LETTER Communicated by Zhi Jin

A Customized Attention-Based Long Short-Term Memory Network for Distant Supervised Relation Extraction

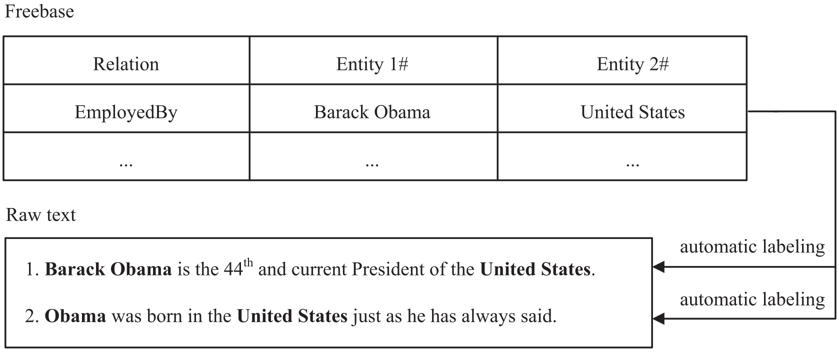
Dengchao He
hdchao1989@163.com
Hongjun Zhang
jsnjzhanghongjun@163.com
Wenning Hao
jsnjhwnbox@163.com
Rui Zhang
jsnjzhangrui@163.com
Kai Cheng
jsnjchengkai@163.com
College of Command Information System, PLA University of Science and Technology, Nan Jing, 210007, P.R.C.

Distant supervision, a widely applied approach in the field of relation extraction can automatically generate large amounts of labeled training corpus with minimal manual effort. However, the labeled training corpus may have many false-positive data, which would hurt the performance of relation extraction. Moreover, in traditional feature-based distant supervised approaches, extraction models adopt human design features with natural language processing. It may also cause poor performance. To address these two shortcomings, we propose a customized attention-based long short-term memory network. Our approach adopts word-level attention to achieve better data representation for relation extraction without manually designed features to perform distant supervision instead of fully supervised relation extraction, and it utilizes instance-level attention to tackle the problem of false-positive data. Experimental results demonstrate that our proposed approach is effective and achieves better performance than traditional methods.

1 Introduction

Relation extraction, defined as the task of extracting semantic relations between a pair of entities expressed in sentences, has received increasing interest. It can play a significant role in various natural language processing (NLP) tasks, for example, information extraction (Banko et al., 2007; Wu & Weld, 2010), question answering (Iyyer, Boyd-Graber, Claudino, Socher, & Daumé, 2014), knowledge-based construction (Suchanek, Fan, Hoffmann, Riedel, & Talukdar, 2013), and ontology learning (Wong, Liu, & Bennamoun,

Neural Computation **27, 1964–1985** (2017) © 2017 Massachusetts Institute of Technology
doi:10.1162/NECO_a_00970



Relation	Entity 1#	Entity 2#
EmployedBy	Barack Obama	United States
...	...	...

1. **Barack Obama** is the 44th and current President of the **United States**.
2. **Obama** was born in the **United States** just as he has always said.

Figure 1: Automatic labeling by distant supervision. The first sentence is a correct labeling instance, and the second sentence is an incorrect labeling instance.

2012), for example. Most approaches to relation extraction use supervised learning of relation-specific instances to achieve high precision and recall. However, traditional fully supervised approaches are unlikely to apply to the extracting large amount of relations found on the Web and are limited by the availability of large amounts of labeled training data (Hoffmann, Zhang, Ling, Zettlemoyer, & Weld, 2011).

Due to the deficiencies of supervised approaches, a more promising approach distant supervised relation extraction, has gained attention; it automatically creates its own training data by heuristically aligning facts in knowledge bases to texts. The idea of distant supervision for relation extraction assumes that if two entities have a relationship in a known knowledge base, then all sentences that contain this pair of entities will express the relationship in some way (Mintz, Bills, Snow, & Jurafsky, 2009). Figure 1 shows a simple example of distant supervised relation extraction. In the figure, *Barack Obama* and *United States* are related entities in Freebase,[1] and we use $r(e1, e2)$ =*EmployedBy*(*Barack Obama*, *United States*) to indicate the relationship. All sentences with synonyms for both entities, $e1 =$ *Barack Obama* and $e2 =$ *United States*, are considered to be an expression of the fact that $r(e1, e2)$ holds and are selected as positive training instances.

The strategy of distant supervision is an effective approach for automatically labeling training data and a sound solution to the problem of the availability of big data. However, it has two major shortcomings that restrict the effectiveness of relation extraction. First, the original distant supervision assumption proposed by Mintz et al. (2009) is too strong and may cause an incorrect labeling problem, that is, it may generate false-positive instances (Riedel, Yao, & McCallum, 2010). The reason is that an instance

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

containing a pair of entities does not necessarily express the relation that is in a knowledge base. It is possible because the entities may appear in the same instance and share the same topic (Zeng, Liu, Chen, & Zhao, 2015). For instance, consider the *EmployedBy* relation between *Barack Obama* and *United States* in Figure 1. The first instance indeed expresses the *EmployedBy* relation between the two entities. According to the distant supervision assumption, *EmployedBy* is assigned to the instance, which becomes a useful training instance. But the second instance does not express the *EmployedBy* relation between two entities. However, the distant supervised relation extraction heuristic labels the sentence as expressing an *EmployedBy* relation and selects the sentence as a positive training example. These false-positive instances would hurt the performance of a relation extraction model trained on such noisy data (Hoffmann et al., 2011).

Moreover, research has typically applied traditional machine learning models to exquisitely designed features with labeled training data obtained through distant supervision (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011). These features are derived from existing NLP tools, such as *openNLP*[2] and Stanford coreNLP.[3] Nevertheless, these NLP tools would generate inevitable errors. Using these tools to obtain traditional designed features leads to error propagation or accumulation (Zeng et al., 2015). Hence, much of the literature (Zeng, Liu, Lai, Zhou, & Zhao, 2014; Zeng et al., 2015; Xu et al., 2015; Lin, Shen, Liu, Luan, & Sun, 2016) adopts deep neural network models to avoid artificially designed features. In the distant supervised relation extraction domain, multi-instance learning is integrated into a deep neural network model, which assumes that if two entities participate in a relation, at least one sentence that mentions these two entities might express that relation (Hoffmann et al., 2011; Surdeanu, Tibshirani, Nallapati, & Manning, 2012). In practice, these integrated models are trained by selecting the most likely instance for each entity pair, so large amounts of information in those neglected instances are omitted.

