An End-to-End Entity and Relation Extraction Network with Multi-head Attention

Lishuang Li*[1],Yuankai Guo,Shuang Qian and Anqiao Zhou

Dalian University of Technology, Dalian, 116024, China
lilishuang314@163.com
{guoyuankai, QShuang, ahashi_syuu}@ mail.dlut.edu.cn

Abstract. Relation extraction is an important semantic processing task in natural language processing. The state-of-the-art systems usually rely on elaborately designed features, which are usually time-consuming and may lead to poor generalization. Besides, most existing systems adopt pipeline methods, which treat the task as two separated tasks, i.e., named entity recognition and relation extraction. However, the pipeline methods suffer two problems: (1) Pipeline model over-simplifies the task to two independent parts. (2) The errors will be accumulated from named entity recognition to relation extraction. Therefore, we present a novel joint model for entities and relations extraction based on multi-head attention, which avoids the problems in the pipeline methods and reduces the dependence on features engineering. The experimental results show that our model achieves good performance without extra features. Our model reaches an F-score of 85.7% on SemEval-2010 relation extraction task 8, which has competitive performance without extra feature compared with previous joint models. On publication, codes will be made publicly available.

Keywords: Relation Extraction, End-to-End Joint Extraction, Named Entity Recognition.

1 Introduction

Named entity recognition (NER) and relation extraction are important semantic processing tasks in natural language processing (NLP). Traditional pipeline methods divide the task into two parts: named entity recognition and relation extraction. Firstly, entities are recognized in the sentences. Then, the identified entities are combined into entity pairs. Finally, the relations between entities pairs are extracted. The mainstream pipeline methods are based on neural network models, such as convolutional neural networks (CNN) and recurrent/recursive neural networks (RNNs). Zeng et al.

*Corresponding author: lilishuang314@163.com
The paper is supported by the National Natural Science Foundation of China under No. 61672126.

[1] exploited a CNN to extract lexical and sentence level features for relation classification and achieved an F1-score of 82.7%. Yan et al. [2] presented a LSTM model with shortest dependency path (SDP) for relation extraction, which reached 83.7% F1-score. Zhou et al. [3] proposed a BLSTM with an attention model for relation extraction and reached 84.0% F1-score. The above experiments are based on the SemEval-2010 task 8 dataset. NER and relation extraction can be improved independently in the pipeline method. However, the pipeline methods ignore the interaction between NER and relation extraction. In addition, the errors in the upstream components are propagated to the downstream components without any feedback [4].

Recent studies show that end-to-end (joint) modeling of entity and relation is important for high performance [5]. The joint model processes NER and relation extraction simultaneously, which can alleviate the errors propagation. Furthermore, the NER and the relation extraction components share some parameters in joint model, which could capture the interaction between the sub-tasks. Li et al. [4] presented a model using structured perceptron with efficient beam-search on ACE04, which employed many features such as global entity mention features and local features. Miwa and Sasaki [6] proposed a history-based structured learning approach using lexical, contextual features and so on, which obtained 69.8% F1-score on CONLL 04 dataset. Although the above models adopted the joint method for entity and relation extraction, they are all based on shallow machine learning methods, which still rely on feature engineering such as dependency features and syntactic features. Miwa et al. [5] proposed a LSTM based sequence and tree-structure model and achieved 85.5% F1-score on SemEval-2010 Task 8. Katiyar A et al. [7] presented attention-based pointer network for joint extraction of entity mentions and relations, reaching an F1-score of 53.6% on ACE05 dataset. Zheng et al. [8] proposed a joint extraction of entities and relations based on a tagging scheme using LSTM and achieved 49.5% F1-score on NYT dataset. The above models utilized the deep-learning models with simple features. Some model also applied attention mechanism such as Katiyar A et al.'s model. The application of attention mechanism obtains a representation weighted by the importance of tokens.

In this paper, we propose a novel end-to-end attention-based BLSTM model for entities and relation extraction without any handcraft features and structure features. The contributions of this paper are as follows:

1. Our model enhances the interaction between entities and relations. The model can utilize the output of the hidden layers in NER to provide more information for relation extraction. The entity labels and the output of hidden layers are fed into the BLSTM with a multi-head attention layer to extract the relation between entity pairs.

