe our global representation learning algorithm. We first introduce how to embed entity pairs into a prototype-based feature space, then we describe how to sample informative and discriminative prototypes.

Prototype-based Entity Pair Embedding

In DS systems, each entity pair **B** contains a set of instances $\mathbf{B} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathbf{B}|}\}$, where each instance $\mathbf{x_i}$ is represented as a feature vector $\mathbf{x_i} = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{in}]$ with $\mathbf{x}_{ij} = 1$ if $\mathbf{x}_i$ contains feature f_j, and 0 otherwise. Figure 3 demonstrates several feature vectors of the instances in Figure 1. Given an entity pair **B**, the goal of our global representation learning algorithm is to learn a global feature vector, which can provide a global view of all relevant information from different instances of **B**.

	CEO	*join*	*co-found*	*launch*	*of*	...
S1	1	0	0	0	1	…
S2	0	1	0	0	0	…
S3	0	0	1	0	0	…
S4	0	0	0	1	0	…

Figure 3. Several feature vectors of instances

In this paper, we learn the global feature vectors of entity pairs by assuming that there exists a set of prototypes for each relation type, and these prototypes can be used to capture the relevant evidence for extracting a specific relation type. For example, the *Founder-of* relation is often expressed using several regular patterns, such as "*X is the founder of Y*", "*X co-found Y*" and "*X launch Y in …*". Using these *Founder-of* prototypes, we can capture the *Founder-of* evidence of an instance by measuring its similarities with these prototypes. For example, we can capture *Founder-of* evidence of *"**Jobs** co-founds **Apple** in 1976"* via its similarity to the *Founder-of* prototype "*X co-found Y*".

Based on the above assumption, each prototype can be viewed as a global feature detector of entity pairs. That is, if an entity pair contains many instances which are similar to the prototypes of a relation type, then it will be highly likely to express this relation type. Formally, given a set of prototypes $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, we embed an entity pair **B** into the prototype-based feature space as $\mathrm{m}(\mathbf{B}) = [\mathrm{m}_1(\mathbf{B}), \mathrm{m}_2(\mathbf{B}), \ldots, \mathrm{m}_K(\mathbf{B})]$, where the k^{th} coordinate of $\mathrm{m}(\mathbf{B})$ is the max similarity between the instances of **B** and prototype $\mathbf{c}_k$:

$$\mathrm{m}_k(\mathbf{B}) = \max_i \ \mathrm{sim}(\mathbf{x}_i, \mathbf{c}_k, \mathbf{w}_k)$$

where $\mathrm{sim}(\mathbf{x}_i, \mathbf{c}_k, \mathbf{w}_k)$ is a parameterized similarity function between $\mathbf{x}_i$ and $\mathbf{c}_k$, and $\mathbf{w}_k$ is its parameters. There are many similarity functions can be used in our algorithm, including the weighted dot product:

$$\mathrm{sim}(\mathbf{x}_i, \mathbf{c}_k, \mathbf{w}_k) = \sum_j \mathbf{w}_{kj}\mathbf{x}_{ij}\mathbf{c}_{kj}$$

the Sigmoid function:

$$\mathrm{sim}(\mathbf{x}_i, \mathbf{c}_k, \mathbf{w}_k) = 1/(1+\exp(-\sum_j \mathbf{w}_{kj}\mathbf{x}_{ij}\mathbf{c}_{kj}))$$

the Radial Basis function (RBF):

$$\mathrm{sim}(\mathbf{x}_i, \mathbf{c}_k, \mathbf{w}_k) = \exp(-\frac{\sum_j (\mathbf{w}_{kj}\mathbf{x}_{ij} - \mathbf{w}_{kj}\mathbf{c}_{kj})^2}{\delta^2})$$

and the Rectifier function:

$$\mathrm{sim}(\mathbf{x}_i, \mathbf{c}_k, \mathbf{w}_k) = \max(0, \sum_j \mathbf{w}_{kj}\mathbf{x}_{ij}\mathbf{c}_{kj})$$

Using the above method, we can represent *(Jobs, Apple)* in Figure 1 as:

$$\text{(Jobs, Apple)} = \begin{bmatrix} \max \mathrm{sim}(., X \textit{ co-found } Y) \\ \max \mathrm{sim}(., X \textit{ launch } Y) \\ \max \mathrm{sim}(., X \textit{ is CEO of } Y) \end{bmatrix} = \begin{bmatrix} \mathrm{sim}(S3, X \textit{ co-found } Y) \\ \mathrm{sim}(S4, X \textit{ launch } Y) \\ \mathrm{sim}(S1, X \textit{ is CEO of } Y) \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.9 \end{bmatrix}$$

We can see that:

1) The global representation can summarize relevant evidence from multiple instances. In prototype-based feature space, each coordinate can represent information from a specific instance, and the information from different instances can be simultaneously represented in different coordinates. For example, in the above *(Jobs, Apple)* representation, the information from *S3*, *S4* and *S1* are simultaneously represented in 1st, 2nd and 3rd coordinates.

