# Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks

**Daojian Zeng, Kang Liu, Yubo Chen and Jun Zhao**
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
{djzeng,kliu,yubo.chen,jzhao}@nlpr.ia.ac.cn

## Abstract

Two problems arise when using distant supervision for relation extraction. First, in this method, an already existing knowledge base is heuristically aligned to texts, and the alignment results are treated as labeled data. However, the heuristic alignment can fail, resulting in wrong label problem. In addition, in previous approaches, statistical models have typically been applied to ad hoc features. The noise that originates from the feature extraction process can cause poor performance.

In this paper, we propose a novel model dubbed the Piecewise Convolutional Neural Networks (PCNNs) with multi-instance learning to address these two problems. To solve the first problem, distant supervised relation extraction is treated as a multi-instance problem in which the uncertainty of instance labels is taken into account. To address the latter problem, we avoid feature engineering and instead adopt convolutional architecture with piecewise max pooling to automatically learn relevant features. Experiments show that our method is effective and outperforms several competitive baseline methods.

## 1 Introduction

In relation extraction, one challenge that is faced when building a machine learning system is the generation of training examples. One common technique for coping with this difficulty is distant supervision (Mintz et al., 2009) which assumes that if two entities have a relationship in a known knowledge base, then all sentences that mention these two entities will express that relationship in some way. Figure 1 shows an example of the automatic labeling of data through distant supervision. In this example, *Apple* and *Steve Jobs* are two related entities in Freebase[1]. All sentences that contain these two entities are selected as training instances. The distant supervision strategy is an effective method of automatically labeling training data. However, it has two major shortcomings when used for relation extraction.

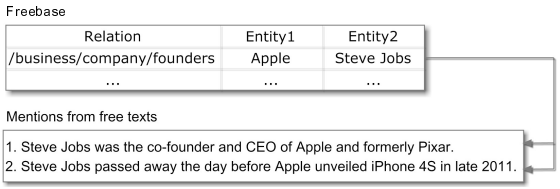



Figure 1: Training instances generated through distant supervision. Upper sentence: correct labeling; lower sentence: incorrect labeling.

First, the distant supervision assumption is too strong and causes the wrong label problem. A sentence that mentions two entities does not necessarily express their relation in a knowledge base. It is possible that these two entities may simply share the same topic. For instance, the upper sentence indeed expresses the "company/founders" relation in Figure 1. The lower sentence, however, does not express this relation but is still selected as a training instance. This will hinder the performance of a model trained on such noisy data.

Second, previous methods (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011) have typically applied supervised models to elaborately designed features when obtained the labeled data through distant supervision. These features are often derived from preexisting Natural Language Processing (NLP) tools. Since errors inevitably exist in NLP tools, the use of traditional features leads to error propagation or accumulation. Distant supervised relation extraction generally ad-

[1]http://www.freebase.com/



*Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762,
Lisbon, Portugal, 17-21 September 2015. 

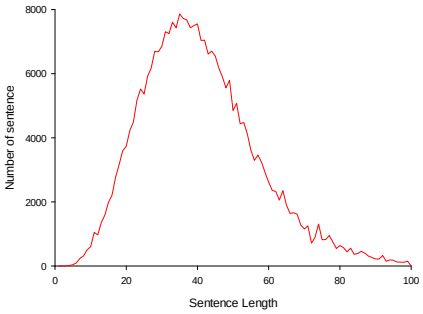



Figure 2: The sentence length distribution of Riedel's dataset.

dresses corpora from the Web, including many informal texts. Figure 2 shows the sentence length distribution of a benchmark distant supervision dataset that was developed by Riedel et al. (2010). Approximately half of the sentences are longer than 40 words. McDonald and Nivre (2007) showed that the accuracy of syntactic parsing decreases significantly with increasing sentence length. Therefore, when using traditional features, the problem of error propagation or accumulation will not only exist, it will grow more serious.

In this paper, we propose a novel model dubbed Piecewise Convolutional Neural Networks (PCNNs) with multi-instance learning to address the two problems described above. To address the first problem, distant supervised relation extraction is treated as a multi-instance problem similar to previous studies (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012). In multi-instance problem, the training set consists of many bags, and each contains many instances. The labels of the bags are known; however, the labels of the instances in the bags are unknown. We design an objective function at the bag level. In the learning process, the uncertainty of instance labels can be taken into account; this alleviates the wrong label problem.