2. Our multi-head attention can focus on the words which have decisive effect for relation extraction. Comparing with other attention mechanisms, our multi -head attention can capture different relevant features for relation extraction.

The experimental results show that our model achieves an F1-score of 85.66% on the SemEval-2010 task 8 [9], which has competitive performance without extra feature compared with previous joint models.

2 Proposed Method

In this section, we describe our end-to-end multi-head attention model in detail. The framework of the model is illustrated in Figure 1. The model mainly contains three components: the word embedding layer, the name entity recognition layer based on BLSTM-CRF and the relation extraction layer based on BLSTM with multi-head attention.

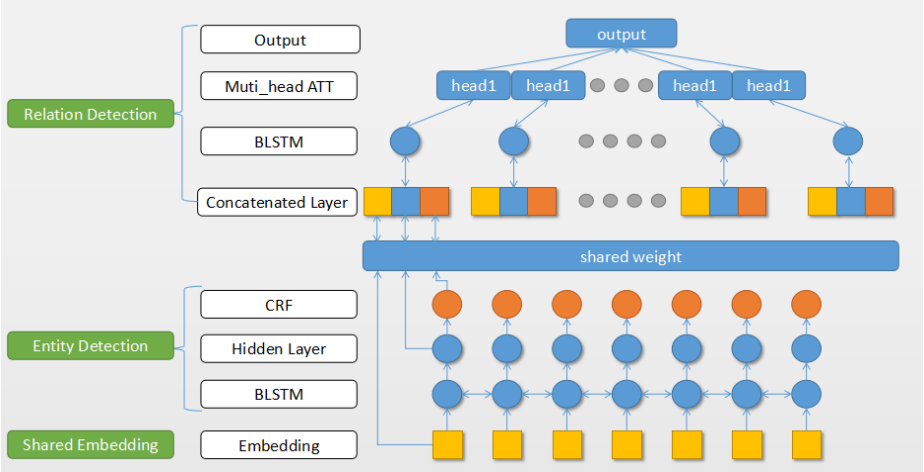

Fig. 1. End-to-End entity and relation extraction model with multi-head attention.

2.1 Entity detection

LSTM units are firstly proposed by Hochreiter and Schmidhuber [10] to overcome gradient vanishing problem. LSTMs are more capable of capturing long-term dependencies between tokens, making it ideal for both entity mentions and relation extraction. We use the following implementation:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
g_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + W_{cc}c_{t-1} + b_c) \\
c_t &= i_t g_t + f_t c_{t-1} \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \tanh(c_t),
\end{aligned}
\tag{1}
$$

where i_t, f_t and o_t are the input gate, forget gate and output gate respectively. W_{xi}, W_{hi} and W_{ci} are the parameters of weights matrix of input gate. W_{xf}, W_{hf} and W_{cf} stand for the corresponding weights matrix of forget gate respectively. Similarly W_{xo}, W_{ho} and W_{co} are the weights matrix of output gate. b is the bias term and the hidden vectors are represented as $H=\{h_1,h_2,...,h_n\}$, where n is the number of words in a sentence. All of those gates are set to generate some degrees, using the current input x_t, the state h_t that

previous step generated, and the current state of this cell c_t. In this paper, we employ bidirectional LSTM (BLSTM) which contains two sub-networks for the left and right sequence context. For each sentence coming from the embedding layer, the forward LSTM encodes a sentence from left to right and the backward LSTM encodes a sentence from right to left. We concatenate the output of forward LSTM and backward LSTM as the following equation:

