Postprint of Semantic Relation Classification Model Based on Stacked Recurrent Neural Networks

Authors: Hao Zhifeng, Chen Peihui, Cai Ruichu, Wen Wen, Wang Lijuan

Date: 2018-11-29T00:00:00+00:00

Abstract

Methods based on recurrent neural networks combined with syntactic structures have been widely employed for relation classification, utilizing neural networks to automatically extract features from encoded input information and achieve relation classification; however, existing approaches are primarily based on models of a single specific syntactic structure, and such models cannot be transferred to other types of syntactic structures. To address this problem, we propose a stacked recurrent neural network model that integrates multiple syntactic structures. This stacked recurrent neural network is constructed in two layers: first, entity pre-training is performed at the sequence layer, where an Attention mechanism is integrated into Bi-LSTM-CRF to enhance the model' s focus on entity information in the text sequence, thereby obtaining more accurate entity feature information to facilitate better classification at the relation layer stage; second, at the relation layer, Bi-Tree-LSTM is nested on top of the sequence layer, and the hidden states of the sequence layer along with entity feature information are passed to the relation layer, where shared parameters are utilized to perform weighted learning on three different syntactic structures, achieving semantic relation classification through end-to-end model training. Experimental results demonstrate that the model achieves a macro-F1 score of 85.9% on the SemEval-2010 Task8 corpus, and further enhances the robustness of the model.

Full Text

Preamble

Vol. 37 No. 1
Application Research of Computers (ChinaXiv Cooperative Journal)

Semantic Relation Classification Model via Hierarchical Recurrent Neural Network

Hao Zhifeng[1,2], Chen Peihui[1], Cai Ruichu[1], Wen Wen[1], Wang Lijuan[1]
(1. School of Computer Science, Guangdong University of Technology, Guangzhou 510006, China;
2. School of Mathematics & Big Data, Foshan University, Foshan, Guangdong 528000, China)

Abstract: Methods based on recurrent neural networks combined with syntactic structures have been widely applied to relation classification, leveraging neural networks to automatically extract features from encoded input information for classification. However, existing approaches primarily rely on models built upon single, specific syntactic structures, which cannot be transferred to other syntactic structure types. To address this limitation, we propose a hierarchical recurrent neural network model that integrates multiple syntactic structures. This hierarchical network is constructed in two layers: First, at the sequence layer, we perform entity pre-training using a Bi-LSTM-CRF fused with an attention mechanism to enhance the model' s focus on entity information within the text sequence, thereby obtaining more accurate entity feature representations that facilitate better classification in the relation layer. Second, at the relation layer, we nest Bi-Tree-LSTM above the sequence layer, feeding the hidden states and entity feature information from the sequence layer into the relation layer. Using shared parameters, the model performs weighted learning across three different syntactic structures and achieves semantic relation classification through end-to-end training. Experimental results demonstrate that our model achieves a macro-F1 score of 85.9% on the SemEval-2010 Task 8 corpus, further improving model robustness.

Keywords: hierarchical recurrent neural network; multi-syntactic structure; Bi-Tree-LSTM; attention mechanism; relation classification

---

0 Introduction

To cope with massive amounts of unstructured text data generated on the internet, it is common practice to convert such data into structured formats to help people obtain important information more quickly. Relation classification, as one of the most widely used techniques for this purpose, has attracted increasing attention from the research community. Relation classification primarily involves two important types of information: text sequences and syntactic dependencies. The key research question addressed in this paper is how to introduce structured knowledge and better fuse these two components to improve classification performance.

In recent years, with the rapid development of deep learning, approaches to relation classification have gradually transitioned from traditional methods based on feature engineering and kernel construction to directly using deep neural networks to learn from encoded input information for classification. Representative works include methods based on recurrent/recursive neural networks

(RNN) and convolutional neural networks (CNN), with RNN-based models being more adaptable to text problems due to their ability to directly represent linguistic structures such as text sequences and syntactic dependency trees.