To address the second problem, we adopt convolutional architecture to automatically learn relevant features without complicated NLP preprocessing inspired by Zeng et al. (2014). Our proposal is an extension of Zeng et al. (2014), in which a single max pooling operation is utilized to determine the most significant features. Although this operation has been shown to be effective for textual feature representation (Collobert et al., 2011; Kim, 2014), it reduces the size of the hidden layers too rapidly and cannot capture the structural information between two entities (Graham, 2014). For example, to identify the relation between *Steve Jobs* and *Apple* in Figure 1, we need to specify the entities and extract the structural features between them. Several approaches have employed manually crafted features that attempt to model such structural information. These approaches usually consider both internal and external contexts. A sentence is inherently divided into three segments according to the two given entities. The internal context includes the characters inside the two entities, and the external context involves the characters around the two entities (Zhang et al., 2006). Clearly, single max pooling is not sufficient to capture such structural information. To capture structural and other latent information, we divide the convolution results into three segments based on the positions of the two given entities and devise a piecewise max pooling layer instead of the single max pooling layer. The piecewise max pooling procedure returns the maximum value in each segment instead of a single maximum value over the entire sentence. Thus, it is expected to exhibit superior performance compared with traditional methods.

The contributions of this paper can be summarized as follows.

- We explore the feasibility of performing distant supervised relation extraction without hand-designed features. PCNNS are proposed to automatically learn features without complicated NLP preprocessing.
- To address the wrong label problem, we develop innovative solutions that incorporate multi-instance learning into the PCNNS for distant supervised relation extraction.
- In the proposed network, we devise a piecewise max pooling layer, which aims to capture structural information between two entities.

## 2 Related Work

Relation extraction is one of the most important topics in NLP. Many approaches to relation extraction have been developed, such as bootstrapping, unsupervised relation discovery and supervised classification. Supervised approaches are the most commonly used methods for relation

extraction and yield relatively high performance (Bunescu and Mooney, 2006; Zelenko et al., 2003; Zhou et al., 2005). In the supervised paradigm, relation extraction is considered to be a multi-class classification problem and may suffer from a lack of labeled data for training. To address this problem, Mintz et al. (2009) adopted Freebase to perform distant supervision. As described in Section 1, the algorithm for training data generation is sometimes faced with the wrong label problem. To address this shortcoming, (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012) developed the relaxed distant supervision assumption for multi-instance learning. The term 'multi-instance learning was coined by (Dietterich et al., 1997) while investigating the problem of predicting drug activity. In multi-instance learning, the uncertainty of instance labels can be taken into account. The focus of multi-instance learning is to discriminate among the bags.

These methods have been shown to be effective for relation extraction. However, their performance depends strongly on the quality of the designed features. Most existing studies have concentrated on extracting features to identify the relations between two entities. Previous methods can be generally categorized into two types: feature-based methods and kernel-based methods. In feature-based methods, a diverse set of strategies is exploited to convert classification clues (e.g., sequences, parse trees) into feature vectors (Kambhatla, 2004; Suchanek et al., 2006). Feature-based methods suffer from the necessity of selecting a suitable feature set when converting structured representations into feature vectors. Kernel-based methods provide a natural alternative to exploit rich representations of input classification clues, such as syntactic parse trees. Kernel-based methods enable the use of a large set of features without needing to extract them explicitly. Several kernels have been proposed, such as the convolution tree kernel (Qian et al., 2008), the subsequence kernel (Bunescu and Mooney, 2006) and the dependency tree kernel (Bunescu and Mooney, 2005).

Nevertheless, as mentioned in Section 1, it is difficult to design high-quality features using existing NLP tools. With the recent revival of interest in neural networks, many researchers have investigated the possibility of using neural networks to automatically learn features (Socher et al., 2012; Zeng et al., 2014). Inspired by Zeng et al. (2014), we propose the use of PCNNs with multi-instance learning to automatically learn features for distant supervised relation extraction. Dietterich et al. (1997) suggested that the design of multi-instance modifications for neural networks is a particularly interesting topic. Zhang and Zhou (2006) successfully incorporated multi-instance learning into traditional Backpropagation (BP) and Radial Basis Function (RBF) networks and optimized these networks by minimizing a sum-of-squares error function. In contrast to their method, we define the objective function based on the cross-entropy principle.