2) The global representation can identify relevant information and filter out irrelevant information for relation extraction: If an instance is irrelevant to the target relation types, then it should not be similar to any prototypes. For example, the information from "*Jobs joins Apple*" will be filtered out for the extraction of *Founder-of* and *CEO-of* relations, because it isn't similar to any *Founder-of* or *CEO-of* prototypes. This characteristic shields our global representation from the noisy information from irrelevant instances, which is especially important for effective DS systems.

Prototype Learning

It is obvious that the quality of prototypes is critical for global representation learning. This section describes how to learn informative and discriminative prototypes. Specifically, we employ the sampling strategy (Gu et al., 2001) to learn prototypes, i.e., we sample representative training instances as prototypes. Generally, a good prototype must satisfy the following two requirements (Liu & Motoda, 2013):

- *Goodness-of-exemplar*: a prototype must abstract out the central tendency of the instances expressing a specific relation type, i.e., a good prototype must cover a lot of instances of a relation type;
- *Goodness-of-discrimination*: a prototype must be able to distinguish relation instances from other relation types. For example, although "*X join Y*" may cover a lot of *CEO-*

of instances, it is not a good prototype because it cannot distinguish instances from many other relation types, such as *Founder-of*, *CFO-of*, *Manager-of* and *Employee-of*. By contrast "*X joint Y as its CEO*" will be a good *CEO-of* prototype because it is both discriminative and representative.

To meet the above two requirements, we propose a prototype learning method, which can: 1) ensure the goodness-of-discrimination by iteratively sampling prototypes from wrongly classified training instances; and 2) ensure the goodness-of-exemplar by sampling instances according to their goodness-of-exemplar scores. Specifically, given the training instances $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ of a specific relation type, we learn prototypes as follows:

Step 1: Initialize the prototype set by sampling from positive training instances via *Weighted Rejection Sampling algorithm*;

Step 2: Train our relation extraction model using current prototypes;

Step 3: Collect wrongly classified training instances from the wrongly classified entity pairs;

Step 4: Sample new prototypes from the wrongly classified instances via *Weighted Rejection Sampling algorithm* and add them to prototype set;

Step 5: Go to Step 2 until coverage.

In this paper, the above algorithm achieves coverage if the prototype number reaches a predefined threshold or our model achieves its best performance on the development data set.

To sample representative prototypes from instances, following the idea of DBSCAN clustering algorithm (Ester et al., 1996), we measure an instance's goodness-of-exemplar via its σ-NN value, i.e., how many instances whose similarity to this instance is larger than σ. For example, in Figure 4 the instance $\mathbf{x}_1$ is more representative than $\mathbf{x}_2$ because it has a larger σ-NN value.

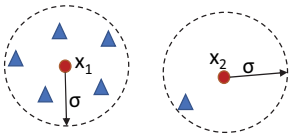

Figure 4.Measuring the goodness-of-exemplar via σ-NN value, here σ-NN($\mathbf{x}_1$)=5 *and* σ-NN($\mathbf{x}_2$)=1

Using the above goodness-of-exemplar measure, the *Weighted Rejection Sampling algorithm* is shown in below. We can see that, the algorithm ensures the goodness-of-exemplar by sampling instances according to their σ-NN values, and avoids redundant prototypes by rejecting prototype candidates similar to current prototypes.