The relation classification process mainly involves two-stage tasks: named entity recognition and semantic relation classification. The mainstream methods for these tasks are sequence labeling-based approaches and syntactic structure-based approaches, respectively. When performing the second-stage task, current methods primarily rely on single, specific syntactic structures. Such models have certain limitations: they can only handle specific syntactic structures and cannot be easily transferred to other structures. For example, consider the sentence: "A thief who tried to steal the truck broke the ignition with screwdriver." Its Shortest Dependency Path (SDP) structure contains the entity pair "thief" and "ignition" along with the predicate "broke" between them. For the same sentence, its SubTree structure contains not only the SDP but also additional components such as the determiner "A" and the modifier "with screwdriver." Therefore, a network model designed for SDP would suffer from information loss when applied to SubTree structures and cannot be directly transferred.

Consequently, this paper proposes a hierarchical recurrent neural network model that fuses multiple syntactic structures: (a) At the sequence layer, we perform entity pre-training using Bi-LSTM-CRF fused with an attention mechanism to improve the model' s focus on entity information in the text sequence, thereby obtaining more accurate entity label information; (b) At the relation layer, we use Bi-Tree-LSTM to receive the fused features of sequence layer output and dependency labels as new input, perform weighted learning across three different syntactic structures, update shared parameters through backpropagation, and finally output semantic relation categories through a softmax classifier. Considering that the interaction between the two-stage tasks is closely coupled, constructing an end-to-end model enables mutual promotion and enhancement of the complete model' s effectiveness.

The significance of this research lies in proposing a network model that fuses multiple syntactic structures and constructing a hierarchical end-to-end network for the two tasks, which can better adapt to different syntactic structures and improve the accuracy and stability of relation classification. The contributions at different network levels are: (a) At the entity pre-training stage of the sequence layer, we consider the correlation between input and output, and combining word-level attention mechanism can effectively enhance the model' s attention to entity information on the text sequence, which promotes relation classification; (b) At the relation layer, fusing multiple syntactic structures enables the model to effectively classify different syntactic structures within the same framework, improving model robustness while also verifying the contribution of different syntactic structures to relation classification.

---

1 Research Status

Relation classification is essentially a classification problem. The typical approach is to first identify entity pairs in a sentence and then use a classifier to determine which components represent the actual relations. Early relation classification problems were solved with the help of knowledge bases, but due to the high cost of constructing knowledge bases, research direction shifted toward machine learning. Current research can be divided into three categories:

a) Feature-based methods. These methods extract a large number of linguistic features, including semantic and syntactic features, combine them to form feature vector sets, and use classifiers (such as maximum entropy models and SVM) for classification. This approach works well for specific domains but relies on expert knowledge for feature selection and design, incurring significant time costs.

b) Kernel-based methods. These methods compute the inner product of two objects in high-dimensional sparse space to obtain structured features. Zelenko et al. designed tree kernel functions for shallow syntactic analysis to obtain structural commonalities. Culotta et al. extended Zelenko's work to dependency trees and incorporated syntactic parsing information. Bunescu et al. fused syntactic shortest paths with kernel functions to explore classification effects across different syntactic structures. Zhang et al. used convolutional tree kernels to explore the role of syntactic features in relation classification. Zhou et al. added textual content information based on Zhang et al.' s work. These methods require skill in kernel function selection and have slow training speeds with large datasets; moreover, classification performance depends on NLP tools, and errors in text preprocessing can affect classifier performance.

c) Neural network-based methods. The advantage of these methods is that they can self-learn features from encoded input information without manual feature construction, while rich encoding information overcomes the sparsity problem of traditional methods. In RNN-based approaches combined with syntactic structures, Socher et al. used recursive matrix-vector operations to obtain semantic compositionality. Xu et al. and Liu et al. demonstrated that syntactic shortest paths are helpful for neural network models to capture semantic relations. Li et al. discussed classification effects of different syntactic tree structures in neural network models. Miwa et al. proposed Bi-Tree-LSTM and simultaneously considered different syntactic structure types and their child node quantity relationships. Zhou et al. directly nested an attention layer on Bi-LSTM to enhance encoded weight information. Xiao et al. segmented long sentences and used a two-layer RNN network with attention to encode information from different layers. Zhang et al. proposed an improved RNN-CNN method that connects CNN to bidirectional LSTM and performs fully connected input to the classifier after convolution on attention-weighted features. With the proposal of reinforcement learning and adversarial networks, related research has also emerged, such as Feng et al.' s reinforcement learning method for relation classification from noisy data and Liu et al.' s adversarial training framework for relation classification.

