Neurocomputing 000 (2017) 1–8

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Joint entity and relation extraction based on a hybrid neural network

Suncong Zheng[a], Yuexing Hao[a], Dongyuan Lu[b], Hongyun Bao[a,*], Jiaming Xu[a], Hongwei Hao[a], Bo Xu[a,c]

[a] Digital Content Technology Research Center, Institute of Automation, Chinese Academy of Sciences, China
[b] The School of Information Technology and Management, University of International Business and Economics, China
[c] Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 15 July 2016
Revised 20 December 2016
Accepted 25 December 2016
Available online xxx

Keywords:
Neural network
Information extraction
Tagging
Classification

ABSTRACT

Entity and relation extraction is a task that combines detecting entity mentions and recognizing entities' semantic relationships from unstructured text. We propose a hybrid neural network model to extract entities and their relationships without any handcrafted features. The hybrid neural network contains a novel bidirectional encoder-decoder LSTM module (BiLSTM-ED) for entity extraction and a CNN module for relation classification. The contextual information of entities obtained in BiLSTM-ED further pass though to CNN module to improve the relation classification. We conduct experiments on the public dataset ACE05 (Automatic Content Extraction program) to verify the effectiveness of our method. The method we proposed achieves the state-of-the-art results on entity and relation extraction task.

1. Introduction

Entity and relation extraction is to detect entity mentions and recognize their semantic relationships from text. It is an important issue in knowledge extraction and plays a vital role in automatic construction of knowledge base.

Traditional systems treat this task as a pipeline of two separated tasks, i.e., named entity recognition (NER) [1] and relation classification (RC) [2]. This separated framework makes the task easy to deal with, and each component can be more flexible. But it pays little attention to the relevance of two sub-tasks. Joint learning framework is an effective approach to correlate NER and RC, which can also avoid cascading of errors [3]. However, most existing joint methods are feature-based structured systems [3–7]. They need complicated feature engineering and heavily rely on the supervised NLP toolkits, which might also lead to error propagation. In order to reduce the manual work in feature extraction, recently, Miwa and Bansal [8] present a neural network-based method for the end-to-end entity and relation extraction. However, when detecting the entity, they use a NN structure to predict the entity tags, which neglects the long relationships between tags.

Based on the above analysis, we propose a hybrid neural network model to settle these problems which contains a named entity recognition (NER) module and a relation classification (RC) module. NER and RC share a same bidirectional LSTM encoding layer, which is used to encode each input word by taking into account the context on both sides of the word. Although bidirectional LSTM can capture long distance interactions between words, each output entity tag is predicted independently. Hence, we also adopt a LSTM structure to explicitly model tag interactions. It can capture the long distance relationships between tags when comparing with NN decoding manner [8]. As for relation classification, sub-sentence between two entities has been proven to effectively reflect the entities relationship [9,10]. Besides, bidirectional LSTM encoding layer can obtain entities' contextual information that is also benefit for identifying relationships between entities. Hence, we adopt a CNN model, which has achieved great success in extracting relations, to extract relation based on the encoding information of entities and the sub-sentence information.

Our model not only considers the relevance of NER module and RC module when comparing with classical pipeline methods, but also considers the long distance relationships between entity tags and without complicated feature engineering, when comparing with existing joint learning methods. We conduct experiments on the public dataset ACE05 (Automatic Content Extraction program)[1]. Our method achieves the state-of-the-art results on entity and relation extraction task. Besides, we also analyze the performance of the two modules alone. On the entity detection task, our

* Corresponding author.
E-mail address: hongyun.bao@ia.ac.cn (H. Bao).

[1] http://www.itl.nist.gov/iad/mig//tests/ace/

http://dx.doi.org/10.1016/j.neucom.2016.12.075
0925-2312/

NER module achieves 2% improvements when comparing with different kinds of LSTM structures, which verifies the effectiveness of NER module. On the task of relation classification, it shows that the entities' contextual information, obtained at the encoding procedure, can promote the accuracy of relation classification.

The remainder of the paper is structured as follows. In Section 2, we review related work about named entity recognition, relation classification and neural networks used in this paper. Section 3 present our hybrid neural network in detail. In Section 4, we describe details about the setup of experiment and presents the experimental results. Finally, we analyze the model in Section 5 and make conclusion in Section 6.

2. Related works

Entity and relation extraction is an important step to construct a knowledge base, which can be benefit for many NLP tasks [11] and social media analysis tasks [12,13]. There are two main frameworks to solve the problem of extracting entity and their relationships: the pipeline method and the joint learning model. The pipeline method treats this task as a pipeline of two separated tasks, i.e., named entity recognition (NER) [14–17] and relation classification (RC) [2,9,10,18,19]. The joint model extracts entities and relations simultaneously. Hence, in this paper, the problem we focused is related to named entity recognition, relation classification and joint entity and relation extraction. The methods we used are related to long short term memory networks (LSTM) and convolutional neural network (CNN).

2.1. Named entity recognition

Named entity recognition is a classic task in NLP. Most existing NER models are traditional linear statistical models, such as Hidden Markov Models (HMM) and Conditional Random Fields (CRF) [14,20]. Their performances rely heavily on hand-crafted features extracted by NLP tools and external knowledge resources. Recently, several neural network architectures have been successfully applied to NER, which is regarded as a sequential token tagging task. Collobert et al. [21] used a CNN and a CRF on top with word embeddings. Nowadays, Recurrent Neural Networks (RNN) has shown better performance than other neural networks in many sequence-to-sequence tasks. Chiu and Nichols [15] proposed a hybrid model by learning both character-level and word-level features. They decoded each tag independently base on a linear layer and a log-softmax layer. [16,17,22] proposed a BiLSTM and a CRF on top for jointly tag decoding. Miwa and Bansal [8] proposed a BiLSTM for encoding and a single incrementally NN structure to decode tags jointly. These RNN models all utilized BiLSTM as encoding models, but the decoding manners were different.