Weighted Rejection Sampling Algorithm
Input: - The wrongly classified instances $\mathbf{X} = \{\mathbf{x}_1, \ldots \mathbf{x}_m\}$ - The number of sampled prototypes K - The similarity threshold σ **Output:** The new prototypes $\mathbf{C}=\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$

For $\mathbf{x}_i$ in $\mathbf{X}$:
Compute σ-NN($\mathbf{x}_i$)
End for
$\mathbf{C} \leftarrow \{\}$
While Size($\mathbf{C}$) < K:
Sample $\mathbf{x}$ from $\mathbf{X}$ with probability $\propto \exp(\sigma\text{-NN}(\mathbf{x}))$
If $\max_k \text{sim}(\mathbf{x}, \mathbf{c}_k) < \sigma$:
Add $\mathbf{x}$ to $\mathbf{C}$
End while

A Neural Network for Joint Global Representation Learning and Entity Pair Classification

This section describes how to build a global representation based relation extractor and how to learn its parameters. Generally, a global representation based relation extractor contains two components: 1) a global representation learning component which maps an entity pair into a global feature vector; and 2) an entity pair classifier which categorizes entity pairs into target relation types. We found that the above two components can be jointly modeled in a neural network architecture. Figure 5 shows the neural network we used in this paper.

The proposed neural network architecture provides a flexible framework for global representation-based DS systems: we can replace the prototype-based embedding layer with other global representation learning layers, e.g., the well-known convolutional layer. We can also change the entity pair classification layer to other classification layers, e.g., a softmax layer or even a multi-layer neural network itself.

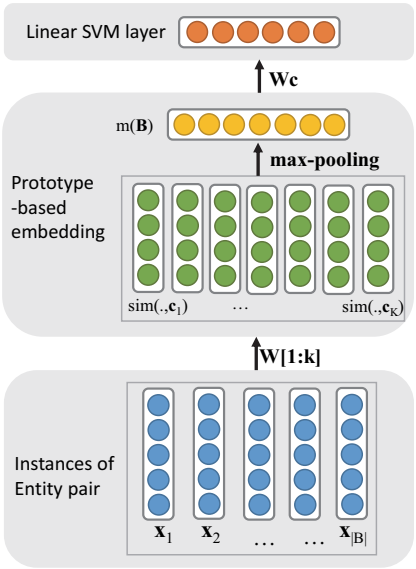

Figure 5. The neural network architecture of our system, where the prototype-based embedding is modeled via a prototype similarity layer and a max-pooling layer, and the entity pair classification is modeled via a linear SVM layer

Parameter Learning. Given a set of prototypes, we need to learn the following parameters:

- The parameters of the prototype-based embedding algorithm: [$\mathbf{w}_1$, $\mathbf{w}_2$, …, $\mathbf{w}_K$];
- The parameters of the entity pair classifier, in our neural network it is the weights of the linear SVM layer **Wc**.

In our above neural network framework, we can learn our system's parameters by first computing gradients of its parameters via Backpropagation algorithm, and then optimizing parameters using optimization algorithms such as SGD, AdaGrad and AdaDelta. In this paper, we optimize our model's parameters using the AdaDelta optimization algorithm. Because many entity pairs have more than one relation (e.g., *Steven Jobs* is both *CEO-of* and *Founder-of Apple Inc.*), this paper solves this multi-label problem by training a neural network model for each relation type using the "*one-versus-all*" strategy.

Experiments

Data Set

We conduct experiments on the commonly used KBP data set (Surdeanu et al., 2012). The KBP data set was constructed by aligning relations from English Wikipedia infoboxes against a document collection which contains the KBP shared task corpus (Ji et al., 2011) and the June 2010 version of Wikipedia. The KBP data set contains 41 relation types, 183,062 training relations and 3,334 testing relations. We evaluate all methods the same as Surdeanu et al. (2012): 1) relations are evaluated regardless of their support documents; and 2) only the gold relations mentioned in matched sentences are considered.

Systems and Baselines

We compare our method with four baselines:

Mintz++ – This is a traditional DS method proposed by Mintz et al.(2009), which represents an entity pair by simply combining all features of its instances together.

Hoffmann – This is an instance-model based multi-instance multi-label DS method proposed by Hoffmann et al. (2011), which first classifies instances into target relation types, then the label of an entity pair is determined from its instance labels via a deterministic *at-least-one* assumption.

Surdeanu – This is an instance-model based multi-instance multi-label DS approach proposed by Surdeanu et al. (2012), which first classifies instances into target relation types, then the entity pair labels are determined using its instance labels via a relational classifier.

DSCNN – This is a simpler version of our model: we replace the prototype-based global feature detector in our neural network (see Figure 5) with the well-known CNN feature detector (i.e., Convolutional layer + Max-pooling layer).

In our experiments, we use the implementations and the optimal settings of Stanford's MIMLRE package (Surdeanu et al., 2012) for the three baselines: *Mintz++*, *Hoffmann* and *Surdeanu*. For the *DSCNN* baseline and our prototype-embedding based method (referred as *DSProto* in below), we use the same partition as Surdeanu et al. (2012) for tuning and testing. Because positive/negative training instances are highly imbalanced, we put a higher misclassification cost (the tuned value is 2) to positive instances during training.