$$h_i = [\vec{h}_i \oplus \overleftarrow{h}_i] \tag{2}$$

2.2 CRF layer

In some traditional methods, the output of BLSTM H is used to get independent tagging for the corresponding words in the sentence. However, NER tasks have strong dependencies across the output label. For example, it is illegal when the label "B" follows the label "I". Compared with the independent output layer, Conditional Random Fields (CRF) [11] can efficiently use the whole sentence tag from the output layer, which can obtain the best sequence of tags. The output H of BLSTM is marked as $X = \{x_1, x_2, ..., x_n\}$, which is the input of CRF layer. We consider P as the matrix of the output by BLSTM with size of $n*m$ (n is the length of the sentence, m is the number of tag's classes). $P_{i,j}$ denotes the score of the *j-th* tag of *i-th* word. For a sequence of predictions $y = \{y_1, y_2, ..., y_n\}$, the score is described as follows:

$$K(x, y) = \sum_{i=0}^{n} A_{y_i, y_{(i+1)}} + \sum_{i=1}^{n} P_{i, y_i}, \tag{3}$$

where A represents the matrix of transition scores. A softmax over all possible tag sequences yields a probability for the sequence y as Equation (4).

$$p(y \mid X) = \frac{e^{s(X, y)}}{\sum_{\tilde{y} \in Y_X} e^{s(X, \tilde{y})}}. \tag{4}$$

We maximize the log-probability of the correct tag sequence as Equation (5):

$$\log(p(y \mid S)) = K(X, y) - \log(\sum_{\tilde{y} \in Y_X} e^{K(X, \tilde{y})}), \tag{5}$$

where YX represents all possible tag sequences including the format which does not obey the "BIO" format constraints. We get the maximum score given by:

$$y^* = \arg\max_{\tilde{y} \in Y_X} K(X, \tilde{y}) \tag{6}$$

The output of entity labels is transformed into one-hot vectors. The output of the hidden layer and the one-hot vectors are fed into the concatenated layer.

3 Relation Extraction detection

3.1 Concatenated layer

In the stage of relation detection, the input contains three parts: the hidden weights of NER $H=\{h_1,h_2,...,h_n\}$. The label of tokens $L=\{l_1,l_2,..,l_n\}$ from CRF layer, and the word embeddings shared with name entity recognition $embs=\{e_1, e_2,...,e_n\}$. We concatenate the hidden weights of NER, token labels and word embeddings as the following format $d_i=\{h_i;l_i;e_i\}$. The concatenate vectors $D=\{d_1,d_2,...,d_n\}$ are fed into the BLSTM with multi-head attention layer to extract the relations between entities pairs. The parameters of the entity layer are shared and they are jointly updated in entity recognition and relation extraction training. The shared weights enhance the relevance of entities and relations, and provide more features for relation extraction.

3.2 Multi-head Attention layer on BLSTM

Some words play a key role in relation extraction. For example, in the sentence "The <e1>burst</e1> has been caused by water hammer <e2>pressure</e2>.", the word '*caused*' is critical to the relation extraction because the it may reflect relation type. These decisive words should get more attention than other words. Guided by this, we employ a multi-head attention mechanism to focus on the importance of different words for relation classification. The multi-head attention applies attention mechanism multiple times over the same inputs using separately attention heads and combines the results which could concentrate on the different relevant feature for relation extraction. The attention function can be described as Equation (7-8):

$$\begin{aligned} &\alpha = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_w}}) \\ &head_i = \sum_n \alpha V \\ &Q \in n*d_w, k \in n*d_w, V \in n*d_w, i \in H, \end{aligned} \tag{7}$$

$$\mathrm{Multi}Head(Q,K,V) = (head_1 \oplus \cdots \oplus head_H), \tag{8}$$

where *Q, K* and *V* represent the attention query matrix, key matrix and value matrix respectively. In our model, we use self-attention mechanism in each head of attention. Thus *Q, K,* and *V* represent the output sequences of BLSTM with size of $n*d_w$, where d_w is the dimension of BLSTM output. *H* represents the number of heads and *i* indicates the *i-th* head of attention. For each head of attention, we compute attention weights by the Equation (7), which captures the different features from the sentences. We concatenate each head from the left to right and get the final results.

4 Relation Extraction detection

4.1 Dataset and Tasks

We evaluate our model on SemEval-2010 task 8 dataset [9]. The dataset contains 8000 training instances and 2717 test instances annotated with 9 different relation types and an artificial relation "Other", which is used to indicate that the relation does not belong to any of the nine main relation types. Table 1 shows the details of the dataset.