2.2. Relation classification

Relation classification is a widely studied task in the NLP community. Various approaches have been proposed to accomplish the task. Existing methods for relation classification can be divided into handcrafted feature based methods [2,23], neural network based methods [19,24–27] and the other valuable methods [25,28].

The handcrafted feature based methods focus on using different natural language processing (NLP) tools and knowledge resources to obtain effective handcrafted features. Kambhatla [23] employs Maximum Entropy model to combine diverse lexical, syntactic and semantic features derived from the text. It is the early work for relation classification. The features they used are not comprehensive. Rink [2] designs 16 kinds of features that are extracted by using many supervised NLP toolkits and resources including POS, WordNet, dependency parse, etc. It can get the best result at SemEval-2010 Task 8 when compared with other handcrafted features based methods. However, it relied heavily on other NLP tools and it also requires a lot of work to design and extract features.

In recent years, deep neural models have made significant progress in the task of relation classification. These models can learn effective relation features from the given sentence without complicated feature engineering. The most common neural-network based models applied in this task are Convolutional Neural Networks (CNN) [18,19,27,29,30] and sequential neural networks such as Recurrent Neural Networks (RNN) [31], Recursive Neural Networks (RecNN) [24,32] and Long Short Term Memory Networks (LSTM) [26,33]. There also exists other valuable methods such as the kernel-based methods [28,34] and compositional model [25]. Nguyen et al. [28] explore the use of innovative kernels based on syntactic and semantic structures for the task and Sun and Han [34] propose a new tree kernel, called feature-enriched tree kernel (FTK) for relation extraction. The compositional model FCM [25] learns representations for the substructures of an annotated sentence. Compared to existing compositional models, FCM can easily handle arbitrary types of input and global information for composition.

2.3. Joint entity and relation extraction

Although, pipeline method can be more flexible to design the system, it neglects the relevance of sub-tasks and may also lead to the error propagation [3]. Most existing joint methods are feature-based structured systems [3,4,35–37], which need complicated feature engineering. [35,36] proposed a joint model that uses optimal results of subtasks and seeks a globally optimal solution. Singh et al. [37] proposed a single joint graphical model that represents the various dependencies between subtasks. Li and Ji [3] proposed the first model to incrementally predict entities and relations using a single joint model, which is a structured perceptron with efficient beam search. Miwa and Sasaki [4] introduced a table to represent the entity and relation structures in sentences, and proposed a history-based beam search structured learning model. Recently, Miwa and Bansal [8] used a LSTM-based model to extract entities and relations, which can reduce the manual work.

2.4. LSTM and CNN models On NLP

The methods used in this paper are based on neural network models: Convolutional neural networks (CNN) and Long Short-Term Memory (LSTM). CNN is originally invented for computer vision [38] and it always be used to extract image's features [39,40]. In recent years, CNNs have been successfully applied to different NLP tasks and have also shown the effectiveness on extracting sentence semantic and keywords information [27,41–43]. Long-Short Term Memory (LSTM) model is a specific kind of recurrent neural networks (RNNs). LSTM replaces the hidden vector of a recurrent neural network with memory blocks which are equipped with gates. It can keep long term memory by training proper gating weights [44,45]. LSTM have also shown powerful capacity on many NLP tasks such as machine translation [46], sentence representation [47] and relation extraction [26].

In this paper, we propose a hybrid neural network based on joint learning the entities and their relationships. It can learn related features from given sentences without complicated feature engineering work, when compared with handcrafted feature based methods. When comparing with the other neural network based method [8], our method considers the long distance relationships between entity tags.

Please cite this article as: S. Zheng et al., Joint entity and relation extraction based on a hybrid neural network, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.12.075

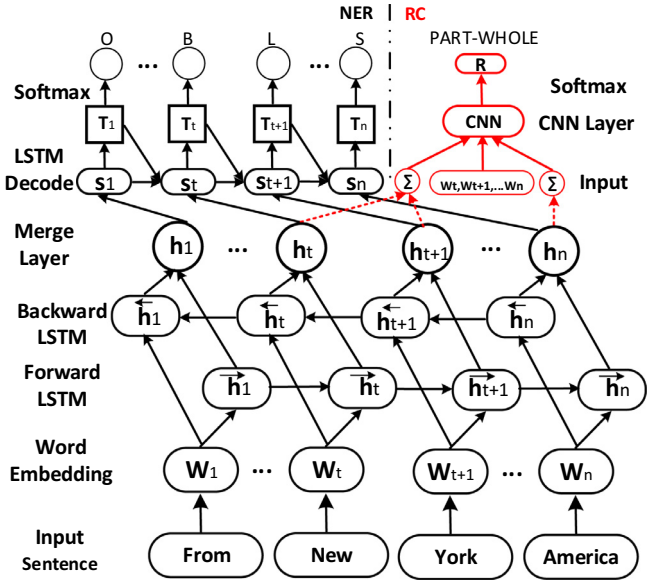

Fig. 1. The framework of the hybrid neural network for jointly extracting entities and relations.