In this letter, we propose a novel long short-term memory (LSTM) network integrated with a customized attention mechanism to address the deficiencies we have described. We use several LSTM networks with shared parameters to acquire the semantic representations of instances that would automatically extract semantic and syntactic features of instances without artificial design and complicated NLP processing (see Figure 2 in section 4). Each LSTM network adopts word-level attention to achieve better instance representation. Thereafter, inspired by Zeng et al. (2015), we consider the relation as a composition of instance semantic embeddings and merge all the embeddings of positive instances to represent the relation. Furthermore, we treat distant supervised relation extraction as a multi-instance learning

[2] http://sourceforge.net/projects/opennlp/.
[3] http://github.com/stanfordnlp/CoreNLP.

problem just as other works to (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Zeng et al., 2015) do to alleviate the problem of wrong labels. Our approach integrates instance-level attention into LSTM networks, which measure the relevance between the instance embedding and the relation. According to the relevance, instance-level attention allocates different weights to instances. Due to the benefits of attention mechanism, the uncertainty of instance labels is taken into account; the weights of false-positive instances to learn the relation vector would be reduced dynamically.

To evaluate the effectiveness of our approach, we applied our model on a benchmark data set developed by Riedel et al. (2010). The experimental results indicate that our approach exhibits superior performance compared to conventional methods.

The contributions of our work can be summarized as follows:

- We propose an LSTM-based architecture to perform distant supervision for relation extraction to avoid artificially designed features. Our model is supposed to automatically learn and extract features without complicated NLP preprocessing.
- To acquire more informative instance embeddings for relation extraction, we introduce word-level attention into our LSTM-based architecture.
- To figure out the incorrect labeling problem, we incorporate an instance-level attention mechanism into our LSTM-based networks to minimize the impact of false-positives. To the best of our knowledge, we are the first to use LSTM-based recurrent neural networks with an attention mechanism for distant supervision in the relation extraction domain.
- In our model, we use all the semantic information of instances containing the same entity pair, which is proved to be beneficial to distant supervised relation extraction.

We review related work in section 2. The formal description of distant supervised relation extraction is introduced in section 3. In section 4, we describe our LSTM-based model with instance-level attention in detail. Section 5 presents quantitative experimental results. We conclude in section 6.

2 Related Work

Relation extraction is a widely studied topic in the NLP community. Various supervised learning methods for relation extraction have been proposed. In these methods, relation extraction is considered to be a multiclass classification problem that can achieve high precision and recall (Zelenko, Aone, & Richardella, 2003; Zhou, Zhang, Ji, & Zhu, 2007; Kambhatla, 2004). However, supervision learning methods may suffer from a lack of labeled training data and are unlikely to scale to the thousands of relations found in text on the web. To address this deficiency of supervised-learning methods, Mintz et al. (2009) introduced distant supervision for relation extraction, adopting Freebase to heuristically label a textual corpus, which would generate false-positive instances inevitably. To solve incorrect labeling, Riedel et al. (2010), Hoffman et al. (2011), and Surdeanu et al. (2012) relax the original distant supervision assumption; if two entities participate in a relation, at least one sentence that mentions these two entities might express that relation and present a series of models casting distant supervision. The relaxed assumption converts distant supervised relation extraction into a multi-instance learning problem (Dietterich, Lathrop, & Lozano-Pérez, 1997). Other work has focused on filtering noisy data in heuristically labeled training data generated by distant supervision. Takamatsu, Sato, and Nakagawa (2012) propose a generative model of the labeling process to improve the quality of labels before training a relation extraction model as a preprocessing step. Xu, Hoffmann, Zhao, and Grishman (2013) analyze a random sample from the *New York Times*, indicating that a large number of entity pairs express a relation Freebase defined as corresponding to false-negative instances. They adopt pseudo-relevance feedback to add missing entries in the knowledge base before training MultiR (Hoffmann et al., 2011).

Most existing approaches have concentrated on designing features to classify the relations between two entities, mainly into two classes: feature-based methods (Kambhatla, 2004; Suchanek, Ifrim, & Weikum, 2006) and kernel-based methods (Qian et al., 2008; Bunescu & Mooney, 2005a, 2005b). In feature-based methods, different sets of features are extracted, selected, and fed into a classifier. Generally three type of features are used: lexical features (e.g., POS), syntactic features (e.g., parsing tree), and semantic features (entity type, entity mention) (Kambhatla, 2004). Feature-based methods may suffer from the difficulty of selecting an appropriate set of features when transforming structured representations into feature vectors. Kernel-based approaches specify the measure of similarity between two instances without explicit feature representation. Several kernels have been proposed, such as convolution tree kernel (Qian, Zhou, Kong, Zhu, & Qian, 2008), subsequence kernel (Bunescu & Mooney, 2005b) and dependency tree kernel (Bunescu & Mooney, 2005a). The potential deficiency of kernel-based methods is that all data information is completely measured by the kernel function designed, so devising an effective kernel becomes crucial. The performance of their model relies heavily on the quality of the artificially designed features and leads to error propagation problems.

Deep neural networks are attracting growing attention (Zeng et al., 2014, 2015; Lin et al., 2016; Xu et al., 2015), which can learn underlying features automatically. Socher, Pennington, Huang, Ng, and Manning (2011) propose a recursive neural network with a parse tree of sentences for classifying relations between entities. Hashimoto, Miwa, Tsuruoka, and Chikayama, (2013) improve the performance by weighting phrases' importance in recursive neural networks. Ebrahimi and Dou (2015) adopt the dependency path between two indicated entities to build a recurrent neural network model (RNN). Xu et al. (2015) input the shortest dependency path instead of whole path into their LSTM-based RNN model and achieve a better result. Other neural networks have been used for relation extraction. Zeng et al. (2014) explore convolution neural networks (CNN) using sequential information of sentences, and Santos, Xiang, and Zhou (2015) propose a ranking loss function to train their CNN model.