Overall Results

Following previous methods, we evaluate all systems using precision, recall and F-measure on the ranked relation extractions, and provide the precision/recall curves of all systems. The overall results are shown in Figure 6 and Table 1.

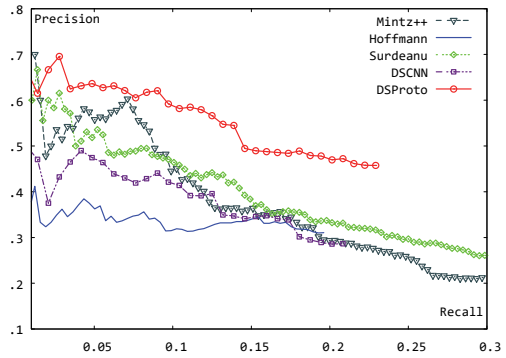

Figure 6. Precision/Recall curves on KBP data set

System	P	R	F1
Mintz++	0.260	0.250	0.255
Hoffmann	0.306	0.198	0.241
Surdeanu	0.249	0.314	0.278
DSCNN	0.286	0.214	0.244
DSProto	**0.459**	**0.231**	**0.307**

Table 1. The best F1-measures in P/R curves

From Figure 6 and Table 1, we can see that:

1)The prototype-based global representation learning method can achieve competitive performance: Compared with *Mintz++*, *Hoffmann*, *Surdeanu* and *DSCNN* baselines, our method correspondingly achieved 20%, 27%, 10% and 26% F1 improvements.

2) By better summarizing evidence from multiple relevant instances, the performance of DS systems can be improved: Compared with the two instance-level model based DS baselines *Hoffmann* and *Surdeanu*, our method correspondingly achieved 27% and 10% F1 improvements. We believe this is because our method provides a better way to exploit evidence from multiple instances: our method can simultaneously represent the evidence from multiple instances in the global feature vector, and can exploit them in discriminative learning/classification process.

3) By distinguishing relevant instances from irrelevant instances, our method can learn more accurate entity pair representations: Compared with the naïve global representation baseline – *Mintz++*, our method achieved 20% F1 improvements. This is because *Mintz++* cannot identify irrelevant instances, therefore its entity pair representation may be dominated by noisy information. By contrast, our method can effectively filter out irrelevant instances via its similarities to the learned prototypes.

4) Compared with the general CNN feature detector, our prototype-based feature detector can better capture the evidence for relation extraction: Compared with the *DSCNN* baseline, our method achieved 26% F1 improvement. We found the main reason is that the feature vectors of relation instances are both sparse and high-dimensional: The KBP data set contains more than 4,000,000 instance features but most instances contain only 10~30 features. In this case, it is too many parameters to learn effective CNN feature detectors. By contrast, the parameter size is compact in our prototype-based feature detector (proportional to the feature number in prototypes).

Detailed Analysis

In this section, we provide a detailed analysis for our method.

The effect of the size of prototypes. To assess how the size of prototypes will affect the extraction performance, we conduct experiments using different prototype sizes (proportional to the optimal size). The experimental results are shown in Table 2. We can see that our method achieved a stable performance on different sizes of prototypes: there is a small F1 decrease using a prototype set whose size is only 25% of the optimal size. This result also verified our assumption that a compact set of prototypes can cover most of the instances of a specific relation type.

	25%	50%	100%	200%
KBP	0.272	0.283	**0.307**	0.298

Table 2. The best F1-measures using different sizes (proportional to optimal size) of prototypes

Furthermore, we found that the optimal prototype numbers are different for different relation types, and our method can learn them adaptively. For instance, our method learned ~500 prototypes for the relation *org:city_of_headquarters*, while learned ~100 prototypes for the relation *per:age*.

The effect of iterative prototype learning. To assess the effect of our prototype learning method, we conduct two experiments: 1) we iteratively sample prototypes from wrongly classified training instances – this is the proposed prototype learning method; 2) we sample prototypes in an one-shot manner, i.e., we sample all prototypes at once from positive training instances via the Weighted Rejection Sampling algorithm. The experimental results are shown in Table 3. We can see that, iteratively sampling from wrongly classified instances can significantly improve the performance of our method. We believe this is because the one-shot sampling can only ensure the goodness-of-exemplar of prototypes, meanwhile our method can ensure both the goodness-of-exemplar and the goodness-of-discrimination.