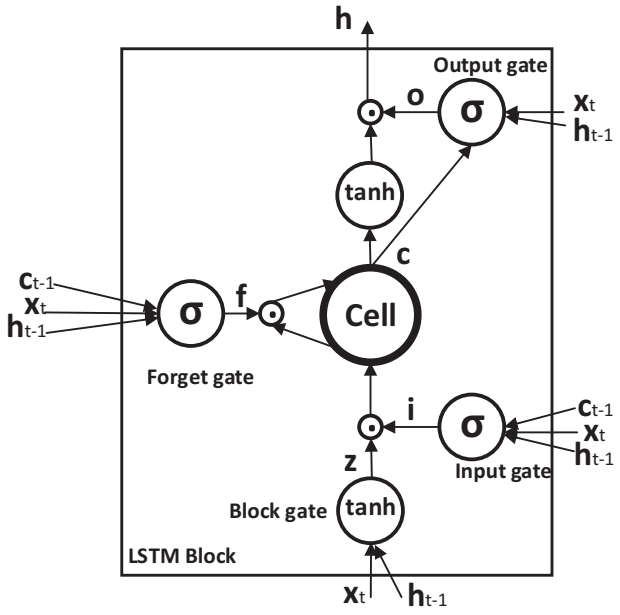

Fig. 2. LSTM memory block with one cell.

3. Our method

The framework of hybrid neural network is shown in Fig. 1. The first layer of hybrid neural network is a bidirectional LSTM encoding layer, which is shared by both named entity recognition (NER) module and relation classification (RC) module. There are two "channels" after encoding layer, one links to the NER module which is a LSTM decoding layer, the other feeds into a CNN layer to extract the relations. In following parts, we describe these components in detail.

3.1. Bidirectional LSTM encoding layer

The Bi-LSTM encoding layer contains word embedding layer, forward lstm layer, backward lstm layer and the concatenate layer. The word embedding layer converts the word with 1-hot representation to an embedding vector. Hence, a sequence of words can be represented as $W = \{w_1, \dots w_t, w_{t+1} \dots w_n\}$, where $w_t \in \mathbb{R}^d$ is the d-dimensional word vector corresponding to the tth word in the sentence and n is the length of the given sentence. After word embedding layer, there are two parallel LSTM layers: forward lstm layer and backward lstm layer. For each word w_t, the forward layer will encode w_t by considering the contextual words information from w_1 to w_t, which marked as $\overrightarrow{h_t}$. In the similar way, the backward layer will encode w_t based on the contextual words information from w_n to w_t, which marked as $\overleftarrow{h_t}$.

The LSTM architecture consists of a set of recurrently connected subnets, known as memory blocks. Each time-step in forward hidden layer and backward hidden layer is a LSTM memory block. A block contains one or more self-connected memory cells and three multiplicative units the input, output and forget gates that provide continuous analogues of write, read and reset operations for the cells [45]. Fig. 2 provides an illustration of a LSTM memory block with a single cell. At each time-step, a lstm memory block is used to compute current hidden vector h_t based on the previous hidden vector h_{t-1}, the previous cell vector c_{t-1} and the current input word embedding w_t, which can be shortly denoted as: $\overrightarrow{h_t} = lstm(\overrightarrow{h_{t-1}}, \overrightarrow{c_{t-1}}, w_t)$ and $\overleftarrow{h_t} = lstm(\overleftarrow{h_{t+1}}, \overleftarrow{c_{t+1}}, w_t)$. The detail operation of *lstm* can be defined as follows:

$$i_t = \delta(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \quad (1)$$

$$f_t = \delta(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \quad (2)$$

$$z_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (3)$$

$$c_t = f_t c_{t-1} + i_t z_t, \quad (4)$$

$$o_t = \delta(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), \quad (5)$$

$$h_t = o_t tanh(c_t), \quad (6)$$

where i, f and o are the input gate, forget gate and output gate respectively, b is the bias term, c is the cell memory, $\cdot$ denotes element-wise multiplication and $W_{(.)}$ are the parameters. Finally, we concatenate $\overleftarrow{h_t}$ and $\overrightarrow{h_t}$ to represent word t's encoded information, which is denoted as $h_t = [\overleftarrow{h_t}, \overrightarrow{h_t}]$.

3.2. Named entity recognition (NER) module

Each word will be assigned an entity tag. The tags are commonly used encoding scheme: BILOS (Begin, Inside, Last, Outside, Single) [22,48]. Each tag contains the position information of a word in the entity. We also adopt a LSTM structure to explicitly model tag interactions. When detecting the entity tag of word t, the inputs of decoding layer are: h_t obtained from Bi-LSTM encoding layer, former tag predicted vector T_{t-1}, and the former hidden state of decoding LSTM s_{t-1}. Each unit of the decoding LSTM is the same as the encoding lstm memory block except for the input gate, which can be rewritten as:

$$i_t = \delta(W_{xi}h_t + W_{hi}s_{t-1} + W_{ti}T_{t-1} + b_i), \quad (7)$$

where the tag predicted vector T is transformed from the hidden state s as follows:

$$T_t = W_{ts}s_t + b_{ts}. \quad (8)$$

The final softmax layer computes normalized entity tag probabilities based on the tag predicted vector T_t:

$$y_t = W_y T_t + b_y, \quad (9)$$

Please cite this article as: S. Zheng et al., Joint entity and relation extraction based on a hybrid neural network, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.12.075

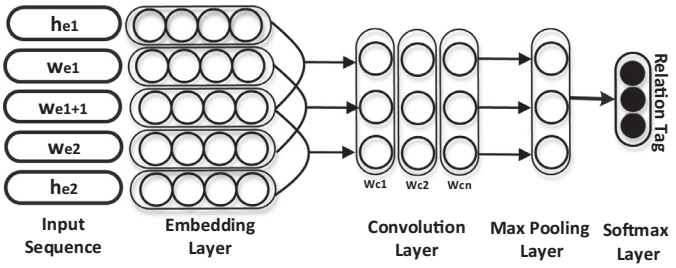

Fig. 3. The convolutional module for relation classification.

$$p_t^i = \frac{exp(y_t^i)}{\sum_{j=1}^{nt} exp(y_t^j)}, \tag{10}$$

where W_y is the softmax matrix, nt is the total number of entity tags. Because the T is similar to tag embedding and LSTM is capable of learning long-term dependencies, this manner can model tag interactions.