Compared with the other two traditional methods, neural network-based methods have the advantage that the network can automatically learn features without manual definition, achieving optimal classification performance. Relative to CNN, RNN-based models can better handle text sequences and syntactic structures. However, current models combining syntactic structures for relation classification suffer from the problem of single syntactic structures, making it impossible to transfer models for specific syntactic structures to other different syntactic structures. Therefore, this paper proposes a hierarchical recurrent neural network model that fuses multiple syntactic structures, using Bi-LSTM-CRF (fused with attention mechanism) and Bi-Tree-LSTM to learn from text sequences and multiple syntactic structures, performing end-to-end training with shared parameters, and finally outputting semantic relation categories.

---

2 H-RNN-Based Relation Classification Framework

Our relation classification framework mainly consists of four parts: input sequence, entity pre-training, multi-syntactic fusion structure, and output semantic relations. The main components are the sequence layer and relation layer, which represent text sequences and syntactic structures using bidirectional recurrent neural networks, with the relation layer nested above the sequence layer to form an end-to-end model. The framework is illustrated in [Figure 1: see original paper]. Specifically, at the sequence layer, the input text sequence is encoded by Bi-LSTM, word-level attention is used to obtain the weight of each word in the sentence, and finally CRF performs bidirectional decoding. At the relation layer, dependency labels are concatenated with the sequence layer output as input for this stage, Bi-Tree-LSTM performs weighted learning across multiple syntactic structures to obtain candidate relations between entity pairs, and finally the softmax classifier obtains the final semantic relations.

2.1 Basic Framework Description

The proposed model framework mainly consists of three components: word vector representation of input sequences, entity pre-training at the sequence layer, and multi-syntactic fusion structure at the relation layer. Word vectors include character embeddings, part-of-speech tag embeddings, and syntactic label embeddings of dimensions d^w, d^p, and d^d, respectively, all obtained through pre-trained embeddings.

At the sequence layer, we adopt the standard Bi-LSTM for sequence encoding. At time step t, LSTM contains an input gate i_t, a forget gate f_t, an output gate o_t, a memory cell c_t, and a hidden state h_t, calculated as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$
$$u_t = \tanh(W_u x_t + U_u h_{t-1} + b_u) \tag{4}$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \tag{5}$$
$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where σ is the element-wise sigmoid function, $\odot$ is element-wise multiplication, W and U are weight matrices, and b are bias vectors.

For a sequence containing n words $\mathbf{x} = [x_1, x_2, \dots, x_n]$, each word is represented by a d-dimensional feature vector. At time step t, for the current t-th word, LSTM receives the current input vector x_t, the previous hidden state h_{t-1}, and cell state c_{t-1}, and returns the current hidden state h_t. Since we use Bi-LSTM, each word obtains a hidden state in both forward and backward directions, denoted as $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, respectively. After concatenation, we get $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$, which serves as one of the inputs to the relation layer.

In the decoding stage of the sequence layer, we adopt the standard CRF model to decode the output vectors from attention, using the BILOU tagging scheme, where each entity label contains entity type and position information. As shown in [Figure 2: see original paper], B-PER and L-PER indicate that "Sidney Yates" is a PER entity type with corresponding positions. The final entity label vector $\mathbf{e}_t$ will serve as one of the inputs to the relation layer.

The attention mechanism at the sequence layer and the relation layer will be elaborated separately below.

2.2 Word-level Attention

The sequence layer contains not only entity information in the sentence but also entity context information and some irrelevant information. To enable the model to better predict relations between entities, it needs to focus more on the most critical information in the sequence, namely the two main types of information: entities and predicate actions.

