Distant Supervision for Relation Extraction with Sentence-level Attention and Entity Descriptions

Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
{guoliang.ji, kliu, shizhu.he, jzhao}@nlpr.ia.ac.cn

Abstract

Distant supervision for relation extraction is an efficient method to scale relation extraction to very large corpora which contains thousands of relations. However, the existing approaches have flaws on selecting valid instances and lack of background knowledge about the entities. In this paper, we propose a sentence-level attention model to select the valid instances, which makes full use of the supervision information from knowledge bases. And we extract entity descriptions from Freebase and Wikipedia pages to supplement background knowledge for our task. The background knowledge not only provides more information for predicting relations, but also brings better entity representations for the attention module. We conduct three experiments on a widely used dataset and the experimental results show that our approach outperforms all the baseline systems significantly.

Introduction

Relation extraction (RE) under distant supervision aims to predict semantic relations between pairs of entities in texts supervised by knowledge bases (KBs). It heuristically aligns entities in texts to a given KB and uses this alignment to learn a relation extractor. The training data are labelled automatically as follows: for a triplet $r(e_1, e_2)$[1] in the KB, all sentences that mention both entities e_1 and e_2 are regarded as the training instances of relation r. Figure 1 shows the training instances of triplet */location/location/contains (Nevada, Las Vegas)*. The sentences from S1 to S4 all mention entities *Nevada* and *Las Vegas*, so they are all training instances of the relation */location/location/contains*. The task is crucial for many Natural Language Processing (NLP) applications such as automatic knowledge completion and question-answering.

Distant supervision strategy is an effective method of automatically labeling training data, however, it is plagued by the wrong label problem (Riedel, Yao, and McCallum 2010). A sentence that mentions two entities may not express the relation which links them in a KB. It is possible that the two entities may just appear in the same sentence because they

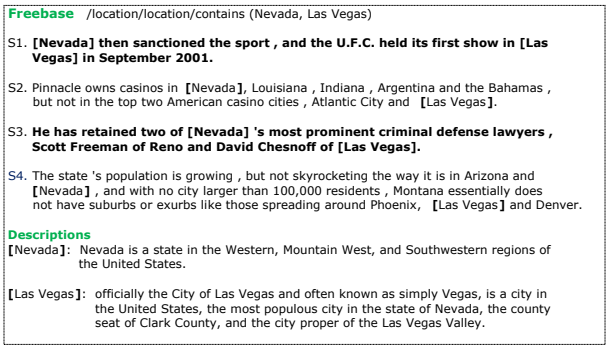


Figure 1: Training instances of the triplet */location/location/contains (Nevada, Las Vegas)*. The low part shows the descriptions of *Nevada* and *Las Vegas*.

are related to the same topic. For example, in Figure 1, sentences S2 and S4 both mention *Nevada* and *Las Vegas*, but they do not express the relation */location/location/contains*. Mintz et al., (2009) ignored the problem and extracted features from all the sentences to feed a relation classifier. Riedel, Yao, and McCallum, (2010) proposed the *expressed-at-least-once*[2] assumption, and used an undirected graphical model to predict which sentences express the relation. Based on the Multi-Instance Learning (Dietterich, Lathrop, and Lozano-Pérez 1997), Hoffmann et al., (2011) and Surdeanu et al., (2012) also used a probabilistic, graphical model to select sentences and added overlapping relations to their relation extraction systems. Zeng et al., (2015) combined multi-instance learning (MIL) and piecewise convolutional neural networks (PCNNs) to choose the most likely valid sentence and predict relations, which achieved state-of-the-art performance on the dataset developed by (Riedel, Yao, and McCallum 2010).

In multi-instance learning paradigm, for the triplet $r(e_1, e_2)$, all the sentences which mention both e_1 and e_2 constitute a bag and the relation r is the label of the bag. Although the above approaches have achieved high performance on RE under distant supervision, they have two main flaws. More specifically, (1) A bag may contain **multiple**

[1] e_1 and e_2 are entities, r is the relation between them. For example, BornIn(Barack Obama, United States).