3.3. Relation classification (RC) module

When recognizing entities' semantic relationships, we merge the encoding information of entities and the sub-sentence between entities, then feed them into the CNN model [49]. It can be represented as:

$$R = CNN([h_{e1}, w_{e1}, w_{e1+1}, \dots w_{e2}, h_{e2}]), \tag{11}$$

where R is the relation label, h_e is the encoding information of entity, w is the word embedding. Especially, a entity may contain two or more words, we sum up these words' encoding information to represent the whole entity information. Besides, the *CNN* denotes the convolutional operations which shown in Fig. 3.

In convolution layer, we use $W_c^{(i)} \in \mathbb{R}^{k \times d}$ to represent the ith convolution filter and $br^{(i)} \in \mathbb{R}$ to represent the bias term accordingly, where k is the context window size of the filter. Filter $W_c^{(i)}$ will slide through the input sequence $S = [h_{e1}, w_{e1}, w_{e1+1}, \dots w_{e2}, h_{e2}]$ to get the latent features z^i. The sliding process can be represented as:

$$z_l^{(i)} = \sigma\,(W_c^{(i)} * s_{l:l+k-1} + br^{(i)}), \tag{12}$$

where $z_l^{(i)}$ is the feature extracted by filter $W_c^{(i)}$ from word s_l to word s_{l+k-1}. Hence, the latent features of the given sequence S are denoted as: $z^{(i)} = [z_1^{(i)}, \dots, z_{L-k+1}^{(i)}]$. We then apply the max-pooling operation to reserve the most prominent feature of filter $W_c^{(i)}$ and denote it as:

$$z_{max}^{(i)} = max\{z^{(i)}\} = max\{z_1^{(i)}, \dots, z_{L-k+1}^{(i)}\}. \tag{13}$$

We use multiple filters to extract multiple features. Therefore, the relation features of the given sequence is represented as: $R_s = [z_{max}^{(1)}, \dots, z_{max}^{(nr)}]$, where nr is the number of filters.

After that we set a soft-max layer [50] with dropout [51] to classify the relations based on relation features R_s, which is defined as:

$$y_r = W_R \cdot (R_s \circ r) + b_R, \tag{14}$$

$$p_r^i = \frac{exp(y_r^i)}{\sum_{j=1}^{nc} exp(y_r^j)}, \tag{15}$$

where $W_R \in \mathbb{R}^{nr \cdot nc}$ is the softmax matrix, nc is the total number of relation classes, symbol $\circ$ denotes the element-wise multiplication operator and $r \in \mathbb{R}^{nr}$ is a binary mask vector drawn from Bernoulli with probability ρ. Dropout guards against overfitting, which makes the model more robust. In Formula 15, p_r^i means the probability that the sentence describes relation i.

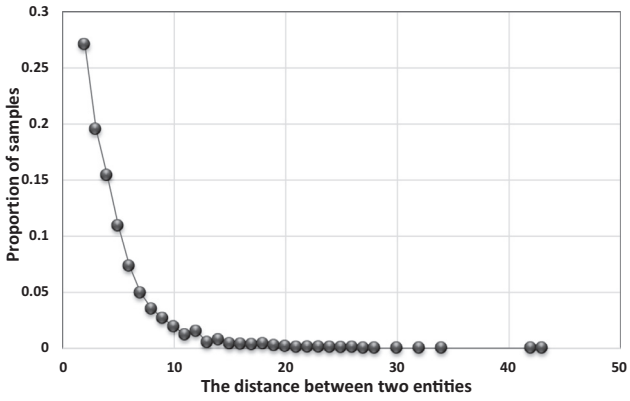

Fig. 4. The distribution of dataset based on the distance between two entities. The horizontal axis is the distance between two entities and the vertical axis represents the number of samples corresponding to distance.

3.4. Training and implementation

We train our models to maximize the log-likelihood of the data and the optimization method we used is RMSprop proposed by Hinton in [52]. The objective function of NER module can be defined as:

$$L_{ner} = max \sum_{j=1}^{|\mathbb{D}|} \sum_{t=1}^{L_j} log(p_t^{(j)} = y_t^{(j)} | x_j, \Theta_{ner}), \tag{16}$$

where $|\mathbb{D}|$ is the size of dataset, L_j is the length of sentence x_j, $y_t^{(j)}$ is the label of word t in sentence x_j and $p_t^{(j)}$ is the normalized entity tag probabilities which defined in Formula 10. Besides, RC module's objective function is:

$$L_{rc} = max \sum_{j=1}^{|\mathbb{D}|} log(p_r^{(j)} = y_r^{(j)} | x_j, \Theta_{rc}), \tag{17}$$

where $p_r^{(j)}$ is defined in Formula 15.

We firstly train NER module to recognize the entities and obtain the encoding information of entities, then further train the RC module to classify relations based on the encoding information and the entities combinations.

Specially, we find that if there is a relationship between the two entities, the distance of two entities always smaller than about 20 words, which is shown in Fig. 4. Hence, when determining the relationship between the two entities, we also make full use of this property that if the distance of two entities is larger than L_{max}, we don't think there exists a relationship between them. L_{max} is around 20 in the ACE05 dataset based on the statistical results of Fig. 4.

4. Experiment

4.1. Experimental setting

Datasets. We use public dataset ACE05 for entity and relation extraction, which 6 coarse-grained relation types and an additional "other" relation to denote non-entity or non-relation classes. The 6 coarse-grained relation types are "ART (artifact)", "G-A (Gen-affiliation)", "O-A (Org-affiliation)", "P-W (PART-WHOLE)", "P-S (person-social)" and "PHYS (physical)". The same relation type

Please cite this article as: S. Zheng et al., Joint entity and relation extraction based on a hybrid neural network, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.12.075

Table 1
Hyper parameters of the hybrid neural network.