## 3 Methodology

Distant supervised relation extraction is formulated as multi-instance problem. In this section, we present innovative solutions that incorporate multi-instance learning into a convolutional neural network to fulfill this task. PCNNs are proposed for the automatic learning of features without complicated NLP preprocessing. Figure 3 shows our neural network architecture for distant supervised relation extraction. It illustrates the procedure that handles one instance of a bag. This procedure includes four main parts: *Vector Representation*, *Convolution*, *Piecewise Max Pooling* and *Softmax Output*. We describe these parts in detail below.

### 3.1 Vector Representation

The inputs of our network are raw word tokens. When using neural networks, we typically transform word tokens into low-dimensional vectors. In our method, each input word token is transformed into a vector by looking up pre-trained word embeddings. Moreover, we use position features (PFs) to specify entity pairs, which are also transformed into vectors by looking up position embeddings.

#### 3.1.1 Word Embeddings

Word embeddings are distributed representations of words that map each word in a text to a 'k'-dimensional real-valued vector. They have recently been shown to capture both semantic and syntactic information about words very well, setting performance records in several word similarity tasks (Mikolov et al., 2013; Pennington et al., 2014). Using word embeddings that have been trained a priori has become common practice for

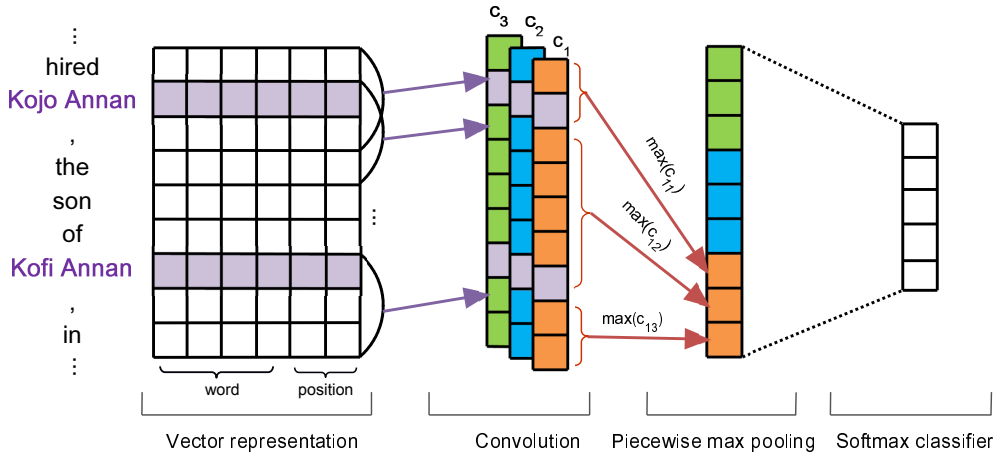


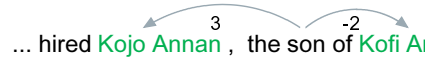

Figure 3: The architecture of PCNNs (better viewed in color) used for distant supervised relation extraction, illustrating the procedure for handling one instance of a bag and predicting the relation between *Kojo Annan* and *Kofi Annan*.

enhancing many other NLP tasks (Parikh et al., 2014; Huang et al., 2014).

A common method of training a neural network is to randomly initialize all parameters and then optimize them using an optimization algorithm. Recent research (Erhan et al., 2010) has shown that neural networks can converge to better local minima when they are initialized with word embeddings. Word embeddings are typically learned in an entirely unsupervised manner by exploiting the co-occurrence structure of words in unlabeled text. Researchers have proposed several methods of training word embeddings (Bengio et al., 2003; Collobert et al., 2011; Mikolov et al., 2013). In this paper, we use the Skip-gram model (Mikolov et al., 2013) to train word embeddings.