[2] If two entities participate in a relation, **at least one sentence** that mentions these two entities might express that relation.

valid sentences. For example, in Figure 1, sentences S1 and S3 both express the relation */location/location/contains*. The probabilistic, graphical models (Riedel, Yao, and McCallum 2010; Hoffmann et al. 2011; Surdeanu et al. 2012) had considered the observation, but the features they designed to choose valid sentences are often derived from preexisting NLP tools which suffer from error propagation and accumulation (Bach and Badaskar 2007). Zeng et al., (2015) extracted sentence features by PCNNs instead of relying on the traditional NLP tools and achieved state-of-the-art performance. However, in the learning process, its MIL module only selected **one sentence** which has the maximum probability to be a valid candidate. This strategy doesn't make full use of the supervision information. Therefore, integrating the merits (considering multiple valid sentences and extracting features by neural networks) of the two approaches may be promising; (2) The **entity descriptions**, which can provide helpful background knowledge, are useful resources for our task. For example, in Figure 1, it's difficult to decide which relation the sentence S1 expresses without the information that *Nevada* is a state and *Las Vegas* is a city. When lacking the background knowledge, *Nevada* may be a government official's name and S1 doesn't express the relation */location/location/contains*. Therefore, the descriptions are beneficial for the task. Unfortunately, none of the existing work uses them for RE under distant supervision.

To select multiple valid sentences, we propose a sentence-level attention model based on PCNNs (denoted by APCNNs), which extracts sentence features using PCNNs and learns the weights of sentences by the attention module. We hope that the attention mechanism is able to selectively focus on the relevant sentences through assigning higher weights for valid sentences and lower weights for the invalid ones. In this way, APCNNs could recognize multiple valid sentences in a bag. Concretely, motivated by TransE (Bordes et al. 2013) which modeled a triplet $r(e_1, e_2)$ with $\boldsymbol{e}_1 + \boldsymbol{r} \approx \boldsymbol{e}_2$ (the bold, italic letters represent vectors), we use $(\boldsymbol{e}_1 - \boldsymbol{e}_2)$ to represent the relation between e_1 and e_2 in sentences (we will show more explanations later). For a bag, we first use PCNNs to extract each sentence's feature vector $\boldsymbol{v}_{sen}$, then compute the attention weight for each sentence through a hidden layer with the concatenation way $[\boldsymbol{v}_{sen}; \boldsymbol{e}_1 - \boldsymbol{e}_2]$ (Luong, Pham, and Manning 2015). At last, the weighted sum of all sentence feature vectors is the bag's features. In addition, to encode more background knowledge into our model, we use convolutional neural networks (CNNs) to extract entity descriptions' feature vectors and let them be close to the corresponding entity vectors via adding constraints on the objective function of APCNNs (called APCNNs+D, where "D" refers to descriptions). The background knowledge not only provides more information for predicting relations, but also brings better entity representations for the attention module.

Therefore, our main contributions in this paper are: (1)We introduce a sentence-level attention model to select multiple valid sentences in a bag. This strategy makes full use of the supervision information; (2) We use entity descriptions to provide background knowledge for predicting relations and improving entity representations; (3) We conduct experiments on a widely used dataset [3] and achieve state-of-the-art performance.

Task Definition

In multi-instance learning paradigm, all sentences labeled by a triplet constitute a bag and each sentence is called an instance. Suppose that there are N bags $\{B_1, B_2, \cdots, B_N\}$ in the training set and that the i-th bag contains q_i instances $B_i = \{b_1^i, b_2^i, \cdots, b_{q_i}^i\}$ $(i = 1, \cdots, N)$. The objective of multi-instance learning is to predict the labels of the unseen bags. We need to learn a relation extractor based on the training data and then use it to predict relations for test set. Specifically, for a bag $B_j = \{b_1^j, b_2^j, \cdots, b_{q_j}^j\}$ in training set, we need to extract features from the bag (from one or several valid instances) and then use them to train a classifier. For a bag in test set, we also need to extract features in the same way and use the classifier to predict the relation between the given entity pair.

Methodology

In this section, we present the main innovative solutions including sentence-level attention and entity descriptions. Sentence-level attention makes our model be able to select multiple valid instances for training, so that we can make full use of the supervision information. Entity descriptions provide more background knowledge about the entities, which could improve the performance of our model and bring better entity representations for attention module. Figure 2 shows the neural network architecture of our model APCNNs. It consists of two parts: PCNNs Module and Sentence-level Attention Module. PCNNs Module includes *Vector Representation, Convolution* and *Piecewise Max-pooling*. Sentence-level Attention Module is composed of *Attention Layer* and *Softmax Classifier*. We describe these parts in details below.