Parameter	Parameter description	Parameter value
d	Dimension of word embedding	300
ne	The number of hidden units in encode layer	300
nd	The number of hidden units in decode layer	300
k	Context window size of CNN module	3
nr	The filter number of CNN	100
ρ	The ratio of dropout	0.3

Table 2
Comparisons with the baselines on the ACE05 test set.

Model	P(%)	R(%)	$F1$(%)
Pipeline (CRF+ME) [3]	65.1	38.1	48.0
Joint w/Global [3]	65.4	39.8	49.5
SPTree [8]	65.8	42.9	51.9
Our method	61.9	45.0	**52.1**

with opposite directions are considered to be two classes. For example, "PART-WHOLE (e1,e2)" and "PART-WHOLE(e2,e1)" are different relations. The "PART-WHOLE(e1,e2)" means that e1 is a part of e2 and "PART-WHOLE(e2,e1)" means e1 contains e2. Hence, there are 13 relation classes in total. The data pre-processing and settings in experiments are the same as [3].

Baselines. The baselines we used are recent methods for the ACE05 dataset, which include a classical pipeline model [3], a joint feature-based model called Joint w/Global [3], and an end-to-end NN-based model SPTree [8].

- Pipeline (CRF+ME) [3] trained a linear-chain Conditional Random Fields model [53] for entity mention extraction and a Maximum Entropy model [54] for relation extraction. It is a classical pipeline method for the task.
- Joint w/Global [3] incrementally extract entity mentions together with relations using a single model. They developed a number of new and effective global features as soft constraints to capture the interdependency among entity mentions and relations.
- SPTree [8] presented a novel end-to-end relation extraction model that represents both word sequence and dependency tree structures by using bidirectional sequential and bidirectional tree-structured LSTM-RNNs.

Metrics. To compare our model with baselines, we use Precision (P), Recall (R) and F- Measure (F1) in the task of joint entity and relation extraction. A relation instance is regarded as correct when its relation type and the head offsets of two corresponding entities are both correct.

Hyper parameters. In this paper, we propose a hybrid neural network to extract entities and their relations. The hyper parameters used in the model are summarized in Table 1.

4.2. Results

The predicted results on test set are shown in Table 2. Our method achieves $F1$ of 52.1%, which is the best result when comparing with the existing methods. It illustrates the effectiveness of our proposed hybrid neural network on the task of jointly extracting the entities and their relationships.

Besides, the Joint w/Global [3] approach outperforms the pipelined method and the neural network based methods (SPTree [8] and our model) can get a higher $F1$ results than these feature based methods [3]. It shows that neural network model accompanied with joint learning manner is a feasible way to extract entities and their relationships.

Especially, the precision results of these methods are similar and the difference is mainly concentrated in the recall results. Our method can balance the precision and recall, which achieve a better $F1$ result.

5. Analysis and discussions

5.1. Analysis of named entity recognition module

The NER module contains a bidirectional LSTM encoding layer and a LSTM decoding layer. We use BiLSTM-ED to represent the structure of NER module. In order to further illustrate the effectiveness of BiLSTM-ED on the task of entity extraction, we compare BiLSTM-ED with its different variations and other effective sequence labeling models. The contrast methods are:

- Forward-LSTM uses a unidirectional LSTM to encode the input sentence from w_1 to w_n, then also applies a LSTM structure to decode the entity tags.
- Backward-LSTM has the similar manner of Forward-LSTM, the difference is the encoding order which is from w_n to w_1.
- BiLSTM-NN uses a bi-directional LSTM to encode the input sentence and uses a feed-forward neural network (NN) architecture to predict the entity tags. It neglects the relationship between tags.
- BiLSTM-NN-2 [8] uses a bi-directional LSTM to encode the input sentence and uses a novel feed-forward neural network (NN) by considering adjacent tags information instead of the long distance relationships between tags.
- CRF [53] is a classic and effective sequence labeling model. In this section, we use CRF as one of the powerful comparison method and the feature used in CRF are the same as [3] used.

We use the standard $F1$ to evaluate the performance of these methods and treat an entity as correct when its type and the region of its head are correct. Table 3 shows the results of the above methods on the task of name entity recognition. When comparing with Forward-LSTM and Backward-LSTM, the bi-directional LSTM encoding manner can have significant improvements. Bi-LSTM encoding considers the whole sentence information when comparing with the ui-LSTM encoding, hence it can achieve much higher accuracies in the tagging task. BiLSTM-NN-2 is better than BiLSTM-NN, which shows the need of considering the relationship between tags. Besides, BiLSTM-ED is batter than BiLSTM-NN-2 which means that considering the long distance relationships between tags can be better than only considering the adjacent tags information. We also compare BiLSTM-ED with the famous sequential model of CRF. The result also shows the effectiveness of BiLSTM-ED.

5.2. Analysis of relation classification module

In the relation classification module, we use two kinds of information: the sub-sentence between entities and the encoding information of entities obtained from bidirectional LSTM layer. In order to illustrate the effectiveness of these information we considered,

Table 3
Comparisons with the different methods on the task of entity detection.

Methods	P(%)	R(%)	$F1$(%)
Forward-LSTM	63.8	59.2	60.0
Backward-LSTM	65.3	60.0	61.0
CRF	83.2	73.6	78.1
BiLSTM-NN	83.3	83.0	82.2
BiLSTM-NN-2	85.5	81.2	83.3
BiLSTM-ED	85.2	85.4	**84.2**

Please cite this article as: S. Zheng et al., Joint entity and relation extraction based on a hybrid neural network, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.12.075

Table 4
Comparisons of different information on the task of relation classification.