#### 3.1.2 Position Embeddings

In relation extraction, we focus on assigning labels to entity pairs. Similar to Zeng et al. (2014), we use PFs to specify entity pairs. A PF is defined as the combination of the relative distances from the current word to $e_1$ and $e_2$. For instance, in the following example, the relative distances from *son* to $e_1$ (*Kojo Annan*) and $e_2$ (*Kofi Annan*) are 3 and -2, respectively.




Two position embedding matrixes ($\mathbf{PF}_1$ and $\mathbf{PF}_2$) are randomly initialized. We then transform the relative distances into real valued vectors by looking up the position embedding matrixes. In the example shown in Figure 3, it is assumed that the size of the word embedding is $d_w = 4$ and that the size of the position embedding is $d_p = 1$. In combined word embeddings and position embeddings, the vector representation part transforms an instance into a matrix $\mathbf{S} \in \mathbb{R}^{s \times d}$, where $s$ is the sentence length and $d = d_w + d_p * 2$. The matrix $\mathbf{S}$ is subsequently fed into the convolution part.

### 3.2 Convolution

In relation extraction, an input sentence that is marked as containing the target entities corresponds only to a relation type; it does not predict labels for each word. Thus, it might be necessary to utilize all local features and perform this prediction globally. When using a neural network, the convolution approach is a natural means of merging all these features (Collobert et al., 2011).

Convolution is an operation between a vector of weights, $\mathbf{w}$, and a vector of inputs that is treated as a sequence $\mathbf{q}$. The weights matrix $\mathbf{w}$ is regarded as the filter for the convolution. In the example shown in Figure 3, we assume that the length of the filter is $w$ ($w = 3$); thus, $\mathbf{w} \in \mathbb{R}^m$ ($m = w * d$). We consider $\mathbf{S}$ to be a sequence $\{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_s\}$, where $\mathbf{q}_i \in \mathbb{R}^d$. In general, let $\mathbf{q}_{i:j}$ refer to the concatenation of $\mathbf{q}_i$ to $\mathbf{q}_j$. The convolution operation involves taking the dot product of $\mathbf{w}$ with each $w$-gram in the sequence $\mathbf{q}$ to obtain another sequence $\mathbf{c} \in \mathbb{R}^{s+w-1}$:

$$c_j = \mathbf{w}\mathbf{q}_{j-w+1:j} \quad (1)$$

where the index $j$ ranges from 1 to $s+w-1$. Out-of-range input values $\mathbf{q}_i$, where $i < 1$ or $i > s$, are

taken to be zero.

The ability to capture different features typically requires the use of multiple filters (or feature maps) in the convolution. Under the assumption that we use $n$ filters ($\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n\}$), the convolution operation can be expressed as follows:

$$c_{ij} = \mathbf{w}_i \mathbf{q}_{j-w+1:j} \quad 1 \leq i \leq n \tag{2}$$

The convolution result is a matrix $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_n\} \in \mathbb{R}^{n \times (s+w-1)}$. Figure 3 shows an example in which we use 3 different filters in the convolution procedure.

### 3.3 Piecewise Max Pooling

The size of the convolution output matrix $\mathbf{C} \in \mathbb{R}^{n \times (s+w-1)}$ depends on the number of tokens $s$ in the sentence that is fed into the network. To apply subsequent layers, the features that are extracted by the convolution layer must be combined such that they are independent of the sentence length. In traditional Convolution Neural Networks (CNNs), max pooling operations are often applied for this purpose (Collobert et al., 2011; Zeng et al., 2014). This type of pooling scheme naturally addresses variable sentence lengths. The idea is to capture the most significant features (with the highest values) in each feature map.

However, despite the widespread use of single max pooling, this approach is insufficient for relation extraction. As described in the first section, single max pooling reduces the size of the hidden layers too rapidly and is too coarse to capture fine-grained features for relation extraction. In addition, single max pooling is not sufficient to capture the structural information between two entities. In relation extraction, an input sentence can be divided into three segments based on the two selected entities. Therefore, we propose a piecewise max pooling procedure that returns the maximum value in each segment instead of a single maximum value. As shown in Figure 3, the output of each convolutional filter $\mathbf{c}_i$ is divided into three segments $\{\mathbf{c}_{i1}, \mathbf{c}_{i2}, \mathbf{c}_{i3}\}$ by *Kojo Annan* and *Kofi Annan*. The piecewise max pooling procedure can be expressed as follows:

$$p_{ij} = \max(\mathbf{c}_{ij}) \quad 1 \leq i \leq n, \ 1 \leq j \leq 3 \tag{3}$$