Although deep neural network–based models work well in relation extraction, they suffer from the incorrect labeling problem as well. Hence, Zeng et al. (2015) integrate multi-instance learning into CNN to perform distant supervised relation extraction. In practice, they select the most positive instance for each entity pair, which inevitably loses the rich semantic information contained in those omitted instances. Lin et al. (2016) build sentence-level attention over multiple instances in CNN and achieve better performance. Inspired by Lin et al. (2016), we propose useing LSTM-based networks with a customized attention mechanism to automatically learn features for distant supervised relation extraction.

3 Distant Supervised Relation Extraction

In this section, we concentrate on the related concept of distant supervised relation extraction. A *relation fact* is defined as an expression $r(e1, e2)$, where r is a relation name (e.g., *Employed* in Figure 1), and $e1$ and $e2$ are two entity names (e.g., *Barack Obama* and *United States*).

An *entity mention* is a contiguous sequence of text tokens denoting an entity name. In this letter, we assume all the entity mentions in a corpus are extracted by a different process, such as a named entity recognizer.

A *relation instance* is a sequence of text with two or more entity mentions, which indicates that a certain relation fact $r(e1, e2)$ is true. In this letter, we focus only on the extraction of binary relations expressed in a single sentence.

Distant supervision for relation extraction utilizes a knowledge base to create labeled data for relation extraction by heuristically matching entity pairs (Takamatsu et al., 2012). A knowledge base is a set of relation instances about predefined relation names. For each sentence in the corpus, we extract all of its entity pairs. Then for each entity pair, we try to retrieve the relation facts about the entity pair from the knowledge base. If we find such a relation fact, then the set of the entity pair and the sentence is stored as a positive instance. Otherwise, the set of the entity pair and the sentence is stored as a negative instance. Training data are automatically constructed through this procedure.

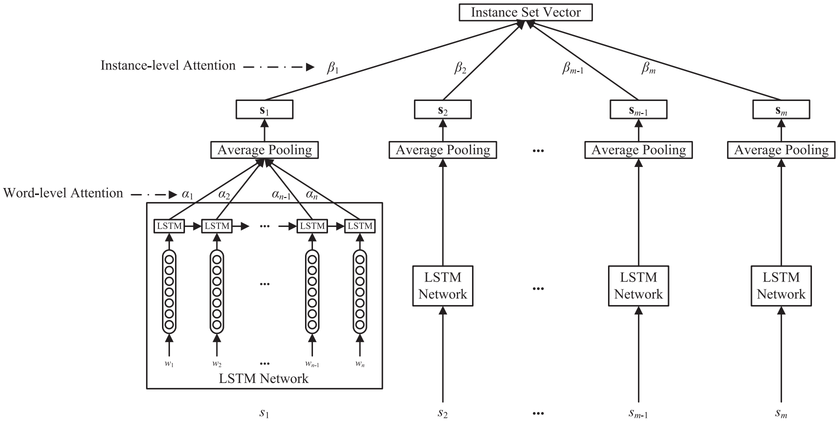


Figure 2: The overall architecture of ATT-LSTM.

4 Methodology

In this section, we describe our LSTM-based network incorporating a customized attention mechanism to fulfill distant supervision for relation extraction, which is formulated as a multi-instance problem. Section 4.1 presents the overall architecture of our model. Section 4.2 discusses the rationale of adopting word embeddings and entity identifiers as our input. LSTM units are explained in section 4.3. We explain the customized attention mechanism in section 4.4. Finally, we introduce the output of our model in section 4.5. We abbreviate the proposed model as ATT-LSTM (LSTM with attention mechanism).

4.1 Overall Architecture. Figure 2 presents the overall architecture of our ATT-LSTM network. Given a set of instances $\{S_1, S_2, \ldots, S_m\}$, which contains the same entity pair, our ATT-LSTM model predicts the relation that the two entities have. In the input layer, word tokens in each instance are mapped to low-dimensional distributed representation, which are real-valued vectors called word embeddings (Collobert et al., 2011; Mikolov, Chen, Corrado, & Dean, 2013) and could capture the underlying syntactic and semantic information of the inputs. Then the recurrent neural networks are used to extract implicit features and information from each instance that adopts LSTM units for effective information propagation. Next, an average pooling layer is applied to obtain the instance representation. After that, the instance-level attention mechanism is incorporated, which gives different weights to different instances to obtain the representation of the instance set. In the end, a softmax output layer is used to predict the relation name according to the representation of the instance set.

4.2 Vector Representation. Raw word tokens in each instance are mapped into low-dimensional vectors (Collobert et al., 2011; Mikolov et al., 2013; Zeng et al., 2014; Santos et al., 2015) when using the neural network model. Every input word token is transformed into a real-valued vector by looking up pretrained word embeddings that are trained on large-scale corpuses. Moreover, we introduce entity identifiers to indicate the specific position of each entity in instances.

4.2.1 Word Embeddings. Word embeddings are distributed representations of words that map every word in a word sequence into a dense, low-dimensional, and real-valued vector through a lookup operation (Mikolov et al., 2013; Pennington, Socher, & Manning, 2014). Each dimension of word embeddings expresses a latent feature of words. Such distributed representations of words have been verified to significantly capture the syntactic and semantic information (Mikolov et al., 2013; Pennington et al., 2014). Using been trained word embeddings has become common practice for enhancing many other NLP tasks (Huang et al., 2014; Parikh, Cohen, & Xing, 2014).

Word embeddings are encoded by column vectors in an embedding matrix $\mathbf{W} \in R^{d \times |V|}$, where d is the dimension of the word embedding and $|V|$ is the size of vocabulary. Each column $\mathbf{W}_i \in R^d$ corresponds to the word embedding of the ith word in the vocabulary. We transform a word w into its word embedding $\mathbf{w}$ by using the matrix-vector product,

$$\mathbf{w} = \mathbf{W} \cdot \mathbf{v}^w, \tag{4.1}$$

where $\mathbf{v}^w$ is a vector of size $|V|$ that has a value of 1 at index w and zero in all other positions. The matrix $\mathbf{W}$ is a parameter to be learned, and the size of the word embeddings d is a hyperparameter to be chosen.

