Character-level Based Conference Named Entity Recognition Using BiLSTM

Rui Xiong[1, a], Chensheng Wu[1, b], Xiong Li[2, c] and Luyu Li[3, d]

[1]Beijing Institute of Science and Technology Information Beijing, China

[2]Information Science Academy of China Electronics Technology Group Corporation Beijing, China

[3]Beijing Institute of Radio Measurement Beijing, China

[a]xiongrui_2418@sina.com, [b]Wu1082@163.com, [c]xiongli892412@163.com, [d]Improve0621@126.com

Abstract. For name entity recognition in specific field, a character-level bidirectional LSTM (Bidirectional Long Short-Term Memory, BiLSTM-based conference named entity recognition method is proposed. Introducing word embedding as the input of BiLSTM, each character in a sentence is mapped from a one-hot vector to a low-dimensional dense character embedding through word embedding. After the BiLSTM layer, viterbi algorithm is used to decode the output of BiLSTM. The results show that the precision rate, the recall rate and F value can reach 89.17%, 90.24% and 89.70%, indicating the effectiveness and practicability of the method of conference named entity recognition for open domain data.

Keywords: Conference named entity recognition, character, BiLSTM, viterbi, open domain.

1. Introduction

Named entity recognition is an important basic tool for high-level applications such as information extraction, syntax analysis and entity linking. It plays an important role in the process of natural language processing technology. In a narrow sense, named entity recognition is the identification of three types of named entities: person name, place name, and organization name (Time, currency name, etc., which have obvious laws, can be identified by regular methods.). In particular fields, various corresponding entity types are defined. More and more researches focus on NER in particular fields such as social media1[1], health-domain [2], astronautics area[3].

In recent years, science and technology is developing rapidly. Academic conferences cover a large number of cutting-edge information, but a large amount of conference information is drowning in the vast amount of text on the Internet. Therefore, conference named entity recognition can help automatically extract valuable information and get the latest science and technology information or advanced technology information in time.

The earliest named entity recognition method is based on manual rules[4]. The rule-based approach has great limitations in terms of generalization. With the deep research, the named entity recognition method based on statistical model and machine learning model has become mainstream. Typical models include hidden markov model (HMM)[5], maximum entropy[6], conditional random fields(CRF) [7] and so on.

With the development of named entity recognition, the academic community has applied the current deep learning technique to named entity recognition, such as convolutional neural network (CNN)[8], recurrent neural network(RNN)[8]. The LSTM model that solves the long-dependency problem of RNN has been proved to have a better performance in named entity recognition. But it may isolate contextual information. What's more, existing methods are mostly based on word-level. It may lead to the failure of handling unregistered words. In this paper, a character-level bidirectional LSTM method is proposed. And viterbi algorithm is used to decode the output of BiLSTM. This method can process open-domain data.

2. BiLSTM Neural Network Model

2.1 LSTM

The application of recurrent neural networks (RNNs) is used to process sequence data. The RNN network can remember the previous information and apply it to the calculation of the current output. Nevertheless, RNN has problems with gradient explosion and gradient disappearance when dealing with language models. To solve this problem, Hochreiter and Schmidhuber proposed the LSTM network, which is a special form of RNN. LSTM enables efficient use of long-distance information and solve the problem of gradient disappearance by adding memory gate unit and threshold limits.

Fig. 1 shows a LSTM unit[9]. The Forgotten Gate *f* indicates the input at the current time, which determines the discarded portion of the information that was sent from the previous moment. i represents the input gate, which determines which values should be updated at time t. $\tilde{c}$ is a vector of candidate values. Combine *i* and $\tilde{c}$ to get C to update the state of the neuron. *o* is the output layer. *h* is the output of the entire network. At time step t, given input *x*, the specific calculation process of the output of the hidden layer of LSTM is as follows:

$$i_t = \sigma(W_i[h_{t-1},x_t] + b_i) \tag{1}$$

$$f_t = \sigma(W_f[h_{t-1},x_t] + b_f) \tag{2}$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1},x_t] + b_c) \tag{3}$$

$$o_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$

$$h_t = o_t \odot \tanh(c_t) \tag{5}$$

where, W is weight matrix that connects two layers, b_i,b_f,b_c are bias vectors.

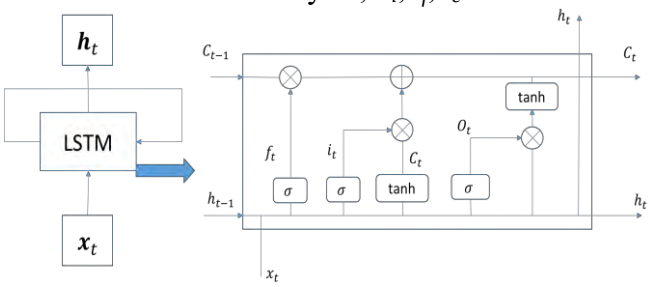

Fig. 1 The LSTM unit

2.2 BiLSTM

The LSTM model neuron information can only be passed from front to back, which means that the input information at the current moment can only use the previous information. For a sequence labeling task, however, the state before and after the current state should be equal. BiLSTM can take advantage of both the information before the current moment and the information after it, which is very suitable for named entity recognition tasks.