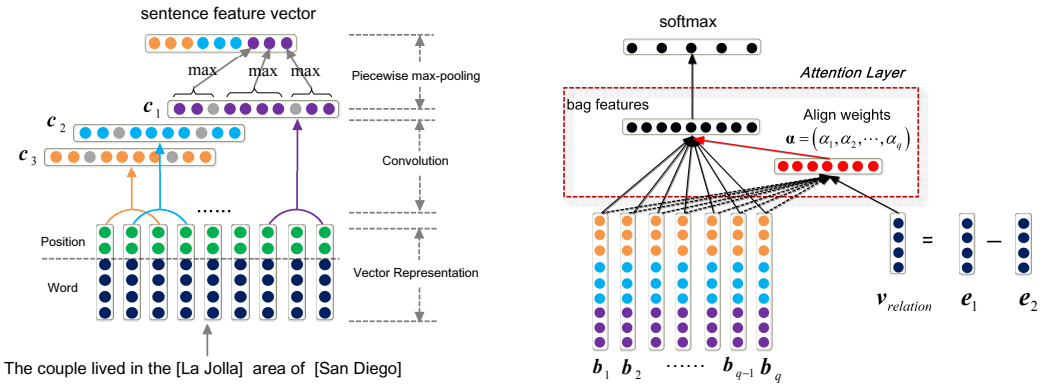
PCNNs Module

This module is used to extract feature vector of an instance (sentence) in a bag.

Vector Representation Since we use neural network model to do our task, we should transform the word tokens into low-dimensional vectors. In this paper, the "word token" refers to word and entity. In the following, we don't distinguish them and call them "word". In our method, we transform words into vectors by looking up the pre-trained word embeddings. And we use position features to specify the given entity pair (Collobert et al. 2011; Zeng et al. 2015), which also need to be transformed into vectors by looking up the position embeddings.

Word Embeddings Word embeddings are distributed representations of words that map each word in texts to a low-dimensional vector. Much work (Socher et al. 2012; 2013) has shown its power in many NLP tasks. In the past years, many methods for training word embeddings have been proposed (Bengio et al. 2006; Collobert et al. 2011;

[3] Download from `http://iesl.cs.umass.edu/riedel/ecml/`



(a) PCNNs Module (b) Sentence-level Attention Module

Figure 2: The architecture of APCNNs model. It has two parts: (a) PCNNs Module and (b) Sentence-level Attention Module. The PCNNs Module is used to extract features from an instance in a bag. The Sentence-level Attention Module computes the weighted average vector (bag features) of all instances in a bag through an attention layer, and then feed the bag features into a softmax classifier. $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_q$ are instance vectors in a bag and they are extracted by the PCNNs. e_1 and e_2 are the first and second entities, respectively. $\boldsymbol{v}_{relation}$ denotes the difference vector of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$.

Mikolov et al. 2013a). We employ the method (Mikolov et al. 2013a) to train word embeddings and denote it by $\mathbf{E}$.

Position Embeddings Zeng et al, (2014) has shown the importance of position features in RE. It is defined as the combination of the relative distances from the current word to e_1 and e_2. Each word has two relative distances. Figure 3 shows an example of the relative distance. The relative distance from word *area* to *La Jolla* and *San Diego* are 1 and -2. We randomly initialize two position embedding matrices $\mathbf{PF}_i (i = 1, 2)$ (for e_1 and e_2), and transform the relative distances into vectors by looking them up.

We concatenate the word representation and position representation as the input of the network (shown in Figure 2(a)). Assume that the size of word representation is k_w and that of position representation is k_d, then the size of a word vector is $k = k_w + 2k_d$.

Figure 3: An example of relative distances.

Convolution Assume that $\mathbf{A} = (a_{ij})_{m \times n}$ and $\mathbf{B} = (b_{ij})_{m \times n}$, then the convolution of $\mathbf{A}$ and $\mathbf{B}$ is defined as $\mathbf{A} \otimes \mathbf{B} = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij}$.

We denote the input sentence by $S = \{s_1, s_2, \cdots, s_{|S|}\}$ where s_i is the i-th word, and use $\boldsymbol{s}_i \in \mathbb{R}^k$ to represent its vector. We use $\mathbf{S}_{i:j}$ to represent the matrix concatenated by sequence $[\boldsymbol{s}_i : \boldsymbol{s}_{i+1} : \cdots : \boldsymbol{s}_j]$ ($[\boldsymbol{x}_1 : \boldsymbol{x}_2]$ denotes the horizontal concatenation of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$). We denote the length of filter by w (Figure 2(a) shows an example of $w = 3$), then the weight matrix of the filter is $\mathbf{W} \in \mathbb{R}^{w \times k}$. Then the convolution operation between the filter and sentence S results in another vector $\boldsymbol{c} \in \mathbb{R}^{|S|-w+1}$:

$$c_j = \mathbf{W} \otimes \mathbf{S}_{(j-w+1):j} \tag{1}$$

where $1 \leq j \leq |S| - w + 1$.