4.2.2 Entity Identifier. Since relation extraction is defined on the basis of entity pair, it is necessary to obtain the position information of entities in instances. Conventional methods adopt a position feature to acquire such information. Zeng et al. (2014) concatenate the position features and word embeddings, which reflect the relative distances of the current word to the two entities. However, there are some deficiencies. On one hand, some entities may have more than one word token, so it would be difficult to calculate the distances. On the other hand, the position features rely on the variable length of instances in a certain corpus, which means it would not be used directly in other data sets. Hence, we introduce an entity identifier to offer position information about two entities in instances as Qin, Xu, and Guo (2016) did. For example, we add an identifier in the instances, which is demonstrated as follow:

$$< \text{e1} > \text{Obama} < /\text{e1} > \text{was born in the} < \text{e2} > \text{United States} < /\text{e2} >$$

just as he has always said.

As this example shows, < e1 >, < /e1 >, < e2 >, and < /e2 > are four entity identifiers to mark out two entities, which would be considered part of instances and be trained as word embeddings. The learned embeddings of four identifiers are irrelevant to the length of instances and word numbers of entities.

4.3 Long Short-Term Memory Units in Recurrent Neural Network.

RNN models are promising deep learning models that can represent phrases of arbitrary length in a vector space of a fixed dimension (Socher et al., 2011; Hashimoto et al., 2013; Ebrahimi & Dou, 2015). RNNs are suitable for modeling sequential data such as text and speech because they keep a hidden state vector h that changes with input data at each step. We use RNN to extract semantic features and information about instances in our approach. Although RNN performs effectively in many NLP tasks, it is generally difficult to learn the long-term dependency within the sequence due to the gradient vanishing problem. RNNs use the gradient backpropagation algorithm to train networks and update parameters. If the propagation path is too long, the gradient either blows up or decays exponentially. One of the effective solutions for this problem in RNN is to adopt LSTM units instead of neurons.

Hochreiter (1998) proposed and extended LSTM architecture. The main mechanism of LSTM is to adopt memory blocks to decide the degree of information that the LSTM unit should keep and memorize. In recent years, many LSTM variants have been proposed. In our work, we adopt a variant version of LSTM that Zaremba and Sutskever (2014) introduced. More specifically, an LSTM unit consists of one recurrent memory cell and three multiplicative blocks—that is, an input gate i_t, an output gate o_t, and a forget gate f_t—that provide continuous analogues of writing, reading, and resetting operations for the cell.

Figure 3 depicts one LSTM unit. Concretely, the LSTM unit at the ith word of the input depends on the previous state $\mathbf{h}_{t-1}$, the current input $\mathbf{w}_t$, and the memory cell $\mathbf{c}_{t-1}$. New vectors are calculated using the following equations:

$$\mathbf{i}_t = \sigma(\mathbf{W}^{(i)} \cdot \mathbf{w}_t + \mathbf{U}^{(i)} \cdot \mathbf{h}_{t-1} + \mathbf{b}^{(i)}), \tag{4.2}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}^{(f)} \cdot \mathbf{w}_t + \mathbf{U}^{(f)} \cdot \mathbf{h}_{t-1} + \mathbf{b}^{(f)}), \tag{4.3}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}^{(o)} \cdot \mathbf{w}_t + \mathbf{U}^{(o)} \cdot \mathbf{h}_{t-1} + \mathbf{b}^{(o)}), \tag{4.4}$$

$$\mathbf{u}_t = \tanh(\mathbf{W}^{(u)} \cdot \mathbf{w}_t + \mathbf{U}^{(u)} \cdot \mathbf{h}_{t-1} + \mathbf{b}^{(u)}), \tag{4.5}$$

$$\mathbf{c}_t = \mathbf{i}_t \otimes \mathbf{u}_t + \mathbf{f}_t \otimes \mathbf{c}_{t-1}. \tag{4.6}$$

where σ denotes the logistic function, $\otimes$ denotes element-wise multiplication, $\mathbf{W}$ and $\mathbf{U}$ are weight matrices, and $\mathbf{b}$ are bias vectors. The output of the

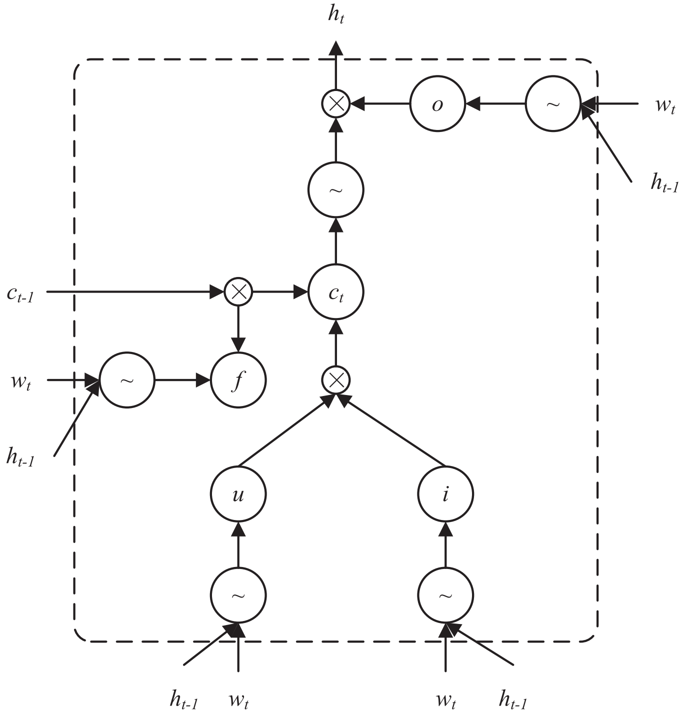


Figure 3: Inner structure of a long short term memory unit.

LSTM unit is the hidden state of recurrent networks, which is computed by equation 4.7 and is passed to the subsequent units:

$$\mathbf{h}_t = \mathbf{o}_t \otimes \tanh(\mathbf{c}_t). \tag{4.7}$$

Then an average-pooling operator runs over all the LSTM units to obtain the instance representation $\mathbf{s}_i$.