Therefore, we introduce a word-level attention mechanism at the sequence layer. The attention mechanism enables the model to process the hidden vectors $\mathbf{h}_t$ generated by Bi-LSTM word by word along the text sequence, obtaining corresponding weight assignments and producing their weighted representation $\mathbf{z}_t$, as shown in Equation (2):

$$\mathbf{a}_t = \tanh(\mathbf{W}_a \mathbf{h}_t + \mathbf{b}_a) \tag{7}$$

$$\alpha_t = \frac{\exp(\mathbf{v}_a^\top \mathbf{a}_t)}{\sum_{j=1}^{T} \exp(\mathbf{v}_a^\top \mathbf{a}_j)} \tag{8}$$

$$\mathbf{z}_t = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t \tag{9}$$

where $\mathbf{W}_a$ is a weight matrix, $\mathbf{v}_a$ is a weight vector, and $\mathbf{v}_a^\top$ is its transpose.

2.3 Relation Layer

The relation layer represents candidate relations between entity pairs on the syntactic dependency tree, as shown in the right part of [Figure 1: see original paper]. Existing work has demonstrated that different syntactic structures provide varying benefits for relation classification, but current models are mostly based on specific syntactic structures. Therefore, we propose integrating three different syntactic structures through weighted fusion, allowing the network to learn relations under different syntactic structures.

First, we define SDP, SubTree, and FullTree structures:

a) SDP (Shortest Dependency Path): The core dependency path between the nearest common node and the two entity target words.

b) SubTree: A syntactic structure that includes the shortest path and the subtree under the nearest common node.

c) FullTree: The complete syntactic dependency tree that captures the full contextual information of the sentence.

We adopt Bi-Tree-LSTM to obtain candidate relations between entity pairs. This network structure can not only fully learn information from leaf nodes and their child nodes but also propagate root node information to leaf nodes, which is beneficial for the network to fully learn information on syntactic structures. Miwa et al. defined two types of syntactic structures in their model and used a mapping function to process the two types of syntactic structure trees. Based on this, we propose an improved Bi-Tree-LSTM that fuses three syntactic structure types through shared weights and can calculate the number of child nodes under different structures.

Since entity recognition and semantic relation classification are joint tasks, nesting the relation layer above the sequence layer and passing effective information obtained from the text sequence to the relation layer for end-to-end training can effectively improve the training effect of the complete model. The LSTM unit at the relation layer receives input at the t-th word as $\mathbf{x}_t^{rel} = [\mathbf{h}_t; \mathbf{e}_t; \mathbf{d}_t]$ (Co-Embedding in the right part of [Figure 2: see original paper]), where $\mathbf{h}_t$ is the hidden state vector from the sequence layer, $\mathbf{e}_t$ is the entity label obtained from entity pre-training, and $\mathbf{d}_t$ is the dependency label of the word.

The Bi-Tree-LSTM unit calculation formula is as follows:

$$i_t = \sigma\Big(W_i \sum_{s\in C(t)} x_s^{rel} + U_i \sum_{s\in C(t)} h_s + b_i\Big) \tag{10}$$

$$f_{ts} = \sigma\Big(W_f x_s^{rel} + U_f h_s + b_f\Big) \tag{11}$$

$$o_t = \sigma\Big(W_o \sum_{s\in C(t)} x_s^{rel} + U_o \sum_{s\in C(t)} h_s + b_o\Big) \tag{12}$$

$$u_t = \tanh\Big(W_u \sum_{s\in C(t)} x_s^{rel} + U_u \sum_{s\in C(t)} h_s + b_u\Big) \tag{13}$$

$$c_t = i_t \odot u_t + \sum_{s\in C(t)} f_{ts} \odot c_s \tag{14}$$

$$h_t = o_t \odot \tanh(c_t) \tag{15}$$

where $C(t)$ represents the set of child nodes of word t in the syntactic structure.

$\mathbf{p}_\phi$ represents the linear weighted learning of syntactic structures. When the model learns different syntactic structures, it needs to calculate the number of child nodes of word t under the corresponding syntactic structure. ϕ represents that structure s is one of all tree structures T. We define two node types: one is nodes on the SDP path, and the other is all other nodes on SubTree and FullTree besides the shortest path.