For the output of each convolutional filter, we can obtain a 3-dimensional vector $\mathbf{p}_i = \{p_{i1}, p_{i2}, p_{i3}\}$. We then concatenate all vectors $\mathbf{p}_{1:n}$ and apply a non-linear function, such as the hyperbolic tangent. Finally, the piecewise max pooling procedure outputs a vector:

$$\mathbf{g} = \tanh(\mathbf{p}_{1:n}) \tag{4}$$

where $\mathbf{g} \in \mathbb{R}^{3n}$. The size of $\mathbf{g}$ is fixed and is no longer related to the sentence length.

### 3.4 Softmax Output

To compute the confidence of each relation, the feature vector $\mathbf{g}$ is fed into a softmax classifier.

$$\mathbf{o} = \mathbf{W}_1 \mathbf{g} + b \tag{5}$$

$\mathbf{W_1} \in \mathbb{R}^{n_1 \times 3n}$ is the transformation matrix, and $\mathbf{o} \in \mathbb{R}^{n_1}$ is the final output of the network, where $n_1$ is equal to the number of possible relation types for the relation extraction system.

We employ dropout (Hinton et al., 2012) on the penultimate layer for regularization. Dropout prevents the co-adaptation of hidden units by randomly dropping out a proportion $p$ of the hidden units during forward computing. We first apply a "masking" operation $(\mathbf{g} \circ \mathbf{r})$ on $\mathbf{g}$, where $\mathbf{r}$ is a vector of Bernoulli random variables with probability $p$ of being 1. Eq.(5) becomes:

$$\mathbf{o} = \mathbf{W}_1(\mathbf{g} \circ \mathbf{r}) + b \tag{6}$$

Each output can then be interpreted as the confidence score of the corresponding relation. This score can be interpreted as a conditional probability by applying a softmax operation (see Section 3.5). In the test procedure, the learned weight vectors are scaled by $p$ such that $\hat{\mathbf{W}}_1 = p\mathbf{W}_1$ and are used (without dropout) to score unseen instances.

### 3.5 Multi-instance Learning

In order to alleviate the wrong label problem, we use multi-instance learning for PCNNs. The PCNNs-based relation extraction can be stated as a quintuple $\theta = (\mathbf{E}, \mathbf{PF}_1, \mathbf{PF}_2, \mathbf{W}, \mathbf{W}_1)$[2]. The input to the network is a bag. Suppose that there are $T$ bags $\{M_1, M_2, \cdots, M_T\}$ and that the $i$-th bag contains $q_i$ instances $M_i = \{m_i^1, m_i^2, \cdots, m_i^{q_i}\}$. The objective of multi-instance learning is to predict the labels of the unseen bags. In this paper, all instances in a bag are considered independently. Given an input instance $m_i^j$, the network with the parameter $\theta$ outputs a vector $\mathbf{o}$, where the $r$-th component $o_r$ corresponds to the score associated

[2] $\mathbf{E}$ represents the word embeddings.

**Algorithm 1** Multi-instance learning

1: Initialize $\theta$. Partition the bags into mini-batches of size $b_s$.
2: Randomly choose a mini-batch, and feed the bags into the network one by one.
3: Find the $j$-th instance $m_i^j$ $(1 \leq i \leq b_s)$ in each bag according to Eq. (9).
4: Update $\theta$ based on the gradients of $m_i^j$ $(1 \leq i \leq b_s)$ via Adadelta.
5: Repeat steps 2-4 until either convergence or the maximum number of epochs is reached.

with relation $r$. To obtain the conditional probability $p(r|m, \theta)$, we apply a softmax operation over all relation types:

$$p(r|m_i^j;\theta) = \frac{e^{o_r}}{\sum_{k=1}^{n_1} e^{o_k}} \quad (7)$$

The objective of multi-instance learning is to discriminate bags rather than instances. To do so, we must define the objective function on the bags. Given all $(T)$ training bags $(M_i, y_i)$, we can define the objective function using cross-entropy at the bag level as follows:

$$J(\theta) = \sum_{i=1}^{T} \log p(y_i|m_i^j;\theta) \quad (8)$$

where $j$ is constrained as follows:

$$j^* = \arg\max_j p(y_i|m_i^j;\theta) \;\; 1 \leq j \leq q_i \quad (9)$$

Using this defined objective function, we maximize $J(\theta)$ through stochastic gradient descent over shuffled mini-batches with the Adadelta (Zeiler, 2012) update rule. The entire training procedure is described in Algorithm 1.