4.4 Customized Attention Mechanism. We introduce a customized attention mechanism to capture the significant semantic representations for each instance and selective true positive instances for distant supervised relation extraction based on the relation label. ATT-LSTM uses word-level attention to achieve informative instance representations for relation extraction and adopts instance-level attention to degrade the influence of false-positive instances. Detailed implementations are introduced next.

4.4.1 Word-Level Attention. In relation extraction task, not all word tokens contribute equally in an instance to the semantic relation information of an entity pair. Therefore, instead of feeding the hidden states of each LSTM unit to the average-pooling layer directly, we employ a word-level attention mechanism to figure out which words are more significant for our purpose—the semantic relation information. The word-level attention would dynamically focus on the words in instances that are more significant. In consequence, an improved instance representation is acquired through a weighted sum of the hidden states of each LSTM unit.

Suppose that given an instance s containing n word tokens, $s = \{w_1, w_2, \ldots, w_n\}$, and every word token w_i (including entity identifiers) is mapped to a real-valued vector (i.e., word embeddings). Then word embeddings are passed to LSTM units respectively to get hidden states $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$. We can obtain the representation of instance through the following equation,

$$\mathbf{s} = \frac{1}{n}\sum_{i=1}^{n} \alpha_i \mathbf{h}_i, \tag{4.8}$$

where α_i is the word-level attention weight, defined as

$$\alpha_i = \frac{\exp(e(\mathbf{h}_i, \mathbf{r}))}{\sum\limits_k \exp(e(\mathbf{h}_k, \mathbf{r}))}. \tag{4.9}$$

In equation 4.9, $e(\cdot)$ is a measure function that reflects the relevance between each word and the corresponding relation r of the entity pair in the instance. Inspired by Chen, Sun, Tu, Lin, and Liu (2016), $e(\cdot)$ is defined as

$$e(\mathbf{h}_i, \mathbf{r}) = \mathbf{v}^{\mathrm{T}} \tanh(\mathbf{W}_H \cdot \mathbf{h}_i + \mathbf{W}_R \cdot \mathbf{r} + \mathbf{b}_H), \tag{4.10}$$

where $\mathbf{W}_H$ and $\mathbf{W}_R$ are weight matrices, $\mathbf{b}_H$ is a bias vector, $\mathbf{v}$ is a weight vector, and $\mathbf{r}$ is the representation of relation r contained in the instance, which would be learned in the training phrase. According to equations 4.8 to 4.10, a word-level attention mechanism would give more weight to the informative words, which are more relevant to the relation label.

4.4.2 Instance-Level Attention. After getting all the representations of instances, we use them to acquire the representation of the instance set of a certain entity pair. According to Lin et al. (2016), the semantic information of set S would rely on the representations of all the instances, each of which contains information about whether the entity pair holds the relation. However, due to the existing false-positive instances in distant supervision for relation extraction, if we make each instance contribute equally to the relation label, an incorrect label would degrade the performance of our model.

Thus, we should calculate the weighted sum of instances contained in set S to obtain the set vector representation.

Suppose a given set S consisting of m instances contains the same entity pair, $S = \{s_1, s_2, \ldots, s_m\}$. Then the representation of S is defined as

$$\mathbf{S} = \sum_{k=1}^{m} \beta_k \mathbf{s}_k, \tag{4.11}$$

where $\mathbf{s}_k$ is the instance representation and β_k is the instance-level attention weight. Due to the incorrect labeling problem in distant supervised relation extraction, we should degrade the weight of false-positive instances to reduce the influence of noisy data. For this purpose, in our ATT-LSTM model, we calculate β_k as

$$\beta_k = \frac{\exp(\Gamma(\mathbf{s}_k, \mathbf{r}))}{\sum_l \exp(\Gamma(\mathbf{s}_l, \mathbf{r}))}, \tag{4.12}$$

where $\Gamma(\cdot)$ is a measure function that reflects the relevance between each instance and corresponding relation r. $\Gamma(\cdot)$ is defined as

$$\Gamma(\mathbf{s}_k, \mathbf{r}) = \mathbf{M}^{\mathrm{T}} \tanh(\mathbf{W}_S \cdot \mathbf{s}_k + \mathbf{W}_R \cdot \mathbf{r} + \mathbf{b}_S), \tag{4.13}$$

where $\mathbf{W}_S$ and $\mathbf{W}_R$ are weight matrices, $\mathbf{b}_{\mathrm{S}}$ is a bias vector, $\mathbf{M}$ is a weight vector, and $\mathbf{r}$ is the representation of relation r similar to equation 4.13. Through equations 4.11 to 4.13, an instance-level attention mechanism would measure the relevance between the instance embedding and the relation r. According to the relevance, the instance-level attention mechanism allocates different weights to different instances. More specifically, an instance-level attention mechanism gives true positive instances more weight and allocates less weight to wrong labeling instances to alleviate the impact of noisy data.

4.5 Output. The output layer determines the relation label of an input instance set through a softmax function. For each instance set, the set representation $\mathbf{S}$ is fed into the softmax layer to compute the confidence of each relation,

$$\mathbf{o} = \mathbf{R}\mathbf{S} + \mathbf{b}, \tag{4.14}$$

where $\mathbf{R}$ is the representation matrix of relations, $\mathbf{b}$ is a bias vector, and $\mathbf{o}$ is the final output of our ATT-LSTM model, which computes the conditional probability of each relation r:

$$p(r|S) = \frac{\exp(o_r)}{\sum\limits_k \exp(o_k)}. \tag{4.15}$$

In the training phrase, we adopt an objective function using cross-entropy at the entity pair level (Zeng et al., 2015; Lin et al., 2016). Suppose that N is the number of entity pairs (i.e., instance sets) and θ is the set of all the parameters of the ATT-LSTM model. Then the objection function is defined as

$$J(\theta) = \sum_{i=1}^{N} \log p(r_i|S_i; \theta). \tag{4.16}$$

We then adopt stochastic gradient descent over shuffled minibatches with the Adadelta (Zeiler, 2012) update rule to learn parameters.