Methods	*P*(%)	*R*(%)	*F*1(%)
Full-CNN	30.8	34.9	32.7
Sub-CNN	57.7	51.9	54.6
Sub-CNN-H	58.3	54.8	**56.5**

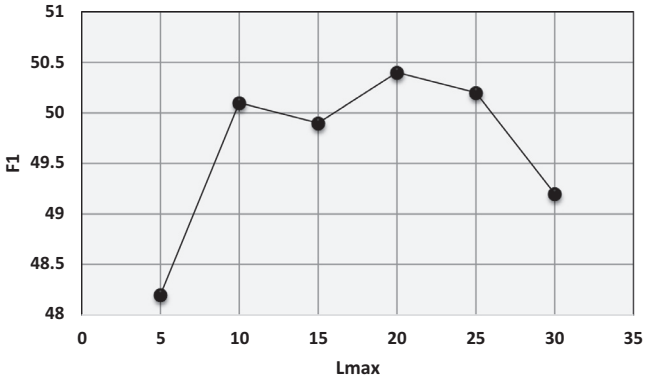

Fig. 5. The *F*1 results of different L_{max} values. The horizontal axis is the distance between two entities and its range is from 5 to 30. The vertical axis represents *F*1 value on the relation classification tasks. In order to exclude the effect of encoding information, we use Sub-CNN to obtain the *F*1 results.

we compare our method with its different variations. We firstly use the NER module to detect the entities in sentence, then uses the right entity recognition results of step 1 to test the RC module. We report the effect of these information on the relation classification task as Table 4 shows. Full-CNN uses a whole sentence to recognize the relationships of entities. Sub-CNN only uses the sub-sentence between two entities. Sub-CNN-H uses both the sub-sentence and the encoding information of entities obtained from bidirectional encoding layer. When comparing Full-CNN with Sub-CNN, the result shows that sub-sentence can achieve a +20% improvement. This result matches [9]'s analysis that most relationships can be reflected by the sub-sentence between the given two entities instead of full sentence. When adding the encoding information of entities into the Sub-CNN, Sub-CNN-H can further promote the accuracy of relation classification. It verifies that entities' contextual information is also benefit for identifying relationships between entities.

5.3. The effect of two entities' distance

From Fig. 4, we know that the data distribution shows the long tail property when the horizontal axis is the distance between two entities. Hence, we set a threshold L_{max} to filter the data. If two entities' distance is larger than L_{max}, we treat that these two entities have no relationship. In order to analyze the effect of threshold L_{max}, we use Sub-CNN to predict entities relationships based on different L_{max} values. The effect is shown in Fig. 5. The smaller of L_{max} is , the more data will be filtered. So if L_{max} is too small, it may filter the right data and make the *F*1 results decline. If L_{max} is too large, it cannot filter the noisy data which may also hurt the final results. Fig. 5 shows that when L_{max} is between 10 and 25, it can perform well. The range also matches the statistical results of Fig. 4.

5.4. Error analysis

To analyze the errors of our method, we visualized the model's predicted results on relation classification task as Fig. 6 shows. The diagonal region indicates the correct prediction results and the other regions reflect the distribution of error samples. The highlighted diagonal region means that our method can perform well on each relation class except for the relation "P-S". Because the test dataset contains a few samples whose relation labels are "P-S", the predicted distribution of "P-S" cannot fully reflect the true situation. Besides, "P-S" means the relationship of "person-social". The entity "person" and entity "social" always are pronoun words in the dataset, so it is hard to recognize "P-S" relationship based on these pronoun words.

Further more, from Fig. 6, we also can see that the distribution of predicted relation is relatively dispersed on the first row of "OTHER", which means that most of the specific relation classes can be predicted as the "OTHER". Namely, we cannot identify some relationships and it directly leads to relatively low recall. From the first column of "OTHER", we can see that if there is no relationship between the two entities the model can be effectively discriminated.

Apart from the class "OTHER", the other problem is that the same relation type with opposite directions are ease to mix up,

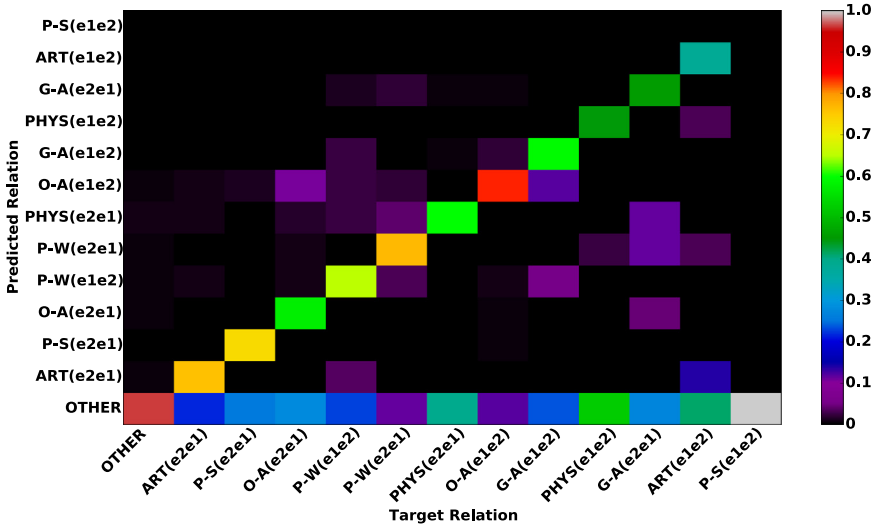

Fig. 6. The distribution of the predicted results for each relation class. The horizontal axis is the target relation and each target relation corresponds to a column of predicted relations. Point (X,Y) means the ratio that the target relation is X and the predicted relation is Y. The sum of each column value equal to 1.