In experiments, we use $n (n > 1)$ filters (or feature maps) to capture different features of an instance. Therefore, we also need n weight matrices $\widehat{\mathbf{W}} = \{\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_n\}$, so that all the convolution operations can be expressed by

$$c_{ij} = \mathbf{W}_i \otimes \mathbf{S}_{(j-w+1):j} \tag{2}$$

where $1 \leq i \leq n$ and $1 \leq j \leq |S| - w + 1$. Through the convolution layer, we obtain the results vectors $\mathbf{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_n\}$.

Piecewise Max-pooling Single max-pooling operation is often used to extract the most significant features (the maximum values) in feature maps. However, in order to capture the structural information and fine-grained features, PCNNs divides an instance into three segments according to the given entity pair (two entities cut the sentence into three parts) and do max-pooling operation on each segment. For the result vector $\boldsymbol{c}_i$ of convolution operations, it can be divided into three parts $\boldsymbol{c}_i = \{\boldsymbol{c}_{i,1}, \boldsymbol{c}_{i,2}, \boldsymbol{c}_{i,3}\}$. Then piecewise max-pooling procedure is $p_{ij} = \max(\boldsymbol{c}_{i,j})$, where $1 \leq i \leq n$ and $j = 1, 2, 3$. After that, we can concatenate all the vectors $\boldsymbol{p}_i = [p_{i,1}, p_{i,2}, p_{i,3}] (i = 1, 2, \cdots, n)$ to obtain vector $\boldsymbol{p} \in \mathbb{R}^{3n}$. Figure 2(a) displays an example of $n = 3$, in which the gray circles are the positions of entities. Finally, we compute the feature vector $\boldsymbol{b}_S = \tanh(\boldsymbol{p})$ for sentence S.

Sentence-level Attention Module

Attention mechanism is one of the most important parts of the proposed approach. We hope that the attention model can learn higher weights for valid instances and lower weights for the invalid ones. In experiments, we will show the weights of an example. Once the bag features have been computed, we feed them into a softmax classifier.

Attention Layer Recently, many knowledge graph embedding approaches (Bordes et al. 2013; Wang et al. 2014; Ji et al. 2015) regarded relation as translation from head entity (e_1) to tail entity (e_2). They used $\boldsymbol{e}_1 + \boldsymbol{r} \approx \boldsymbol{e}_2$ to model the translation for triplet $r(e_1, e_2)$ and achieved state-of-the-art performance on knowledge graph completion task. Wang et al., (2014) also used the translation-based method to model the relation between two words in texts. Furthermore, Mikolov et al., (2013b) showed some properties of word embeddings such as $\boldsymbol{v}$("Madrid") $-$ $\boldsymbol{v}$("Spain") $=$ $\boldsymbol{v}$("Paris") $-$ $\boldsymbol{v}$("France") which illustrates that the difference vector of this entity pair may reflect some features of their relation. Motivated by these ideas, we utilize the difference vector to represent the features of the relation which links e_1 and e_2. Specifically, for a bag labelled by $r(e_1, e_2)$, the difference vector $\boldsymbol{v}_{relation} = \boldsymbol{e}_1 - \boldsymbol{e}_2$ contains the features of relation r. Each instance in the bag may express the relation r or another relation. If an instance expresses the relation r, its feature vector should has higher similarity with $\boldsymbol{v}_{relation}$, otherwise lower similarity.