To alleviate the overfitting problem and reduce the training time, we adopt the dropout strategy proposed by Hinton, Srivastava, Krizhevsky, Sutskever, and Salakhutdinov (2012) in the output layer. A dropout strategy can generate more robust network units and achieve better performance by dropping out neural units randomly during the training phrase. Given an instance set vector **S** and a probability of omitting the unit p, which means each dimension in the vector **S** is set to zero with probability p, the output of our model is computed as

$$\mathbf{o} = \mathbf{R} \cdot D(\mathbf{S}) + \mathbf{b}, \tag{4.17}$$

where $D()$ denotes the dropout operator.

5 Experiments

In order to verify that our proposed ATT-LSTM model is effective in automatically learning features and can reduce of incorrect labels instances using a customized attention mechanism, we carried out extensive experiments. Section 5.1 introduces the data set we adopted. Section 5.2 describes our experimental hyperparameter settings and evaluation metrics. In section 5.3, we compare ATT-LSTM's performance with several traditional feature-based methods and state-of-the-art neural network approaches.

5.1 Experimental Data Set. Following Riedel et al. (2010), Hoffmann et al. (2011), Zeng et al. (2015), and Lin et al. (2016), we evaluate our method on the same data set in our experiments developed by Riedel et al. (2010).[4] This data set was generated by aligning Freebase relations with the *New York*

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

Table 1: Hyperparameters in the Experiments.

Word Embedding Dimension	Instance Embedding Dimension	Dropout Probability	Batch Size	Adadelta Parameter ε	Adadelta Parameter ρ
50	200	0.5	160	$1e^{-6}$	0.95

Times corpus (NYT). Freebase is an online database that stores facts about entities and their relations. The *New York Times* corpus contains over 1.8 million articles written and published by the newspaper between January 1, 1987, and June 19, 2007. In Riedel's data, sentences from the years 2005 to 2006 are adopted as the training corpus, and sentences from 2007 are used as the testing corpus. There are 53 relations that include a label "NA," which means there is no relation between two entities. The training corpus contains 570,088 instances and 292,497 entity pairs. The testing corpus consists of 172,448 instances and 96,678 entity pairs.

5.2 Experimental Settings

5.2.1 Parameter Settings. This section describes the hyperparameter tuning for our approach. We use a threefold validation to tune all the hyperparameters on the training corpus following previous work. Table 1 shows the hyperparameters used in the experiments. We set word embeddings to be 50-dimensional and the instance embedding size is 200-dimensional. In the training phrase, the batch size is fixed to 160. Following Zeiler (2012), two main parameters that Adadelta relies on, ε and ρ, are set to be $1e^{-6}$ and 0.95, respectively.

5.2.2 Pretrained Word Embeddings. The word embeddings used in our experiments are 50-dimensional and initialized by means of unsupervised pretraining. We use the Skip-gram neural network architecture available in the word2vec tool designed by Mikolov et al. (2013).[5] To ensure our pretrained word embeddings are large scale and effective, we adopt the December 2013 snapshot of the English Wikipedia corpus to train word embeddings with word2vec.[6] Then we continually train the word embeddings on Riedel's corpus. Following Santos et al. (2015), we preprocess the training corpus using the following steps:

- Remove paragraphs that are not in English.
- Substitute non-Western characters for a special character.

[5] http://code.google.com/p/word2vec/.
[6] http://en.wikipedia.org/.

- Remove sentences with fewer than 20 characters, including spaces.
- Lowercase all words, and substitute each numerical digit by a zero.

5.2.3 Evaluation Metrics. Like Riedel et al. (2010), we adopt held-out evaluation to assess our ATT-LSTM model in distant supervised relation extraction. In held-out evaluation, the extracted relation instances are compared only with Freebase facts, and the precision and recall curves of the experiments are shown to indicate whether our ATT-LSTM model is effective.

5.3 Experimental Results and Discussions

5.3.1 Baselines. We compare our approach against two neural network approaches and three traditional feature-based methods to evaluate the efficiency of our proposed ATT-LSTM model.

Traditional Feature-Based Methods

- Mintz: This is an original model for distant supervised relation extraction proposed by Mintz et al. (2009).
- MultiR: This is a DS for a relation extraction model based on multi-instance learning, which was developed by Hoffmann et al. (2011). MultiR transforms relation extraction into a multi-instance problem, but learns using a perceptron algorithm and uses a deterministic "at-least-one" assumption instead of a relation classifier.
- MIML: This multi-instance multilabel model for distant supervised relation extraction was proposed by Surdeanu et al. (2012). It models the latent assignment of labels to instances and dependencies between labels assigned to the same entity pair.

Neural Network Approaches

- PCNN with MIL: This improved CNN model proposed by Zeng (2015) is integrated with multi-instance learning and adopts only one instance in each instance set to train.
- LSTM with MIL: This uses a regular LSTM network model to encode instance representations and employ multi-instance learning to reduce false-positive instances.
- CNN with ATT: This approach, proposed by Lin et al. (2016), incorporates multi-instance learning with a CNN model via instance-level attention to alleviate incorrect labeling.

5.3.2 Results Analysis and Evaluation. Following Mintz et al. (2009), Riedel et al. (2010), Hoffmann et al. (2011), and Surdeanu et al. (2012), we use held-out evaluation to demonstrate whether our ATT-LSTM model is effective. This evaluation provides an approximate measure of precision without

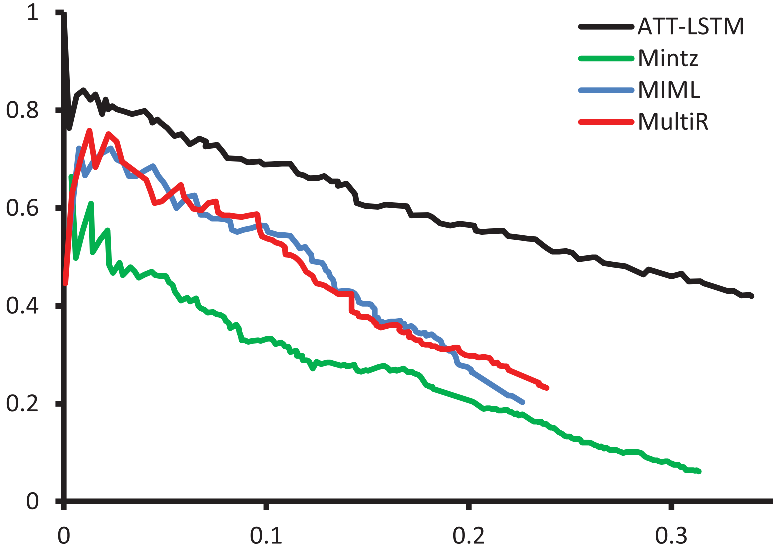


Figure 4: Comparison of precision and recall curve for the Riedel data set of ATT-LSTM and three feature-based baseline methods.

manual evaluating. The relation facts discovered from the test instances are automatically compared with those in Freebase.