From the introduction presented above, we know that the traditional backpropagation algorithm modifies a network in accordance with all training instances, whereas backpropagation with multi-instance learning modifies a network based on bags. Thus, our method captures the nature of distant supervised relation extraction, in which some training instances will inevitably be incorrectly labeled. When a trained PCNN is used for prediction, a bag is positively labeled if and only if the output of the network on at least one of its instances is assigned a positive label.

# 4 Experiments

Our experiments are intended to provide evidence that supports the following hypothesis: automatically learning features using PCNNs with multi-instance learning can lead to an increase in performance. To this end, we first introduce the dataset and evaluation metrics used. Next, we test several variants via cross-validation to determine the parameters to be used in our experiments. We then compare the performance of our method to those of several traditional methods. Finally, we evaluate the effects of piecewise max pooling and multi-instance learning[3].

## 4.1 Dataset and Evaluation Metrics

We evaluate our method on a widely used dataset[4] that was developed by (Riedel et al., 2010) and has also been used by (Hoffmann et al., 2011; Surdeanu et al., 2012). This dataset was generated by aligning Freebase relations with the NYT corpus, with sentences from the years 2005-2006 used as the training corpus and sentences from 2007 used as the testing corpus.

Following previous work (Mintz et al., 2009), we evaluate our method in two ways: the held-out evaluation and the manual evaluation. The held-out evaluation only compares the extracted relation instances against Freebase relation data and reports the precision/recall curves of the experiments. In the manual evaluation, we manually check the newly discovered relation instances that are not in Freebase.

## 4.2 Experimental Settings

### 4.2.1 Pre-trained Word Embeddings

In this paper, we use the Skip-gram model (word2vec)[5] to train the word embeddings on the NYT corpus. Word2vec first constructs a vocabulary from the training text data and then learns vector representations of the words. To obtain the embeddings of the entities, we concatenate the tokens of a entity using the ## operator when the entity has multiple word tokens. Since a comparison of the word embeddings is beyond the scope

[3] With regard to the position feature, our experiments yield the same positive results described in Zeng et al. (2014). Because the position feature is not the main contribution of this paper, we do not present the results without the position feature.

[4] http://iesl.cs.umass.edu/riedel/ecml/

[5] https://code.google.com/p/word2vec/

| *Window size* | *Feature maps* | *Word dimension* | *Position dimension* | *Batch size* | *Adadelta parameter* | *Dropout probability* |
|---|---|---|---|---|---|---|
| $w = 3$ | $n = 230$ | $d_w = 50$ | $d_p = 5$ | $b_s = 50$ | $\rho = 0.95, \varepsilon = 1e^{-6}$ | $p = 0.5$ |

Table 1: Parameters used in our experiments.

of this paper, our experiments directly utilize 50-dimensional vectors.

#### 4.2.2 Parameter Settings

In this section, we experimentally study the effects of two parameters on our models: the window size, $w$, and the number of feature maps, $n$. Following (Surdeanu et al., 2012), we tune all of the models using three-fold validation on the training set. We use a grid search to determine the optimal parameters and manually specify subsets of the parameter spaces: $w \in \{1, 2, 3, \cdots, 7\}$ and $n \in \{50, 60, \cdots, 300\}$. Table 1 shows all parameters used in the experiments. Because the position dimension has little effect on the result, we heuristically choose $d_p = 5$. The batch size is fixed to 50. We use Adadelta (Zeiler, 2012) in the update procedure; it relies on two main parameters, $\rho$ and $\varepsilon$, which do not significantly affect the performance (Zeiler, 2012). Following (Zeiler, 2012), we choose $0.95$ and $1e^{-6}$, respectively, as the values of these parameters. In the dropout operation, we randomly set the hidden unit activities to zero with a probability of 0.5 during training.