Figure 2(b) shows the details of the attention module. In Figure 2(b), $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_q$ are feature vectors (computed by PCNNs) of all instances in a bag. And we use $\boldsymbol{v}_{relation} = \boldsymbol{e}_1 - \boldsymbol{e}_2$ to denote the relation between e_1 and e_2. We propose the following formulas (Luong, Pham, and Manning 2015) to compute the attention weight (similarity or relatedness) between each instance's feature vector and $\boldsymbol{v}_{relation}$.

$$\alpha_i = \frac{\exp(\omega_i)}{\sum_{j=1}^{q} \exp(\omega_j)} \tag{3}$$

$$\omega_i = \mathbf{W}_a^{\top}(\tanh[\boldsymbol{b}_i; \boldsymbol{v}_{relation}]) + b_a \tag{4}$$

where $[\boldsymbol{x}_1; \boldsymbol{x}_2]$ denotes the vertical concatenation of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, $1 \le i \le q$, $\mathbf{W}_a \in \mathbb{R}^{1\times(3n+k_w)}$ is an intermediate matrix and b_a is an offset value. $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_q]$ are the weight vector of all instances in the bag. Then the bag features can be computed as follows.

$$\bar{\boldsymbol{b}} = \sum_{i=1}^{q} \alpha_i \boldsymbol{b}_i \tag{5}$$

where $\bar{\boldsymbol{b}} \in \mathbb{R}^{3n}$.

Softmax To compute the confidence of each relation, we feed the feature vector $\bar{\boldsymbol{b}}$ into a softmax classifier.

$$\mathbf{o} = \mathbf{W}_s \bar{\boldsymbol{b}} + b_s \tag{6}$$

where $\mathbf{o} \in \mathbb{R}^{n_o}$ is the output, $\mathbf{W}_s \in \mathbb{R}^{n_o \times 3n}$ is the weight matrix and $b_s \in \mathbb{R}^{n_o}$ is the bias. Let $\theta = (\mathbf{E}, \widehat{\mathbf{W}}, \mathbf{PF}_1, \mathbf{PF}_2, \mathbf{W}_a, \mathbf{W}_s)$[4] to denote all parameters and B represent a bag. Then the conditional probability of i-th relation is

$$p(r_i|B;\theta) = \frac{\exp(o_i)}{\sum_{j=1}^{n_o} \exp(o_j)} \tag{7}$$

Entity Descriptions

Entity descriptions can provide rich background knowledge for entities. Here, we use another traditional CNN (a convolution layer and a single max-pooling layer) to extract features from entity descriptions. We denote the set of (entity, description) pairs by $\mathcal{D} = \{(e_i, d_i)|i = 1, \cdots, |\mathcal{D}|\}$. The vectors of $\boldsymbol{e}_i$ and that of words in descriptions can be obtained by looking up the word embeddings $\mathbf{E}$. The vectors of $\boldsymbol{d}_i$ are computed by a CNNs whose weight matrices are denoted by $\widehat{\mathbf{W}}_d$. In our method, we let the vectors of entities be close to that of descriptions. Therefore, we define the errors between them as follows.

$$\mathcal{L}_e = \sum_{i=1}^{|\mathcal{D}|} \parallel \boldsymbol{e}_i - \boldsymbol{d}_i \parallel_2^2 \tag{8}$$

The background knowledge extracted from descriptions not only provides more information for prediction relations, but also brings better representations of entities for the attention module. We extract descriptions for entities from Freebase and Wikipedia pages. In Freebase, there are 25,271 entities which have unique descriptions. The other 14,257 entities have no descriptions in Freebase, we extract their descriptions on Wikipedia pages. For the latter, there are 3,197 (8.1%) entity descriptions containing string " may refer to" which means that they are ambiguous, so we do not use them. We extract the first 80 words for descriptions.

Training Objective

Assume that there are N bags in training set $\{B_1, B_2, \cdots, B_N\}$, and their labels are relations $\{r_1, r_2, \cdots, r_N\}$. To analyze the effects of attention mechanism and entity descriptions respectively, we train our model in three settings.

First, we train the APCNNs which only contains the sentence-level attention module (no entity descriptions). Then we define the objective function using cross-entropy as follows.

$$\min \mathcal{L}_A = -\sum_{i=1}^{N} \log p(r_i|B_i, \theta) \tag{9}$$

where $\theta = (\mathbf{E}, \widehat{\mathbf{W}}, \mathbf{PF}_1, \mathbf{PF}_2, \mathbf{W}_a, \mathbf{W}_s)$.

Second, we train the model APCNNs+D which contains both sentence-level attention module and entity descriptions. The objective function is

$$\min \mathcal{L} = \mathcal{L}_A + \lambda \mathcal{L}_e \tag{10}$$

where $\lambda > 0$ is the weight of $\mathcal{L}_e$.

Beyond that, we enforce the $\mathcal{L}_e$ as a constraint on the objective function of PCNNs+MIL (Zeng et al. 2015), and report its performance. We denote the setting as PCNNs+MIL+D. The difference between APCNNs and PCNNs+MIL is that the former use sentence-level attention module to replace the later's MIL module which only selects one instance during training.