Each instance contains a sentence marked with two nominals <e1> and <e2.>, and the task is used to predict the relation between the two nominals considering the directionality. It means that the relation Cause-Effect (e1, e2) is different from the relation Cause-Effect (e2, e1), as shown in the examples below:

"The current view is that the chronic <e1>inflammation</e1> in the distal part of the stomach caused by Helicobacter pylori <e2>infection</e2> results in an increased acid production from the non-infected upper corpus region of the stomach."

Cause-Effect (e2, e1)

"The <e1>singer</e1>, who performed three of the nominated songs, also caused a <e2>commotion</e2> on the red carpet."

Cause-Effect (e1, e2)

The results are measured using official evaluation metric, which is based on macro-averaged F1-score for the nine proper relations and others. The definition of Precision (*P*), Recall (*R*) and *F1-score* are shown as Equation (9):

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \; F1 - score = \frac{2 * P * R}{P + R}, \tag{9}$$

where *TP* is short for true positives, *FP* represents false positives, and *FN* stands for false negatives.

Table 1. Annotation statistics of SemEval-2010 task 8 dataset.

Relation	Freq in Train set	Freq in Test set
Cause-Effect	(1003)12.53%	(328)12.07%
Instrument-Agency	(504)6.30%	(156)5.74%
Product-Producer	(717)8.96%	(231)8.50%
Content-Container	(540)6.75%	(192)7.06%
Entity-Origin	(716)8.95%	(258)9.49%
Entity-Destination	(845)10.56%	(292)10.47%
Component-Whole	(941)11.76%	(312)11.48%
Member-Collection	(690)8.62%	(233)8.57%
Message-Topic	(634)7.92%	(261)9.60%
other	(1410)17.62%	(454)16.70%
Total	(8000)100%	(2717)100%

4.2 Hyper-Parameters Settings

This subsection presents the hyper parameter tuning for our model. The model is implemented in Keras and trained on a single 1080Ti GPU. We employ Adam method [12] to optimize our model. The learning rate is set to 0.001 and batch size is 32. For multi-head attention we set the number of attention heads to 4. We use the publicly available GloVe [13] to train word embeddings. The dimensions of word vectors are set as 200. The dropout rate is set to be 0.4 to prevent overfitting [14]. L2-regularizations are also employed in training to prevent overfitting.

4.3 Overall Performance

Table 2. Results on SemEval-2010 task 8 based on entity hidden weights and multi-head attention

Model	P(%)	R(%)	F(%)
BLSTM	82.76%	83.10%	82.93%
BLSTM +entity_hidden_weights	84.08%	84.35%	84.21%
BLSTM +multi-head attention	83.94%	84.74%	84.33%
BLSTM +entity_hidden_weights +multi-head attention	84.92%	86.41%	85.66%

Table 2 shows the experimental results of our method on SemEval-2010 task 8 dataset. The model achieves 84.21% F1-score only using the entity hidden weights, which is 1.31 percentage points higher than that of BLSTM pipeline model. The entity hidden weights can provide the information of entities and context in the process of NER. Experimental results demonstrate that the output of the hidden layer of NER is of benefit to the relation extraction and improves the performance of the model.

In addition, we propose a multi-attention mechanism to capture the related semantic information of each word. The multi-head attention is able to focus on the words which are critical to the relation extraction. Comparing with other attention mechanisms, our multi-head attention can capture different relevant features for relation extraction. In Table 2, we show the results of models with multi-head attention mechanisms. From the results, we can observe that BLSTM with multi-head attention performs well, whose F-score is 1.4% percentage points higher than the BLSTM model, which proves the effectiveness of our attention mechanism.

Table 2 also shows the result of our model based on multi-head attention combined with entity hidden weights, which achieves 85.66% F1-score. The results indicate that our model with multi-head attention and entity hidden weights can promote the performance.

4.4 Comparison with Previous Models

Table 3. Comparison with other models on SemEval-2010 task.