In the decoding stage of the complete model, based on the composition idea, we combine the last words of each detected entity to obtain the final candidate relation, i.e., words tagged with L and U labels in BILOU mode. In the [Figure 1: see original paper] example, we mark a candidate relation for Yates (L-PER) and Chicago (U-LOC). For each input sentence, the relation layer obtains candidate relation $\mathbf{h}_{root}^{\downarrow}$ through Bi-Tree-LSTM, passes through a hidden layer containing a tanh function, and finally predicts relation labels through a softmax layer.

Since we use Bi-Tree-LSTM at the relation layer, the candidate relation is $\mathbf{h}_{root}^{\downarrow} = [\overrightarrow{\mathbf{h}}_{root}^{\downarrow}; \overleftarrow{\mathbf{h}}_{e1}^{\downarrow}; \overleftarrow{\mathbf{h}}_{e2}^{\downarrow}]$, where "$\overrightarrow{\mathbf{h}}_{root}^{\downarrow}$" represents the hidden state vector of the top unit in the bottom-up direction (i.e., the predicate component), and "$\overleftarrow{\mathbf{h}}_{e1}^{\downarrow}$, $\overleftarrow{\mathbf{h}}_{e2}^{\downarrow}$" represent the hidden state vectors of the two bottom leaf nodes in the top-down direction (i.e., entity nodes).

In the final relation prediction stage, the model predicts the final candidate relation through a two-layer neural network: a hidden layer $\mathbf{h}_d$ and a softmax layer, with formulas as follows:

$$\mathbf{h}_d = \tanh(\mathbf{W}_d \mathbf{h}^{\downarrow}_{root} + \mathbf{b}_d) \quad (16)$$

$$\mathbf{p}_d = \text{softmax}(\mathbf{W}_p \mathbf{h}_d + \mathbf{b}_p) \quad (17)$$

2.4 Model Training

Regarding the model training process and parameter settings: At the sequence layer, we adopt the standard linear-chain CRF based on the BILOU scheme for entity pre-training on the training set, using the obtained labels as one of the inputs to the relation layer. At the relation layer, we use softmax as the classifier, placed above the hidden layer of the relation layer, which receives the sentence representation $\mathbf{h}_d$ from the hidden layer and produces the probability distribution of semantic relations for the sentence (as shown in Equation (4)). The training objective is to minimize the cross-entropy between the predicted relations and actual semantic relations of sentences:

$$\text{loss} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{C} y_{ik} \log y'_{ik} + \frac{\lambda}{2}\|\theta\|^2$$

where y_{ik} represents the probability that sentence i belongs to class k, y'_{ik} represents the predicted probability that sentence i belongs to class k, $\frac{\lambda}{2}\|\theta\|^2$ is the L2 regularization term, and θ is its parameter. We only regularize the weights $\mathbf{W}$ and $\mathbf{U}$.

To prevent overfitting, we employ early stopping during training by setting a value $n = 100$; if the cross-entropy does not achieve a better value within n iterations, training can be terminated early. To ensure consistency in the network training process, we pad sentences of unequal length. Detailed parameter settings for model training are described in Section 3.2.

---

3 Experimental Results and Discussion

The experiments aim to demonstrate: (a) that fusing word-level attention mechanism can effectively enhance the model' s focus on entity information in text sequences, promoting the classification effect of the complete model; (b) that weighted learning of multiple syntactic structures can improve model robustness, ensuring the model can handle different syntactic structures. Below, we first introduce the experimental data and evaluation metrics, experimental settings, then present experimental tests and result analysis, and finally compare with other methods.

3.1 Experimental Data and Evaluation Metrics

We use SemEval-2010 Task 8 as the experimental dataset, which is a classic benchmark for evaluating relation classification tasks. The dataset contains 8,000 training samples and 2,717 test samples, with relation categories including 9 semantic relations plus an Other relation (indicating no relation between entity pairs). The specific semantic relation-sample distribution is shown in .

Table 1 Semantic relation-sample distribution of SemEval-2010 Task 8

We adopt macro-F1 as the evaluation metric, which depends on precision and recall, calculated as follows:

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \tag{18}$$

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \tag{19}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

where TruePositive represents the number of correctly predicted relations, FalsePositive represents the number of incorrectly predicted relations, and FalseNegative represents the number of relations that were not predicted.