### 4.3 Comparison with Traditional Approaches

#### 4.3.1 Held-out Evaluation

The held-out evaluation provides an approximate measure of precision without requiring costly human evaluation. Half of the Freebase relations are used for testing. The relation instances discovered from the test articles are automatically compared with those in Freebase.

To evaluate the proposed method, we select the following three traditional methods for comparison. *Mintz* represents a traditional distant-supervision-based model that was proposed by (Mintz et al., 2009). *MultiR* is a multi-instance learning method that was proposed by (Hoffmann et al., 2011). *MIML* is a multi-instance multi-label model that was proposed by (Surdeanu et al., 2012). Figure 4 shows the precision-recall curves for each method, where *PCNNs+MIL* denotes our method, and demonstrates that *PCNNs+MIL* achieves higher precision over the entire range of recall. *PCNNs+MIL* enhances the recall to approximately 34% without any loss of precision. In terms of both precision and recall, *PCNNs+MIL* outperforms all other evaluated approaches. Notably, the results of the methods evaluated for comparison were obtained using manually crafted features. By contrast, our result is obtained by automatically learning features from original words. The results demonstrate that the proposed method is an effective technique for distant supervised relation extraction. Automatically learning features via PCNNs can alleviate the error propagation that occurs in traditional feature extraction. Incorporating multi-instance learning into a convolutional neural network is an effective means of addressing the wrong label problem.

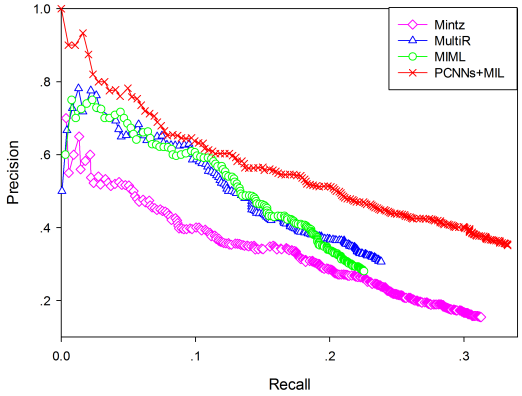



Figure 4: Performance comparison of the proposed method with traditional approaches.

| Top N | Mintz | MultiR | MIML | PCNNs+MIL |
|---|---|---|---|---|
| Top 100 | 0.77 | 0.83 | 0.85 | **0.86** |
| Top 200 | 0.71 | 0.74 | 0.75 | **0.80** |
| Top 500 | 0.55 | 0.59 | 0.61 | **0.69** |
| Average | 0.676 | 0.720 | 0.737 | **0.783** |

Table 2: Precision values for the top 100, top 200, and top 500 extracted relation instances upon manual evaluation.

#### 4.3.2 Manual Evaluation

It is worth emphasizing that there is a sharp decline in the held-out precision-recall curves of *PCNNs+MIL* at very low recall (Figure 4). A manual check of the misclassified examples that were produced with high confidence reveals that the ma-

jorities of these examples are false negatives and are actually true relation instances that were misclassified due to the incomplete nature of Freebase.

Thus, the held-out evaluation suffers from false negatives in Freebase. We perform a manual evaluation to eliminate these problems. For the manual evaluation, we choose the entity pairs for which at least one participating entity is not present in Freebase as a candidate. This means that there is no overlap between the held-out and manual candidates. Because the number of relation instances that are expressed in the test data is unknown, we cannot calculate the recall in this case. Instead, we calculate the precision of the top N extracted relation instances. Table 2 presents the manually evaluated precisions for the top 100, top 200, and top 500 extracted instances. The results show that *PCNNs+MIL* achieves the best performance; moreover, the precision is higher than in the held-out evaluation. This finding indicates that many of the false negatives that we predict are, in fact, true relational facts. The sharp decline observed in the held-out precision-recall curves is therefore reasonable.