In experiments, we adopt dropout strategy (Hinton et al. 2012) and ADADELTA (Zeiler 2012) to train our models.

Experiments

In this section, we first introduce the dataset and evaluation metrics,then show the experimental results and analysis.

Dataset

We evaluate our approach using the dataset developed by (Riedel, Yao, and McCallum 2010) by aligning Freebase[5]

[4] We neglect the biases.

[5] freebase.com

relations with the New York Times (NYT) corpus. The training data is aligned to the years 2005-2006 of the NYT corpus, and the testing to the year 2007. This dataset also has been used by (Hoffmann et al. 2011; Surdeanu et al. 2012; Zeng et al. 2015). Its entities are annotated with Stanford NER and linked to Freebase. The dataset contains 52 relations (including no relation "NA") and 39,528 entities. We train word embeddings on the NYT corpus with `word2vec`[6] and use the embeddings as initial values .

Evaluation Metrics

Following the previous work (Riedel, Yao, and McCallum 2010; Zeng et al. 2015), we evaluate our approach in two ways: Held-out evaluation and Human evaluation. The former only compares the relation instances extracted from bags against Freebase relations data (bags' labels) automatically. Noting the fact that distant supervision could produce some wrong labels due to the incomplete nature of Freebase, we use a human evaluation to manually check the newly discovered relation instances that are not in Freebase. Following (Riedel, Yao, and McCallum 2010; Zeng et al. 2015), we conduct a manual evaluation by choosing the entity pairs for which at least one participating entity is not presented in Freebase as a candidate. It means that the objects of our manual evaluation are the bags which are labelled by "NA" in corpus, but our model predict a relation (not "NA") for each of them with a high confidence. For held-out evaluation, we present the precision/recall curves in experiments. For manual evaluation, we can not calculate recall because we cannot provide all the relation instances expressed in all bags. Hence, we report the precision of top-k bags with high confidence produced by our methods.

Experimental Results and Analysis

In this section, we show the settings of parameters, experimental results and comparisons with previous baselines.

Parameter Settings In our experiments, we tune all of the models using three-fold validation on the training set. We select the dimension of word embedding k_w among $\{50, 100, 200, 300\}$, the dimension of position embedding k_d among $\{5, 10, 20\}$, the windows size w among $\{3, 5, 7\}$, the number of feature maps n among $\{50, 100, 150, 200, 230\}$, the weight λ among $\{0.001, 0.01, 0.1, 1.2\}$, batch size among $\{50, 100, 150, 200\}$. The best configurations are: $k_w = 50$, $k_d = 5$, $w = 3$, $n = 200$, $\lambda = 0.01$, the batch size is 50. Following (Hinton et al. 2012), we set the dropout rate 0.5.

Result and Comparisons We compare our method with four previous work. *Mintz* is proposed by (Mintz et al. 2009) which extracted features from all instances; *MultiR* is a multi-instance learning method proposed by (Hoffmann et al. 2011); *MIML* is a multi-instance multi-labels method proposed by (Surdeanu et al. 2012); *PCNNs+MIL* is the state-of-the-art method proposed by (Zeng et al. 2015).

Held-out Evaluation Figure 4 displays the aggregate precision/recall curves of our approach and all the baselines. From Figure 4, we can see that our models (PCNNs+MIL+D, APCNNs, and APCNNs+D) outperform all the baseline systems, especially the APCNNs+D improves the results significantly. First, our models can achieve the recall by 0.37, which is higher than 0.34 of PCNNs+MIL. Second, on the entire range of recall, our models achieve higher precision than all the baselines. As APCNNs achieves better performance than all baselines including PCNNs+MIL, we can conclude that attention mechanism can use more supervision information than other models. PCNNs+MIL+D also outperforms PCNNs+MIL, which shows that entity descriptions can provide background knowledge to improve the prediction accuracies. Finally, APCNNs+D achieves state-of-the-art performance. It demonstrates that the combination of attention and entity descriptions is beneficial for our task.