	One-Shot	Iterative
KBP	0.286	**0.307**

Table 3. The best F1-measures of one-shot prototype sampling and iterative prototype sampling

Related Work

Currently, most DS approaches are focused on improving classification performance in local instance feature. Two most common strategies are multi-instance learning techniques and better training instance labeling algorithms.

The multi-instance learning based DS approaches focus on learning instance-level classifiers by modeling the label relationship between instance labels and entity pair labels. Currently, one of the most common approaches uses the at-least-one assumption (Bunescu and Mooney, 2007; Riedel et al., 2010; Yao et al., 2010; Hoffmann et al., 2010). In recent years, several other label relationship models are also proposed, e.g., the relational classifier (Surdeanu et al., 2012), the Markov Logic Network (Han & Sun, 2016). Due to the missing label problem, the learning of multi-instance models is often a challenging problem. Xu et al. (2013), Min et al. (2013), Ritter et al. (2013) and Zhang et al. (2013) further took the incompleteness of KB into consideration.

One other common strategy is to develop better training instance labeling algorithms. It is obvious that the original simple DS assumption (Craven and Kumlien, 1999; Wu et al., 2007; Mintz et al., 2009) will often fail and result in wrongly labeled training instances. Therefore a lot of methods are focused on eliminating wrongly labeled training instances (Takamatsu et al., 2012; Roth and Klakow, 2013; Han and Sun, 2014; Hoffmann et al., 2010; Zhang et al., 2010; Roller et al., 2015; Bing et al., 2015).

There were also some other strategies for improving system. Nguyen and Moschitti (2011) and Pershina et al. (2014) infused labeled corpus with heuristically labeled DS corpus. Riedel et al. (2013) and Fan et al. (2014) exploited the co-occurrence statistics between relations/instances/features. Zeng et al.(2014) propose a piecewise CNN which can better represent relation instances.

Conclusions

This paper describes a new distant supervision paradigm – *global representation learning-based distant supervision* and proposes an effective global representation learning algorithm – *prototype-based embedding*. By learning informative entity pair representations, our method can achieve competitive performance. This paper uses manually designed instance features to represent instances, in future we want to develop a neural network which can jointly embed relation instances and entity pairs.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants no. 61572477, 61433015 and 61272324, and the National High Technology Development 863 Program of China under Grants no. 2015AA015405. Moreover, we sincerely thank the reviewers for their valuable comments.

References

Augenstein, I., Vlachos, A., & Maynard, D. *Extracting Relations between Non-Standard Entities using Distant Supervision and Imitation Learning*. In: Proceedings of EMNLP 2015, pp. 747-757.

Bing, L., Chaudhari, S., Wang, R. C., & Cohen, W. W. 2015. *Improving Distant Supervision for Information Extraction Using Label Propagation through Lists*. In: Proceedings of EMNLP 2015, pp. 524-529.

Bunescu, R. C. and Mooney, R. 2007. *Learning to extract relations from the web using minimal supervision*. In: Proceedings of ACL 2007, pp. 576-583.

Craven, M. and J. Kumlien. 1999. *Constructing biological knowledge bases by extracting information from text sources*. In: Proceedings of AAAI 1999, pp. 77-86.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. 1996. *A density-based algorithm for discovering clusters in large spatial databases with noise*. In: Proceedings of KDD 1996, pp. 226-231.

Fan, M., Zhao, D., Zhou, Q., Liu, Z., Zheng, T. F., & Chang, E. Y. 2014. *Distant Supervision for Relation Extraction with Matrix Completion*. In: Proceedings of ACL 2014, pp. 839-849.

Gu, B., Hu, F., & Liu, H. 2001. *Sampling: knowing whole from its part.* In: Instance Selection and Construction for Data Mining (pp. 21-38). Springer US.

Gupta, R. and Sarawagi, S. 2011. *Joint training for open-domain extraction on the web: exploiting overlap when supervision is limited*. In: Proceedings of WSDM 2011, pp. 217-226.

Han, X. and Sun, L. 2014. *Semantic Consistency: A Local Subspace Based Method for Distant Supervised Relation Extraction*. In: Proceedings of ACL 2014, pp. 718-724.

Han, X. and Sun, L. 2016. Global Distant Supervision for Relation Extraction. In: Proceedings of AAAI 2016.