Comparison with feature-based methods. In Figure 4 we compare the precision and recall curves for the three baseline models—Mintz, MultiR, and MIML—and our proposed ATT-LSTM model. The curve is constructed by ranking the predicted relation instances using their log-linear score. From Figure 4, we can see that our approach consistently outperforms the three baseline methods by achieving higher precision over the entire range of recall. It is worth noting that the three baseline models are using artificially designed features. However, our approach learns semantic representations of instances automatically without human intervention, which avoids error propagation caused by NLP tools. The results of the experiments indicate that ATT-LSTM is effective for distant supervised relation extraction and that instance representations obtained by ATT-LSTM can degrade the influence of error propagation that occurs in traditional feature engineering. Moreover, the customized attention mechanism adopted by ATT-LSTM is a sound and valid solution to reduce the influence of false-positive instances in distant supervised relation extraction.

Comparison with neural network approaches. To evaluate the effect of our customized attention mechanism, we compare our ATT-LSTM to three

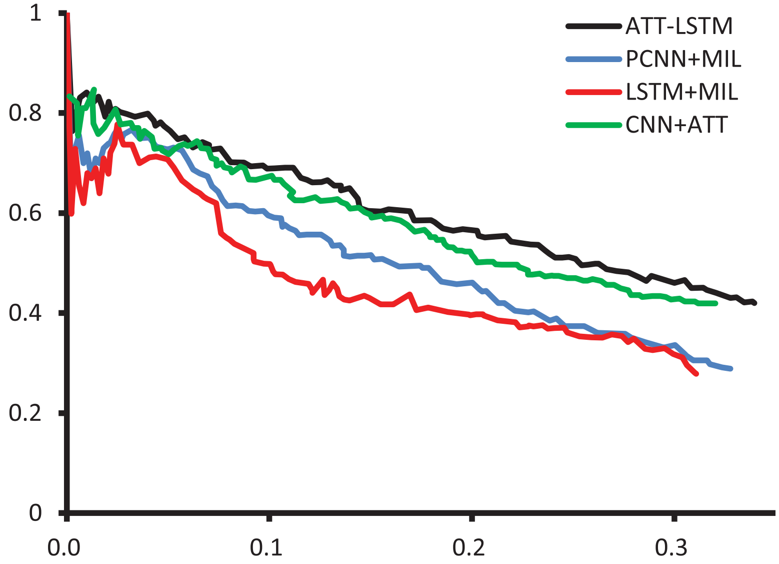


Figure 5: Comparison of precision and recall curve for the Riedel data set of ATT-LSTM and three neural network baseline methods.

neural network approaches. The PCNN model proposed by Zeng et al. (2015) and a regular LSTM network model are trained as multi-instance learning: they select only the most confident instance in the instance set. Moreover we chose a CNN model with instance-level attention proposed by Lin et al. (2016) to demonstrate the effectiveness of our LSTM model integrated with word-level attention. The results are demonstrated in Figure 5.

Our ATT-LSTM model achieves the best performance in the three approaches. More specifically, it obtains higher precision over almost the entire range of recall compared to PCNN with MIL and LSTM with MIL. The reason is that although the training model in multi-instance learning is able to reduce the false-positive instances, this method would lose a large amount of semantic information from omitted instances. By contrast, the proposed customized attention mechanism could take all of the instances into consideration and reduce the negative effects of false-positive instances through degrading their weight. Compared with CNN with ATT, ATT-LSTM obtains better performance for the most part, which indicates that as a feature extractor for relation extraction, LSTM integrated with word-level attention is effective and would achieve better performance than traditional CNN model. The reason is that the word-level attention would dynamically focus on the more significant words in instances to the purpose relation.

Table 2: Instance-Level Attention Weights of Samples in the Testing Set.

Instance	Attention Weight
Relation: ***Place_of_birth***	
#1 **Ernst Haefliger**, a Swiss tenor who was most renowned as an interpreter of German art song and oratorio roles, died on Saturday in **Davos**, Switzerland, where he maintained a second home.	0.1569
#2 **Ernst Haefliger** was born in **Davos** on July 6, 1919, and studied at the Wettinger Seminary and the Zurich conservatory before moving to Vienna, where he became a student of the tenor Julius Patzak.	0.8431
Relation: ***Founders***	
It appears to be seeking only one big fish: **Stephen A. Schwarzman**, the chief executive of the **Blackstone group**.	0.0347
The proposal clearly rankled a co-founder of the **Blackstone group**, **Stephen A. Schwarzman**, who was speaking at an awards presentation during a Yale-sponsored conference at the New York stock exchange yesterday.	0.1581

Note: The entity mentions are in bold.

The example of instance-level attention weights. To evaluate our instance-level attention mechanism more intuitively, we selected two samples of entity pairs from the testing set. In each sample, we display our attention weights of instances. The detailed content is in Table 2.

From Table 2, we see that in the first sample of the relation fact, *Place_of_birth*(*Ernst haefliger, Davos*), which contains only two instances, the former instance with a low attention weight, 0.1569, is not expressing the relation. The next instance, with a high weight, 0.8431, conveys that *Ernst_haefliger* was born in *Davos*. In the second sample, the relation fact *Founders*(*Blackstone group, Stephen A. Schwarzman*), which contains 11 instances, we choose the 2 instances with the lowest weight and highest weight, respectively. The instance with the lowest weight expresses only that *Stephen A. Schwarzman* works at the *Blackstone group*, while the highest one conveys that *Stephen A. Schwarzman* is a cofounder of the *Blackstone group*. From these examples, it appears that our instance-level attention mechanism makes true positive instances contribute more for the purpose relation and effectively figures out the incorrect labeling problem.