Fig. 2 shows the structure of Bilstm. For each x_t, the corresponding word vector is x_t. The character embedding sequence (x_1, x_2, ..., x_n) of each word of a sentence is used as input for each time step of the bidirectional LSTM. Then, the hidden state sequence $(\overrightarrow{h_1},\overrightarrow{h_2},...,\overrightarrow{h_n})$ of the forward LSTM output and the hidden state sequence ($\overleftarrow{h_1},\overleftarrow{h_2},...,\overleftarrow{h_n}$) of the backward LSTM are spliced at their respective output positions.

A complete hidden state sequence $(h_1,h_2,...,h_n) \in R^{n\times m}$ is obtained. Then a matrix $P = (p_1,p_2,...,p_n) \in R^{n\times k}$ is obtained through a linear layer so that the hidden state vectors are mapped from m-dimension to k-dimension. k is tag number. Each dimension p_{ij} of $p_i \in R^k$ can be regarded as a score value that classifies the word x_i into the jth label. After performing

softmax on ***P***, the corresponding tags can be obtained. But it is equivalent to classifying each location independently. Softmax ignores contextual characteristics information. In this paper, viterbi algorithm is applied to decode matrix ***P***.

3. Conference Named Entity Recognition Model

In this paper, conference named entity recognition is regarded as a character-level sequence labeling problem. Therefore, it can be seen as multi-classification problem. Each character in sentences is classified into 4 categories (B-con, M-con, E-con, O-con). B-con represents the first character of conference name. M-con represents the middle character of conference name. E-cons is the last character of conference name. O-con stands for character that is not related to the conference name. The character-based Conference named entity recognition model based on BiLSTM is mainly composed of three parts：

Text vectorization layer;

BiLSTM layer;

Label inference layer based on Viterbi.

The model structure is shown as Fig. 2.

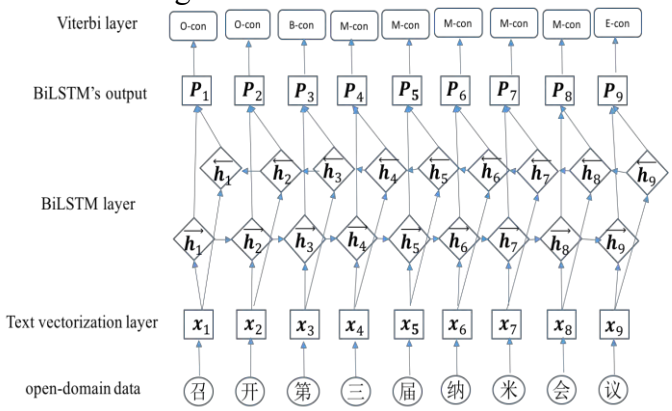

Fig. 2 The structure of conference named entity recognition model

3.1 Text Vectorization Layer

To use the neural network model to process data, the input data should be first vectorized. Previous studies have shown that adding pre-trained word embedding vectors can improve the performance of natural language processing tasks. Word embedding is also called distributed representation of words. The words' semantic relationship can be represented by word embedding.

A sentence containing n words is represented as sequence:

$$x = (x_1, x_2, x_3, ..., x_n) \tag{6}$$

where, x_i is the id of the ith word of sentence in dictionary. The one-hot vector of each word can be obtained. Dimension is the dictionary size. In the one-hot vector representation, any two words are isolated and unconnected. Each character x_i in a sentence is mapped from a one-hot vector to a low-dimensional dense character embedding $\boldsymbol{x}_i \in R^d$ by training neural network language model. d is the embedding dimension.

3.2 BiLSTM Layer

BiLSTM neural network layer consists of two parts: a forward LSTM and a backward LSTM. According to section II-B. The char embedding sequence ($\boldsymbol{x_1}$, $\boldsymbol{x_2}$, ..., $\boldsymbol{x_n}$) of a sentence is the input for each time step of the bidirectional LSTM. After the BiLSTM layer, the matrix ***P*** that shows the score of each character on each tab is obtained.