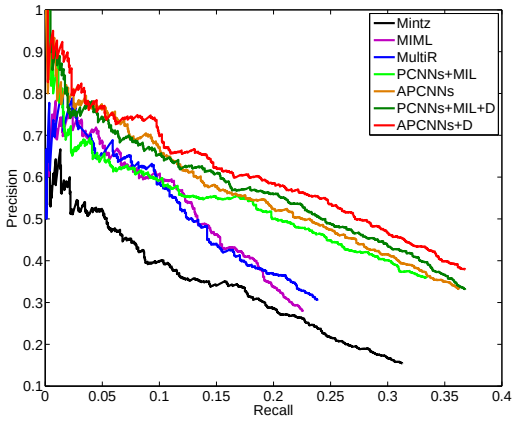


Figure 4: Aggregate extraction precision/recall curves for a variety of methods. For the sake of clarity, we show all the curves with different colors and bold lines.

Manual Evaluation From Figure 4, we can see that there is a sharp decline in the held-out precision-recall curves of the most models at very low recall. That's because the held-out evaluation suffers from false negative in Freebase. Our manual evaluation can eliminate the problems. We conduct the evaluation by three PhD students whose research directions are Natural Language Processing. Table 2 shows the precisions of our manual evaluation on top-100, top-200, top-500 extracted relation instances. From Table 2, we can see that: (1) APCNNs obtains better prediction accuracies than PCNNs+MIL, which shows that the attention module can select more valid instances; (2) PCNNs+MIL+D also outperforms PCNNs+MIL, which proves that the entity descriptions provide more useful background information; and (3) APCNNs+D achieves state-of-the-art performance, so the attention module and entity descriptions are both useful.

Analysis of Weight α Table 1 shows an example of Weight α of a bag. The bag contains five instances in which the 4-th instance are invalid sentence. Our models assign it lower weights (0.09 and 0.073). The remaining instances are valid because they all contain some significant keywords about the relation. Our attention module also assigns them higher weights. Therefore, the attention mechanism can select the valid instances and is useful in our task. As shown

[6] `https:code.google.com/p/word2vec/`

Bag Label	Instances	APCNNs	APCNNs+D
/location/location/contains (New Orleans, Dillard University)	1. She graduated from [Dillard University] in [New Orleans] and received a master's degree in marine science from the College of William and Mary.	0.223	0.239
	2. Jinx Broussard, a communications professor at [Dillard University] in [New Orleans], said four members of her family had lost their houses to the hurricanes.	0.216	0.235
	3. I was grieving from the death when I graduated from high school, but I decided to go to [Dillard University] in [New Orleans].	0.268	0.242
	4. When he came here in May 2003 to pick up an honorary degree from [Dillard University], his dense schedule didn't stop him from calling Dooky Chase's, the Creole restaurant he sang about in "Early in the Morning Blues, "where he'd eaten his favorite dish ever since he lived in [New Orleans] in the 1950's."	0.090	0.073
	5. He is waiting because of his involvement with a group that oversees a partnership between the University of Colorado and [Dillard University] in [New Orleans]."	0.203	0.211

Table 1: An example of Weight $\boldsymbol{\alpha}$. The red strings are significant keywords to predict the relation.

Accuracy (%)	Top 100	Top 200	Top 500	Average
Mintz	0.77	0.71	0.55	0.676
MultiR	0.83	0.74	0.59	0.720
MIML	0.85	0.75	0.61	0.737
PCNNs+MIL	0.86	0.80	0.69	0.783
APCNNs	**0.87**	0.82	0.72	0.803
PCNNs+MIL+D	0.86	0.82	0.71	0.797
APCNNs+D	**0.87**	**0.83**	**0.74**	**0.813**

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

before, attention module relies on the entity representations severely and entity descriptions can bring better representations for entities. Hence, we argue that the entity descriptions could enhance the performance of the attention module. In Table 1, for the 4-th instance in the bag, we can see that its weight computed by APCNNs is 0.09 which is higher than 0.073 computed by APCNNs+D. Obviously, the attention module recognizes invalid/valid instances better with the help of entity descriptions. Therefore, the background knowledge provided by entity descriptions can improve the performance of attention module.

Related Work

The approaches about relation extraction (RE) can be roughly divided into two groups: supervised and distant supervised methods.

Supervised Methods

Much work on relation extraction has focused on fully-supervised approaches, and they regard the task as a multi-class classification problem. GuoDong et al., (2005) explored a set of features (lexical and syntactic) that are selected by performing textual analysis, then they converted the features into symbolic IDs and feed them into a SVM classifier. Other work (Bunescu and Mooney 2005; Mooney and Bunescu 2005; Zelenko, Aone, and Richardella 2003) used kernel methods (such as subsequence kernel and dependency tree kernel) for the task, which requires pre-processed the input data with NLP tools. These approaches are effective. Conversely, Zeng et al., (2014) exploited a convolutional deep neural network (CNN) to extract lexical and sentence level features. Based on the CNN model, dos Santos, Xiang, and Zhou, (2015) proposed a Classification by Ranking CNN (CR-CNN) model. These methods have achieved high precision and recall. Unfortunately, they need explicitly human annotated texts, which makes them unlikely to scale to the large text corpus.