Please cite this article as: S. Zheng et al., Joint entity and relation extraction based on a hybrid neural network, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.12.075

such as: P-W(e2e1) and P-W(e1e2), ART(e1e1)and ART(e2e1), O-A(e1e1)and O-A(e2e1). The reason is that the same relation type always have similar description even if they are not in the same direction.

6. Conclusion

Entity and relation extraction is an important issue in knowledge extraction and plays a vital role in automatic construction of knowledge base. In this paper, we propose a hybrid neural network model to extract entities and their semantic relationships without any handcrafted features. When comparing with the other neural network based method, our method considers the long distance relationships between entity tags. In order to illustrate our methods' effectiveness, we conduct experiments on the public dataset ACE05 (Automatic Content Extraction program). The experimental results on the public dataset ACE05 verify the effectiveness of our method.

In the future, we will explore how to better link these two modules based on the neural network, so that it can perform better. Besides, we also need to solve the problem of neglecting some relationships and try to promote the recall value.

Acknowledgment

We thank Qi Li and Miwa for dataset details and helpful discussions. We also thank Qi Li for providing the partition of dataset so that we can conduct contrast experiment in a fair environment. This work is also supported by the National High Technology Research and Development Program of China (863 Program) (Grant No. 2015AA015402), the Hundred Talents Program of Chinese Academy of Sciences (No. Y3S4011D31) and National Natural Science Foundation (Grant No. 71402178).

References

[1] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Lingvisticae Investigationes 30 (1) (2007) 3–26.
[2] B. Rink, Utd: classifying semantic relations by combining lexical and semantic resources, in: Proceedings of the 5th International Workshop on Semantic Evaluation, 2010, pp. 256–259.
[3] Q. Li, H. Ji, Incremental joint extraction of entity mentions and relations., in: Proceedings of the 52rd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 402–412.
[4] M. Miwa, Y. Sasaki, Modeling joint entity and relation extraction with table representation., in: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1858–1869.
[5] Y.S. Chan, D. Roth, Exploiting syntactico-semantic structures for relation extraction, in: Proceedings of the 49rd Annual Meeting of the Association for Computational Linguistics, 2011, pp. 551–560.
[6] X. Yu, W. Lam, Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach, in: Proceedings of the 21th COLING International Conference, 2010, pp. 1399–1407.
[7] L. Li, J. Zhang, L. Jin, R. Guo, D. Huang, A distributed meta-learning system for chinese entity relation extraction, Neurocomputing 149 (2015) 1135–1142.
[8] M. Miwa, M. Bansal, End-to-end relation extraction using lstms on sequences and tree structures, in: Proceedings of the 54rd Annual Meeting of the Association for Computational Linguistics, 2016.
[9] C.N. dos Santos, B. Xiang, B. Zhou, 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, vol. 1, 2015, pp. 626–634.
[10] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, Z. Jin, Classifying relations via long short term memory networks along shortest dependency paths, in: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2015.
[11] L. Zou, R. Huang, H. Wang, J.X. Yu, W. He, D. Zhao, Natural language question answering over RDF: a graph data driven approach, in: Proceedings of the 2014 ACM SIGMOD international conference on Management of data, ACM, 2014, pp. 313–324.
[12] J. Sang, C. Xu, J. Liu, User-aware image tag refinement via ternary semantic analysis, IEEE Trans. Multimed. 14 (3) (2012) 883–895.
[13] J. Sang, C. Xu, Right buddy makes the difference: An early exploration of social relation analysis in multimedia applications, in: Proceedings of the 20th ACM International Conference on Multimedia, ACM, 2012, pp. 19–28.
[14] G. Luo, X. Huang, C.-Y. Lin, Z. Nie, Joint entity recognition and disambiguation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015, pp. 879–888.
[15] J.P. Chiu, E. Nichols, Named entity recognition with bidirectional lstm-cnns, arXiv:1511.08308 (2015).
[16] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv:1508.01991 (2015).
[17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2016.
[18] K. Xu Y. Feng, S. Huang, D. Zhao, Semantic relation classification via convolutional neural networks with simple negative sampling, arXiv:1506.07650 (2015).
[19] D. Zeng, K. Liu, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: Proceedings of the 25th COLING International Conference, 2014, pp. 2335–2344.
[20] A. Passos, V. Kumar, A. McCallum, Lexicon infused phrase embeddings for named entity resolution, in: Proceedings of the International Conference on Computational Linguistics, 2014, pp. 78–86.
[21] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537.
[22] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, arXiv:1603.01354 (2016).
[23] N. Kambhatla, Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations, in: Proceedings of the 43th ACL International Conference, 2004, p. 22.
[24] R. Socher, B. Huval, C.D. Manning, A.Y. Nq, Semantic compositionality through recursive matrix-vector spaces, in: Proceedings of the EMNLP International Conference, 2012, pp. 1201–1211.
[25] M. Yu, M. Gormleyl, M. Dredze, Factor-based compositional embedding models, in: Proceedings of the NIPS Workshop on Learning Semantics, 2014.
[26] X. Yan, L. Moul, G. Li, Y. Chen, H. Peng, Z. Jin, Classifying relations via long short term memory networks along shortest dependency paths, in: Proceedings of EMNLP International Conference, 2015.
[27] C.N. dos Santos, B. Xiangl, B. Zhou, Classifying relations by ranking with convolutional neural networks, in: Proceedings of the 53th ACL International Conference, vol. 1, 2015, pp. 626–634.
[28] T.-V.T. Nguyen, A. Moschittil, G. Riccardi, Convolution kernels on constituent, dependency and sequential structures for relation extraction, in: Proceedings of the EMNLP International Conference, 2009, pp. 1378–1387.
[29] P. Qin, W. Xu, J. Guo, An empirical convolutional neural network approach for semantic relation classification, Neurocomputing 190 (2016) 1–9.
[30] S. Zheng, J. Xu, P. Zhou, H. Bao, Z. Qi, B. Xu, A neural network framework for relation extraction: Learning entity semantic and relation pattern, Knowl. Based Syst. 114 (2016) 12–23.
[31] D. Zhang D. Wang, Relation classification via recurrent neural network, arXiv:1508.01006 (2015).
[32] J. Ebrahimi, D. Dou, Chain based RNN for relation classification, in: Proceedings of the NAACL International Conference, 2015, pp. 1244–1249.
[33] S. Zhang, D. Zheng, X. Hu, M. Yang, Bidirectional long short-term memory networks for relation classification, in: Proceedings of the Pacific Asia Conference on Language, Information and Computation, 2015, pp. 73–78.
[34] L. Sun, X. Han, A feature-enriched tree kernel for relation extraction, in: Proceedings of the 52th ACL International Conference, 2014, pp. pages 61–67.
[35] D. Roth, W.-t. Yih, Global inference for entity and relation identification via a linear programming formulation, in: Introduction to Statistical Relational Learning, 2007, pp. 553–580.
[36] B. Yang, C. Cardie, Joint inference for fine-grained opinion extraction., in: Proceedings of the 51rd Annual Meeting of the Association for Computational Linguistics, 2013, pp. 1640–1649.
[37] S. Singh, S. Riedel, B. Martin, J. Zheng, A. McCallum, Joint inference of entities, relations, and coreference, in: Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, ACM, 2013, pp. 1–6.
[38] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[39] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. (2016), doi:10.1109/TCYB.2016.2591583.
[40] J. Yu, B. Zhang, Z. Kuang, D. Lin, J. Fan, Image privacy protection by identifying sensitive objects via deep multi-task learning, in: Proceedings of the IEEE Transactions on Information Forensics and Security, 2016.
[41] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the EMNLP International Conference, 2014.
[42] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: Proceedings of the 52th ACL International Conference, 2014.
[43] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, H. Hao, Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification, Neurocomputing 174 (2016) 806–814.
[44] X. Zhu, P. Sobihani, H. Guo, Long short-term memory over recursive structures, in: Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 2015, pp. 1604–1612.
[45] A. Graves, Supervised Sequence Labelling, Springer, 2012.
[46] M.-T. Luong, I. Sutskever, Q.V. Le, O. Vinyals, W. Zaremba, Addressing the rare word problem in neural machine translation, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015, pp. 11–19.