3.2 Experimental Settings

Word vectors are initialized using GloVe vectors pre-trained on English Wikipedia with a dimension of 200. We use the Stanford Dependency Parser for syntactic dependency parsing of sentences. The model activation function is tanh, with 200 hidden layer nodes. We use softmax as the classifier and apply L2 regularization to weights $\mathbf{W}$ and $\mathbf{U}$. To prevent overfitting, we introduce dropout strategies for both the sequence layer and relation layer during training, with a dropout value of 0.3. Additionally, we use batched Adam with gradient clipping to update weight parameters during model training, with batch size set to 1 and training epochs set to 100. Other parameters are initialized randomly.

3.3 Experimental Results and Analysis

Our experiments use the results from Miwa et al. as the baseline, which achieves 84.4% classification performance without additional prior knowledge WordNet.

3.3.1 Sequence Layer Improvements In the first group of experiments, based on Miwa et al.' s method, we improved the sequence layer of their model by using different decoding methods and adding word-level attention mechanism. The experimental results are shown in .

Table 2 Experimental results of improved sequence layer (without extra prior knowledge)

Model	Macro-F1 (%)
Baseline (SDP)	84.4
+CRF	-
+ATTENTION	-

The results show that improving the decoding method to CRF and adding attention mechanism enhances the final classification effect of the baseline model. The reason is that the original baseline model' s sequence layer uses a unidirectional greedy decoding approach for entity pre-training. Replacing it with CRF can capture bidirectional contextual information, thereby learning more comprehensive information. On the other hand, although replacing with CRF decoding improves performance, the classification effect does not significantly improve, possibly because the learning of entity information is insufficient. Adding word-level attention mechanism improves classification performance by enhancing the model' s focus on entity information in the text sequence, allowing the model to better learn entity information. This experiment demonstrates that using CRF and attention mechanism can effectively improve entity pre-training at the sequence layer.

3.3.2 Multi-Syntactic Structure Fusion In the second group of experiments, while keeping the sequence layer for entity pre-training from Miwa et al. unchanged, we replaced the baseline model' s dependency layer with a relation layer that fuses multi-syntactic structures. The experimental results are shown in .

Table 3 Experimental results of multi-syntactic structure (with extra prior knowledge, WordNet)

Model	Macro-F1 (%)
Baseline	-
Multi-syntactic structure	-
SubTree	-
FullTree	-

The baseline model uses a mapping function to select which syntactic structure to use, while our model fuses multiple syntactic structures into one, aiming to improve model robustness so that the model can directly handle different syntactic structures. The results show that our proposed multi-syntactic structure fusion model demonstrates stable and reliable performance across different syntactic structures, with improved results compared to the baseline model, exhibiting good robustness.

Furthermore, from the first group of experiments, we know that using CRF decoding and adding attention mechanism can improve entity pre-training at the sequence layer, which promotes classification in the complete model. Therefore, by combining the two groups of experiments—stacking the improved sequence layer with the multi-syntactic fusion structure for end-to-end training of the complete model—the experimental results are shown in .

Table 4 Experimental results of improved sequence layer + multi-syntactic structure (with extra prior knowledge, WordNet)

Model	Macro-F1 (%)
Multi-syntactic structure	-
SubTree	-
FullTree	-
SDP (baseline)	-

As shown in , compared with other models, our H-RNN achieves state-of-the-art results with fewer additional features.