Hoffmann, R., Zhang, C., et al. 2010. *Learning 5000 relational extractors*. In: Proceedings of ACL 2010, pp. 286-295.

Hoffmann, R., Zhang, C., et al. 2011. *Knowledge-based weak supervision for information extraction of overlapping relations*. In: Proceedings of ACL 2011, pp. 541-550.

Ji, H., Grishman, R., et al. 2011. *Overview of the TAC 2011 knowledge base population track*. In: Proceedings of TAC 2011.

Kambhatla, N. 2004. *Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations*. In: Proceedings of ACL 2004, pp. 178–181.

Krause, S., Li, H., et al. 2012. *Large-Scale learning of relation-extraction rules with distant supervision from the web*. In: Proceedings of ISWC 2012, pp. 263-278.

Liu, H., & Motoda, H. (Eds.). 2013. *Instance selection and construction for data mining* (Vol. 608). Springer.

Mintz, M., Bills, S., et al. 2009. *Distant supervision for relation extraction without labeled data*. In: Proceedings of ACL-AFNLP 2009, pp. 1003-1011.

Min, B., Grishman, R., et al. 2013. *Distant Supervision for Relation Extraction with an Incomplete Knowledge Base*. In: Proceedings of NAACL-HLT 2013, pp. 777-782.

Nguyen, T. T. and Moschitti, A. 2011. *Joint distant and direct supervision for relation extraction*. In: Proceedings of IJCNLP 2011, pp. 732-740.

Pershina, M., et al. 2014. *Infusion of Labeled Data into Distant Supervision for Relation Extraction*. In: Proceedings of ACL 2014, pp. 732-738.

Riedel, S., Yao, L., et al. 2010. *Modeling relations and their mentions without labeled text*. In: Proceedings of Machine Learning and Knowledge Discovery in Databases, 2010, pp. 148-163.

Riedel, S., Yao, L., et al. 2013. *Relation Extraction with Matrix Factorization and Universal Schemas*. In: Proceedings of NAACL-HLT 2013, pp. 74-84.

Ritter, A., Zettlemoyer, L., Mausam, Etzioni, O. 2013. *Modeling Missing Data in Distant Supervision for Information Extraction*. In: Transactions of the Association for Computational Linguistics, Vol 1, pp. 367-378.

Roller, R., Agirre, E., Soroa, A., & Stevenson, M. 2015. Improving distant supervision using inference learning. In: Proceedings of ACL 2015, pp. 273--278.

Roth, B. and Klakow, D. 2013. *Combining Generative and Discriminative Model Scores for Distant Supervision*. In: Proceedings of ACL 2013, pp. 24-29.

Surdeanu, M., Tibshirani, J., et al. 2012. *Multi-instance multi-label learning for relation extraction*. In: Proceedings of EMNLP-CoNLL 2012, pp. 455-465.

Takamatsu, S., Sato, I., et al. 2012. *Reducing wrong labels in distant supervision for relation extraction*. In: Proceedings of ACL 2012, pp. 721-729.

Takase, S., Okazaki, N. and Inui, K., 2015. *Fast and Large-scale Unsupervised Relation Extraction.* In: Proceedings of 29th Pacific Asia Conference on Language, Information and Computation 2015, pp.96-105.

Wu, F. and Weld, D. S. 2007. *Autonomously semantifying Wikipedia*. In: Proceedings of CIKM 2007, pp. 41-50.

Xu, W., Hoffmann, R., et al. 2013. *Filling Knowledge Base Gaps for Distant Supervision of Relation Extraction*. In: Proceedings of ACL 2013, pp. 665-670.

Yao, L., Riedel, S., et al. 2010. *Collective cross-document relation extraction without labelled data*. In: Proceedings of EMNLP 2010, pp. 1013-1023.

Zeng, D., Liu, K., Chen, Y., & Zhao, J. 2015. *Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks*. In: Proceedings of EMNLP 2015, pp. 1753-1762.

Zhang, C., Hoffmann, R., et al. 2012. *Ontological smoothing for relation extraction with minimal supervision*. In: Proceedings of AAAI 2012, pp. 157-163.

Zhang, X., Zhang, J., et al. 2013. *Towards Accurate Distant Supervision for Relational Facts Extraction*. In: Proceedings of ACL 2013, pp. 810-815.

Zhang, M., Zhang, J., and Su, J. 2006. *Exploring syntactic features for relation extraction using a convolution tree kernel*. In: Proceedings of NAACL-HLT 2006, pp. 288–295.