Please cite this article as: S. Zheng et al., Joint entity and relation extraction based on a hybrid neural network, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.12.075

[47] R. Kiros, Y. Zhu, R.R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, S. Fidler, Skip-thought vectors, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 3276–3284.
[48] L. Ratinov, D. Roth, Design challenges and misconceptions in named entity recognition, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, 2009, pp. 147–155.
[49] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2014.
[50] K. Duan, S.S. Keerthi, W. Chu, S.K. Shevade, A.N. Poo, Multi-category classification by soft-max combination of binary classifiers, in: Multiple Classifier Systems, Springer, 2003, pp. 125–134.
[51] G.E. Dahl, T.N. Sainath, G.E. Hinton, Improving deep neural networks for LVCSR using rectified linear units and dropout, in: Proceedings of the ICASSP, 2013, pp. 8609–8613.
[52] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop, COURSERA: Neural networks for machine learning (2012).
[53] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML, vol. 1, 2001, pp. 282–289.
[54] S.J. Phillips, R.P. Anderson, R.E. Schapire, Maximum entropy modeling of species geographic distributions, Ecol. Modell. 190 (3) (2006) 231–259.

Suncong Zheng is a Ph.D candidate in Institute of Automation Chinese Academy of Sciences. He received his B.S. degree in School of TianJin University, China, in 2012. His research interests include information extraction and web/text mining.

Yuexing Hao is a M.S. candidate in Institute of Automation Chinese Academy of Sciences. She received his B.S. degree in School of University of Science Technology Beijing, China, in 2014. Her research interests include information extraction and web/text mining.

Dongyuan Lu is a Lecturer with the School of Information Technology and Management, University of International Business and Economics, Beijing, China. She received the B.S. degree from Beijing Normal University, Beijing, China, in 2007, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2012. Then she continued her research work in National University of Singapore as a research fellow for 2 years. Her research interests include social media analysis, information retrieval, and data mining.

Hongyun Bao is an assistant researcher in the Institute of Automation Chinese Academy of Sciences. She received his B.S. degree in School of Mathematical Sciences from Capital Normal University, China, in 2008, and Ph.D. degree from Chinese Academy of Sciences, in 2013. Her research interests include information extraction and web/text mining.

Jiaming Xu is an assistant researcher in the Institute of Automation Chinese Academy of Sciences. He received his M.S. degree in School of University of Science Technology Beijing, China, in 2012, and Ph.D. degree from Chinese Academy of Sciences, in 2016. His research interests include information extraction and web/text mining.

Hongwei Hao is the deputy director of Interactive Digital Media Technology Research Center, Institute of Automation, Chinese Academy of Sciences. His research interests include semantic computation, pattern recognition, machine learning, and image processing. He has published over 50 papers in Chinese Journals, international journals and conferences.

Bo Xu is a Professor at the Institute of Automation Chinese Academy of Sciences. He received the B.S. degree in Zhejiang University in 1988. From 1988 he joined the speech recognition research and received the Master's and Doctor's Degree in the field in 1992 and 1997 respectively. Now he is the President of CASIA and takes position in committee of National high-tech Program in fields of Chinese Information Processing, Multimedia and Virtuality. He has published more than 100 papers on major journals and processdings including IEEE Trans.

Please cite this article as: S. Zheng et al., Joint entity and relation extraction based on a hybrid neural network, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.12.075