Table 5 Comparison between H-RNN and other methods

Model	Additional Features	Macro-F1 (%)
SVM	POS, WordNet, prefixes and other morphological features, dependency parse, Levin classes, PropBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner	-
CNN	position features, words around nominals, WordNet	-
SDP-LSTM	position features, WordNet	-
BLSTM	POS embeddings, WordNet embeddings, grammar relation embeddings	-
Att-BLSTM	POS, NER, WordNet, position features, dependency feature, relative-dependency feature	-
2ATT-BLSTM-BLSTM	Without any other features, only use word vector (50 dim & 100 dim) and position indicators	-
BLSTM-BTLSTM (SDP)	position features, WordNet, NER	-

Model	Additional Features	Macro-F1 (%)
H-RNN (SDP)	POS, dependency feature, WordNet	**85.9**

The results show that our proposed H-RNN model outperforms the baseline model with additional prior knowledge and sequence-based models. The reasons may be: (a) Entity recognition and relation classification are interrelated tasks; improvements at the sequence layer enhance entity pre-training, which promotes training of the end-to-end complete model, thereby improving classification performance; (b) Both text sequences and syntactic structures contain different information components; sequence-based models have overly single information content, and introducing structured information into text sequences can enrich the model' s classification capability; (c) It further proves that SDP is more effective for relation classification than SubTree and FullTree, as SDP is more intuitive and concise in structure, while the other two structures introduce redundant node information that may interfere with learning important information during training.

---

4 Conclusion

To address the problem that existing relation classification models based on specific syntactic structures cannot be transferred to other syntactic structures, this paper proposes a hierarchical recurrent neural network model that fuses multiple syntactic structures. The model uses Bi-LSTM-CRF (fused with attention mechanism) and Bi-Tree-LSTM to represent text sequences and syntactic structures, integrates entity pre-training and relation classification into an end-to-end framework for training with shared parameters, incorporates attention mechanism during entity pre-training to enhance focus on entities, and simultaneously performs weighted learning across multiple syntactic structures.

Experiments on the SemEval-2010 Task 8 dataset demonstrate that: (1) The multi-syntactic structure network model can effectively classify relations across different syntactic structures, exhibiting certain robustness. It further proves that the SDP structure (compared to other structures) is the most effective for relation classification. (2) By constructing an end-to-end model, using attention mechanism during entity pre-training to enhance focus on entities, and employing shared parameter learning, the model' s classification accuracy is effectively improved.

References

[1] Iris H, Su Nam K, Zornitsa K, et al. Semeval-2010 task 8: multi-way classi-

fication of semantic relations between pairs of nominals [C]// Proc of the 5th International Workshop on Semantic Evaluation, SemEval' 10. Stroudsburg, PA: ACL Press, 2010: 33–38.

[2] Kambhatla N. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations [C]// Proc of the 42nd ACL on Interactive poster and demonstration sessions. Stroudsburg, PA: ACL Press, 2004: No. 22.

[3] Zhou GuoDong, Su Jian, Zhang Jie, et al. Exploring various knowledge in relation extraction [C]// Proc of the 43rd annual meeting on association for computational linguistics. Stroudsburg, PA: ACL Press, 2005: 427-434.

[4] Zhang Zhu. Weakly-supervised relation classification for information extraction [C]// Proc of the 13th ACM international conference on Information and knowledge management. New York: ACM Press, 2004: 581-588.

[5] Zelenko D, Aone C, Richardella A. Kernel methods for relation extraction [J]. Journal of machine learning research, 2003, 3 (Feb): 1083-1106.

[6] Culotta A, Sorensen J. Dependency tree kernels for relation extraction [C]// Proc of the 42nd Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA: ACL Press, 2004: No. 423.

[7] Bunescu R C, Mooney R J. A shortest path dependency kernel for relation extraction [C]// Proc of Conference on Human Language Technology and Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL Press, 2005: 724-731.

[8] Katiyar A, Cardie C. Nested named entity recognition revisited [C]// Proc of the 16th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: ACL Press, 2018: 861-871.

[9] Wang Qi, Xia Yuhang, Zhou Yangming, et al. Incorporating dictionaries into deep neural networks for the Chinese clinical named entity recognition [J]. arXiv, 2018, preprint arXiv: 1804. 05017.

[10] Zheng Suncong, Wang Feng, Bao Hongyun, et al. Joint extraction of entities and relations based on a novel tagging scheme [C]// Proc of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL Press, 2017: 1227–1236.

[11] Yan Xu, Mou Lili, Li Ge, et al. Classifying relations via long short term memory networks along shortest dependency paths [C]// Proc of the 20th Conference Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL Press, 2015: 1785-1794.