6 Conclusion

In this letter, we propose a novel neural network model for distant supervised relation extraction, ATT-LSTM, which automatically realizes learning features from data and makes full use of all instances' information. We incorporate a customized attention mechanism into our ATT-LSTM.

ATT-LSTM adopts word-level attention to achieve better instance representation for the relation extraction task to perform distant supervision instead of full supervision and uses instance-level attention to tackle the problem of false-positive instances in distant supervision for relation extraction. Experimental results indicate that our proposed approach outperforms three neural network approaches and traditional feature-based methods and could alleviate the incorrect labeling problem in distant supervised relation extraction.

Acknowledgments

We thank members of the Military Data and Knowledge Engineering Lab and the anonymous reviewers for their comprehensive feedback on all the research described in this letter. This work has been supported by the National Natural Science Foundation of Jiangsu Province (grant BK20150720).

References

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of the International Joint Conference on Artificial Intelligence* (vol. 7, pp. 2670–2676). Menlo Park, CA: AAAI Press.

Bunescu, R. C., & Mooney, R. J. (2005a). A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 724–731). Stroudsburg, PA: Association for Computational Linguistics.

Bunescu, R., & Mooney, R. J. (2005b). Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems, 18* (pp. 171–178). Cambridge, MA: MIT Press.

Chen, H., Sun, M., Tu, C., Lin, Y., & Liu, Z. (2016). Neural sentiment classification with user and product attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA: Association for Computational Linguistics.

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

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

Ebrahimi, J., & Dou, D. (2015). Chain based RNN for relation classification. In *Proceedings of the Chapter of the Association for Computational Linguistics* (pp. 1244–1249). Stroudsburg PA: Association for Computational Linguistics.

Hashimoto, K., Miwa, M., Tsuruoka, Y., & Chikayama, T. (2013). Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1372–1376). Stroudsburg, PA: Association for Computational Linguistics.

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

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*, 107–116.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., & 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—Vol. 1* (pp. 541–550). Stroudsburg, PA: Association for Computational Linguistics.

Huang, F., Ahuja, A., Downey, D., Yang, Y., Guo, Y., & Yates, A. (2014). Learning representations for weakly supervised natural language processing tasks. *Computational Linguistics*, *40*(1), 85–120.

Iyyer, M., Boyd-Graber, J. L., Claudino, L. M. B., Socher, R., & Daumé III, H. (2014). A neural network for factoid question answering over paragraphs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 633–644). Stroudsburg, PA: Association for Computational Linguistics.

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions* (p. 22). Stroudsburg, PA: Association for Computational Linguistics.

Lin, Y., Shen, S., Liu, Z., Luan, H., & Sun, M. (2016). Neural relation extraction with selective attention over instances. In *Proceedings of the Association for Computational Linguistics* (vol. 1, pp. 2124–2133). Stroudsburg, PA: Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv:1301.3781.

Mintz, M., Bills, S., Snow, R., & 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* (vol. 2, pp. 1003–1011). Stroudsburg, PA: Association for Computational Linguistics.

Parikh, A. P., Cohen, S. B., & Xing, E. P. (2014). Spectral unsupervised parsing with additive tree metrics. In *Proceedings of the Association for Computational Linguistics* (pp. 1062–1072). Stroudsburg, PA: Association for Computational Linguistics.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (vol. 1, pp. 1532–1543). Stroudsburg, PA: Association for Computational Linguistics.

Qian, L., Zhou, G., Kong, F., Zhu, Q., & Qian, P. (2008). Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics* (vol. 1, pp. 697–704). Stroudsburg, PA: Association for Computational Linguistics.

Qin, P., Xu, W., & Guo, J. (2016). An empirical convolutional neural network approach for semantic relation classification. *Neurocomputing*, *190*, 1–9.

Riedel, S., Yao, L., & McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 148–163). Berlin: Springer.

Santos, C. N. D., Xiang, B., & Zhou, B. (2015). *Classifying relations by ranking with convolutional neural networks*. arXiv:1504.06580.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 151–161). Stroudsburg, PA: Association for Computational Linguistics.

Suchanek, F. M., Ifrim, G., & Weikum, G. (2006). Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 712–717). New York: ACM.

Suchanek, F., Fan, J., Hoffmann, R., Riedel, S., & Talukdar, P. P. (2013, March). Advances in automated knowledge base construction. *SIGMOD Records Journal*.

Surdeanu, M., Tibshirani, J., Nallapati, R., & 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* (pp. 455–465). Stroudsburg, PA: Association for Computational Linguistics.

Takamatsu, S., Sato, I., & Nakagawa, H. (2012). Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers* (vol. 1, pp. 721–729). Stroudsburg, PA: Association for Computational Linguistics.

Wong, W., Liu, W., & Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Computing Surveys*, *44*(4), 20.

Wu, F., & Weld, D. S. (2010). Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 118–127). Stroudsburg, PA: Association for Computational Linguistics.

Xu, W., Hoffmann, R., Zhao, L., & Grishman, R. (2013). Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the Association for Computational Linguistics* (vol. 2, pp. 665–670). Stroudsburg, PA: Association for Computational Linguistics.

Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z. (2015). Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1785–1794). Stroudsburg, PA: Association for Computational Linguistics.

Zaremba, W., & Sutskever, I. (2014). *Learning to execute*. arXiv:1410.4615.

Zeiler, M. D. (2012). *ADADELTA: An adaptive learning rate method*. arXiv:1212.5701.

Zelenko, D., Aone, C., & Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*, *3*, 1083–1106.

Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of the International Conference on Computational Linguistics* (pp. 2335–2344). Stroudsburg, PA: Association for Computational Linguistics.

Zeng, D., Liu, K., Chen, Y., & Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1753–1762). Stroudsburg, PA: Association for Computational Linguistics.

Zhou, G., Zhang, M., Ji, D. H., & Zhu, Q. (2007). Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 728–736). Stroudsburg, PA: Association for Computational Linguistics.

Received November 21, 2016; accepted February 13, 2017.