### 4.4 Effect of Piecewise Max Pooling and Multi-instance Learning

In this paper, we develop a method of piecewise max pooling and incorporate multi-instance learning into convolutional neural networks for distant supervised relation extraction. To demonstrate the effects of these two techniques, we empirically study the performance of systems in which these techniques are not implemented through held-out evaluations (Figure 5). *CNNs* represents convolutional neural networks to which single max pooling is applied. Figure 5 shows that when piecewise max pooling is used (*PCNNs*), better results are produced than those achieved using *CNNs*. Moreover, compared with *CNNs+MIL*, *PCNNs* achieve slightly higher precision when the recall is greater than 0.08. Since the parameters for all the model are determined by grid search, we can observe that *CNNs* cannot achieve competitive results compared to *PCNNs* when increasing the size of the hidden layer of convolutional neural networks. It means that we cannot capture more useful information by simply increasing the network parameter. These results demonstrate that the proposed piecewise max pooling technique is beneficial and

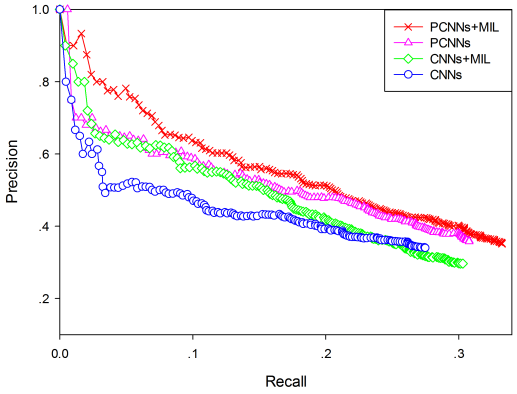



Figure 5: Effect of piecewise max pooling and multi-instance learning.

can effectively capture structural information for relation extraction. A similar phenomenon is also observed when multi-instance learning is added to the network. Both *CNNs+MIL* and *PCNNs+MIL* outperform their counterparts *CNNs* and *PCNNs*, respectively, thereby demonstrating that incorporation of multi-instance learning into our neural network was successful in solving the wrong label problem. As expected, *PCNNs+MIL* obtains the best results because the advantages of both techniques are achieved simultaneously.

## 5 Conclusion

In this paper, we exploit Piecewise Convolutional Neural Networks (PCNNs) with multi-instance learning for distant supervised relation extraction. In our method, features are automatically learned without complicated NLP preprocessing. We also successfully devise a piecewise max pooling layer in the proposed network to capture structural information and incorporate multi-instance learning to address the wrong label problem. Experimental results show that the proposed approach offers significant improvements over comparable methods.

## Acknowledgments

This work was sponsored by the National Basic Research Program of China (no. 2014CB340503) and the National Natural Science Foundation of China (no. 61272332 and no. 61202329). We thank the anonymous reviewers for their insightful comments.

## References


Yoshua Bengio, Rjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of HLT/EMNLP*, pages 724–731.

Razvan Bunescu and Raymond Mooney. 2006. Subsequence kernels for relation extraction. *Proceedings of NIPS*, 18:171–178.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Journal of Artificial intelligence*, 89(1):31–71.

Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. 2010. Why does unsupervised pretraining help deep learning? *Journal of Machine Learning Research*, 11:625–660.

Benjamin Graham. 2014. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

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 ACL*, pages 541–550.

Fei Huang, Arun Ahuja, Doug Downey, Yi Yang, Yuhong Guo, and Alexander Yates. 2014. Learning representations for weakly supervised natural language processing tasks. *Journal of Computational Linguistics*, 40(1):85–120, March.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of ACLdemo*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751.

Ryan T McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of EMNLP-CoNLL*, pages 122–131.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-AFNLP*, pages 1003–1011.

Ankur P Parikh, Shay B Cohen, and Eric P Xing. 2014. Spectral unsupervised parsing with additive tree metrics. In *Proceedings of ACL*, pages 1062–1072.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1746–1751.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of COLING*, pages 697–704.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML PKDD*, pages 148–163.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP-CoNLL*, pages 1201–1211.

Fabian M. Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of KDD*, pages 712–717.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP-CoNLL*, pages 455–465.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, abs/1212.5701.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

Minling Zhang and Zhihua Zhou. 2006. Adapting rbf neural networks to multi-instance learning. *Neural Processing Letters*, 23(1):1–26.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of ACL*, pages 825–832.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL*, pages 427–434.