Distant Supervised Methods

Distant supervision methods for relation extraction heuristically align texts to the given KB and use the alignment to learn a relation extractor. They regard the large amounts of structured data sources (such as Freebase) as the weak supervision information. Since these methods do not need a hand-labeled dataset and KBs grow fast recently, they have appealed much attention. Mintz et al., (2009) extracted features from all sentences and then feed them into a classifier, which neglected the data noise and would learn some invalid instances. Riedel, Yao, and McCallum, (2010), Hoffmann et al., (2011) and Surdeanu et al., (2012) used graphical model to select the valid sentences and prediction relations. Nguyen and Moschitti, (2011) utilized relation definitions and Wikipedia documents to improve their systems. These methods extracted sentence features relying on traditional NLP tools. Zeng et al., (2015) used PCNNs to automatically learn sentence level features and considered the structure information of entity positions. But its MIL module could only select one valid sentence in training process, which doesn't make full use of the supervision information. Lin et al., (2016) proposed to use attention to select informative sentences. Our work has two innovations as compared with it: (1) Lin et al., (2016) initialized the embedding of relation $\boldsymbol{r}$ as a parameter in models, and our work uses $\boldsymbol{r} = \boldsymbol{e}_1 - \boldsymbol{e}_2$ to represent the relation, where $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are embeddings of the two given entities; and (2) The descriptions in our model provide more background knowledge for the RE task and improve the entity representations for the attention module. Another line is to introduce external semantic repositories such as knowledge graphs (Weston et al. 2013; Wang et al. 2014). These work did relation extraction experiments by connecting knowledge graphs and texts.

Conclusions and Future Work

We introduce a sentence-level attention model and entity descriptions to extract relations from texts under distant supervision. The attention mechanism can select multiple valid instances in a bag by assigning higher weights for valid instances and lower weights for the invalid ones. The entity descriptions can provide more background knowledge to predict relations and improve entity representations for

the attention module. We conduct experiments on a widely used dataset and our models outperform all the baseline systems.We observe that some other work about knowledge graphs, such as TransE (Bordes et al. 2013), can provide more meaningful representations of entities. In the future, we will explore to combine our method with them.

Acknowledgements

This work was supported by the Natural Science Foundation of China (No. 61533018), the National Basic Research Program of China (No. 2014CB340503) and the National Natural Science Foundation of China (No. 61272332).

References

Bach, N., and Badaskar, S. 2007. A review of relation extraction. *Literature review for Language and Statistics II*.

Bengio, Y.; Schwenk, H.; Senécal, J.-S.; Morin, F.; and Gauvain, J.-L. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*. Springer. 137–186.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, 2787–2795.

Bunescu, R. C., and Mooney, R. J. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, 724–731. Association for Computational Linguistics.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

Dietterich, T. G.; Lathrop, R. H.; and Lozano-Pérez, T. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence* 89(1):31–71.

dos Santos, C. N.; Xiang, B.; and Zhou, B. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, 626–634.

GuoDong, Z.; Jian, S.; Jie, Z.; and Min, Z. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, 427–434. Association for Computational Linguistics.

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 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*, 541–550. Association for Computational Linguistics.

Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL*, 687–696.

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 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*, 1003–1011. Association for Computational Linguistics.

Mooney, R. J., and Bunescu, R. C. 2005. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, 171–178.

Nguyen, T.-V. T., and Moschitti, A. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, 277–282. Association for Computational Linguistics.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 148–163.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1201–1211. Association for Computational Linguistics.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, 926–934.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 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*, 455–465. Association for Computational Linguistics.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph and text jointly embedding. In *EMNLP*, 1591–1601. Citeseer.

Weston, J.; Bordes, A.; Yakhnenko, O.; and Usunier, N. 2013. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zelenko, D.; Aone, C.; and Richardella, A. 2003. Kernel methods for relation extraction. *The Journal of Machine Learning Research* 3:1083–1106.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J.; et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, 2335–2344.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. EMNLP.