Model	Features	F1-score
Yan et al., (SDP-LSTM) [2]	Word vector+Grammer relation+POS+WordNet	83.7%
Zhang et al., (RNN) [15]	Word vector +POS+position feature	80.0%
Miwa et al., (LSTM-RNN) [5]	Word vector+SDPTree + WordNet	85.5%
Dos santos et al.,(CR-CNN) [16]	Word vector+Position feature	84.1%
Xu et al., (depLCNN+NS) [17]	Word vector+shortest dependency paths	85.6%
Zhou et al., (Att-BLSTM) [3]	Word vector	84.0%
Our model	Word vector	85.7%

The follow works are based on the SemEval-2010 task 8：

SDP-LSTM: Yan et al. [2] presented SDP-LSTM to classify the relation of two entities in a sentence. The model leveraged the SDP and multichannel RNNs with 8LSTM units picking up heterogeneous information along the SDP including word representations, part-of-speech tags (POS), grammatical relations and WordNet hypernyms. The model achieved an F1-score of 83.7%.

RNN: Zhang et al. [14] utilized a framework based on RNN and several modifications to enhance the model, including a max-pooling approach and a bi-directional architecture. This model used POS and position features, which achieved an F1-score of 80.0%.

LSTM-RNN: Miwa et al. [5] built an end-to-end LSTM based sequence and tree structured model. They extracted entities via a sequence layer and relations between the entities via the shortest path dependency tree network, which achieved an F1-score of 85%.

CR-CNN: Dos santos et al. [15] tackled the relation classification task using a CNN that performed classification by ranking and proposed a pairwise ranking loss function that made it easy to reduce the impact of artificial classes. CR-CNN achieved an F1-score of 84.1% with using position features.

depLCNN+NS: Xu et al. [16] employed CNN to learn more robust relation representations from the shortest dependency path. Furthermore, they proposed a straightforward negative sampling strategy to improve the assignment of subjects and objects. Their model reached an F1-score of 85.6%.

Att-BLSTM: Zhou et al. [3] proposed attention-based BLSTM to capture the most important semantic information in a sentence. We also use attention mechanism called multi-head attention, which can capture the words that play an import role in the sentence. Comparing with Zhou et al.'s pipeline model only using attention and word vectors, we leverage the end-to-end structure and a multi-head attention mechanism to enhance the effect of emphasis. Most of the above models used handcrafted features (in Table 3), while our model achieves competitive results only using the word vectors and output of the hidden layer of the NER. Our model leverages multi-head attention and entity hidden weights, which achieves an F1-score of 85.7%, without using lexical resources such as WordNet or NLP systems like dependency parser and NER to get high-level features.

4.5 Discussion

From the above section, we know that our method achieves the competitive results on the SemEval-2010 task 8 without feature engineering. The specific results of the analysis are as follows:

1. *Shared the output of hidden layers*. The output of the hidden layers in NER is shared with the relation extraction, which contains contextual information. Therefore, our model achieves competitive results without utilizing any handcrafted features and syntax features.

2. *Multi-head attention.* The model can highlight effective information by multi-head attention. Our multi-head attention can obtain the information from multiple respects compared with other attention mechanisms for relation extraction.

The experimental results show that our model achieves an F1-score of 85.7% on the SemEval-2010 task 8, which has competitive performance without extra feature compared with previous joint models.

4.6 Error analysis

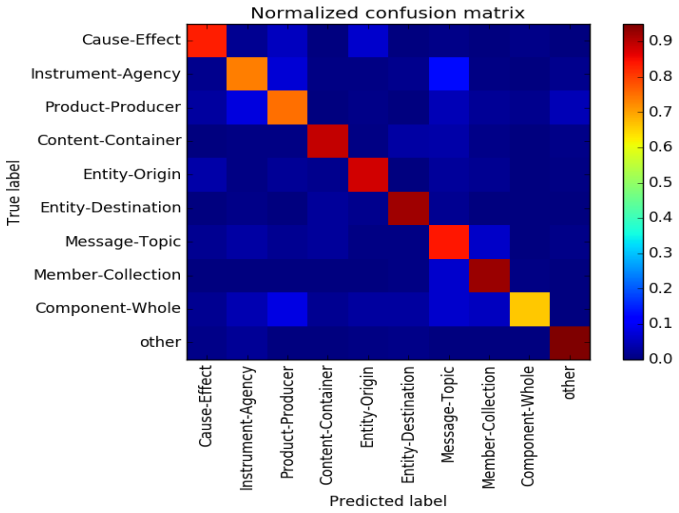

Fig. 2. The distribution of the predicted results for each relation class. The horizontal X-axis represents the predict label and the vertical Y-axis represents the true label.