[12] Zhou Peng, Wei Shi, Tian Jun, et al. Attention-based bidirectional long short-term memory networks for relation classification [C]// Proc of the 54th

Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL Press, 2016: 207-212.

[13] Zeng Daojian, Liu Kang, Lai Siwei, et al. Relation classification via convolutional deep neural network [C]// Proc of the 25th International Conference on Computational Linguistics. Stroudsburg, PA: ACL Press, 2014: 2335-2344.

[14] Aone C, Ramos-Santacruz M. REES: a large-scale relation and event extraction system [C]// Proc of the 6th conference on Applied natural language processing. Stroudsburg, PA: ACL Press, 2000: 76-83.

[15] Zhang Min, Zhang Jie, Su Jian. Exploring syntactic features for relation extraction using a convolution tree kernel [C]// Proc of Main Conference on Human Language Technology, Conference of the North American Chapter of the Association of Computational Linguistics. Stroudsburg, PA: ACL Press, 2006: 288-295.

[16] Zhou Guodong, Zhang Min, Ji Donghong, et al. Tree kernel-based relation extraction with context-sensitive structured parse tree information [C]// Proc of the Joint Conference on EMNLP-CoNLL. Stroudsburg, PA: ACL Press, 2007: 728-736.

[17] Socher R, Huval B, Manning C D, et al. Semantic compositionality through recursive matrix-vector spaces [C]// Proc of Joint Conference on EMNLP-CoNLL. Stroudsburg, PA: ACL Press, 2012: 1201-1211.

[18] Xu Kun, Feng Yansong, Huang Songfang, et al. Semantic relation classification via convolutional neural networks with simple negative sampling [J]. arXiv, 2015, preprint arXiv: 1506. 07650.

[19] Liu Yang, Wei Furu, Li Sujian, et al. A dependency-based neural network for relation classification [J]. arXiv, 2015, preprint arXiv: 1507. 04646.

[20] Li Jiwei, Thang Luong, Dan Jurafsky, et al. When are tree structures necessary for deep learning of representations? [C]// Proc of Conference on EMNLP. Stroudsburg, PA: ACL Press, 2015: 2304-2314.

[21] Miwa M, Bansal M. End-to-end relation extraction using lstms on sequences and tree structures [C]// Proc of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL Press, 2016: 1105–1116.

[22] Xiao Minguang, Liu Cong. Semantic relation classification via hierarchical recurrent neural network with attention [C]// Proc of the 26th International Conference on Computational Linguistics. Stroudsburg, PA: ACL Press, 2016: 1254-1263.

[23] Zhang Xiaobin, Chen Fucai, Huang Ruiyang. A Combination of RNN and CNN for attention-based relation classification [J]. Procedia Computer Science, 2018, 131: 911-917.

[24] Feng Jun, Huang Minlie, Zhao Li, et al. Reinforcement Learning for Relation Classification from Noisy Data [C]// Proc of AAAI. Palo Alto, CA: AAAI Press,

2018.

[25] Liu Wenpeng, Cao Yanan, Cao Cong, et al. An adversarial training framework for relation classification [C]// Proc of ICCS. Berlin: Springer Press, 2018: 194-205.

[26] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM networks [C]// Proc of IJCNN on IEEE. Piscataway, NJ: IEEE Press, 2005, 4: 2047-2052.

[27] Ratinov L, Roth D. Design challenges and misconceptions in named entity recognition [C]// Proc of the 13th Conference on Computational Natural Language Learning. Stroudsburg, PA: ACL Press, 2009: 147-155.

[28] Rink B, Harabagiu S. Utd: Classifying semantic relations by combining lexical and semantic resources [C]// Proc of the 5th International Workshop on Semantic Evaluation. Stroudsburg, PA: ACL Press, 2010: 256-259.

[29] Zhang Shu, Zheng Dequan, Hu Xinchen, et al. Bidirectional long short-term memory networks for relation classification [C]// Proc of the 29th Pacific Asia Conference on Language, Information and Computation. Stroudsburg, PA: ACL Press, 2015: 73-78.

Note: Figure translations are in progress. See original paper for figures.