We visualize the model's predicted results to analyze the errors of our approach as shown in Figure 2. The diagonal region indicates the correct prediction results and the other regions reflect the distribution of error samples. The highlighted region means the higher F-score. In Figure 2 we can find that the F1-scores of relation of "Instrument-Agency" and the "Component-Whole" are lower than the other relations. The analysis for the errors are as follows:

1. The class imbalanced problem is one of the critical factor affecting the results. As shown in Table1, only 6.3% of the instances belong to the relation of 'Instrument-

Agency' in the train set, which is the smallest of all instances. Undersampling and oversampling will be adopted to balance the number of each class in our future works.

2. The lack of information between entities is another problem which leads to bad performance on some instances. For example, in the sentence of "This <e1>bed</e1> <e2>pole</e2> has a hook type handle, and fits under the mattress and provides a firm handle to assist with moving and positioning in bed.", there is no words between entities 'bed' and 'pole'. Due to the deficiency of key words, our model cannot effectively learn useful features for classification, thus such instances are classified by mistake.

5 Conclusion and future work

In this paper, we propose a novel model based on multi-head attention for joint entities and relations extraction. Instead of employing feature engineering, we utilize the hidden layer weights of NER which can supply the information of entities and context in the process of named entity recognition. We also leverage the multi-head attention for relation extraction, which helps our model extract the significant information from different aspects. Our model extracts entities and relations in joint model and does not apply extra handcraft features or NLP tools. The model achieves the competitive results on SemEval-2010 task 8 dataset, which indicates that the effectiveness of the hidden layer weights and multi-head attention. In future works, we will explore the effective way to deal with the problems of class imbalanced and insufficient information.

References

1. Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING, pp. 2335-2344. (2014).
2. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., and Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1785-1794. (2015).
3. Yan, X., Mou, L., Li, G., Chen, Y., Peng, H., and Jin, Z.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 207-212. (2016).
4. Li, Q., and Ji, H.: Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 402-412. (2014).
5. Miwa, M., and Bansal, M.: End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In: Meeting of the Association for Computational Linguistics, pp. 1105-1116. (2016).
6. Miwa, M., and Sasaki, Y.: Modeling joint entity and relation extraction with table representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1858-1869. (2014).

7. Katiyar, A., and Cardie, C.: Going out on a limb: Joint Extraction of Entity Mentions and Relations without Dependency Trees. Meeting of the Association for Computational Linguistics, pp. 917-928. (2017).
8. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., and Xu, B.: Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. arXiv preprint arXiv:1706.05075. (2017).
9. Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., and Szpakowicz, S.: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions, pp. 94-99. (2009).
10. Hochreiter, S., and Schmidhuber, J.: Long short-term memory. Neural computation, 9(8), pp. 1735-1780. (1997).
11. Lafferty, J., McCallum, A., Pereira, F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML. Vol.3, pp. 282-289. (2001).
12. Kingma, D. P., and Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. (2014).
13. Pennington, J., Socher, R., and Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) pp. 1532-1543. (2014).
14. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958. (2014).
15. Zhang, D., and Wang, D.: Relation classification via recurrent neural network. arXiv preprint arXiv:1508.01006. (2015)
16. Santos, C. N. D., Xiang, B., and Zhou, B.: Classifying relations by ranking with convolutional neural networks. arXiv preprint arXiv:1504.06580. (2015)
17. Xu, K., Feng, Y., Huang, S., and Zhao, D. Semantic relation classification via convolutional neural networks with simple negative sampling. arXiv preprint arXiv:1506.07650. (2015).