3.3 Viterbi Decoding Layer

The Viterbi algorithm can actually be regarded as the shortest path problem for the lattice network as shown in fig.3. In this paper, the lattice network can be presented as figure. There are four states (B-con, M-con, E-con, O-con) at each timestep t. What needs to be emphasized is that the first state

can only be B-con and O-con. The probability of each state at each timestep $P_{ij}(i = \mathrm{B-con, M-con, E-con, O-con}; j = 1, 2, ..., t-1, t, ..., n)$ is shown in matrix ***P***. According to Viterbi algorithm, the most probable path probability of reaching state S is

$$P_r(S\ at\ time\ step\ t) = max_{i=\mathrm{B-con, M-con, E-con, O-con}}\{P_r(i\ at\ time\ step\ (t-1))\} \times P_r(S/i) \times P_r(obs.S\ at\ time\ t) \quad (7)$$

where,

$max_{i=\mathrm{B-con, M-con, E-con, O-con}}\{P_r(i\ at\ time\ step\ (t-1))\}$ is the probability of local optimal path at time t-1; $P_r(S/i)$ is the transition probability; $P_r(obs.S\ at\ time\ t)$ is the observation probability P_{ij}; S is any state of the four states.

Then, calculate all $P_r(S\ at\ time\ step\ t)$, the maximum is the the probability of optimal path. The optimal state combination is the result of the annotation.

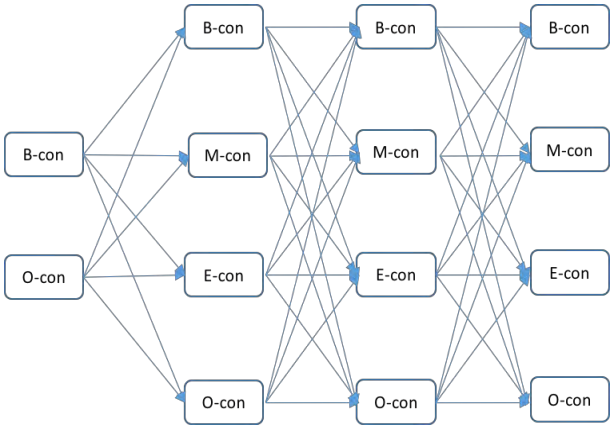

Fig. 3 The lattice network

4. Experiment and Analysis

4.1 Corpus Preparation

Machine learning-based methods for conference-named entity recognition require a corresponding corpus for training and testing. In this paper, news reports is used as corpus for conference-named entity recognition. In the experiment, select the report related to the conference name as the source of the corpus. The size of the training corpus is 2898kB. The corpus contains 2897085 characters.

4.2 Test Set

The test set in this paper can be open domain. Input any text file, in which the conference named entity can be identified and extracted. In this paper, the test set is 105kB. It contains 435 conference named entities.

4.3 Experimental Parameter Setting

Through multiple experiments to optimize the parameters, we finally set the parameters as follows: The initial learning rate is set to 0.1; the minimum batch size is 20; The number of hidden layer nodes is set to 128 and the dimension of the word embedded vector is 128. The BiLSTM layer number is 2. The drop rate for Dropout is 0.2.

4.4 Performance Comparison Between BiLSTM and LSTM

In the conference-named entity recognization, whether BiLSTM is superior to LSTM performance is verified. Replace the BiLSTM layer in figure with LSTM. The experimental results are shown in Table I. BiLSTM's precision, recall and F values were 4.82%, 4.87% and 4.49% higher than LSTM, respectively. BiLSTM performs better than LSTM in conference named entity recognition. LSTM is a one-way mechanism that only contains the pre-text information of the current word of the lexical sequence of the event sentence, but does not involve the following information. BiLSTM adds a reverse LSTM based on LSTM. The forward LSTM is used to capture the above feature information. The reverse LSTM is used to capture the following feature information, and then get global contextual

information through the fusion of the above feature information and the following feature information.

Table 1 Performance comparison between BiLSTM and LSTM

	P	R	F
LSTM	84.35%	85.37%	85.21%
BiLSTM	89.17%	90.24%	89.7%

（P: Precision value; R: Recall value; F: F value）

4.5 Performance Comparison Between BiLSTM+Softmax and BiLSTM+Viterbi

To compare the performance of BiLSTM+softmax and BiLSTM+Viterbi, replace the Viterbi layer in figure with softmax layer. The experimental results are shown in Table II. BiLSTM+viterbi's accuracy, recall and F values were 12.16%, 13.97% and 13.43% higher than LSTM, respectively. BiLSTM+viterbi performs better than BiLSTM+softmax in conference named entity recognition. Softmax isolates the connection between characters and characters so that it lost a lot of contextual feature information. Viterbi algorithm can get contextual information as the transition probability contains the contextual information.

Table 2 Performance comparison between BiLSTM+softmax and BiLSTM+viterbi

	P	R	F
LSTM	77.01%	76.27%	76.27%
BiLSTM	89.17%	90.24%	89.70%

In general, the method proposed in this paper can effectively recognize the conference named entities, and the experimental effect is obvious.

5. Conclusion

Conference named entity recognition is beneficial to automatically extracting valuable information and getting the latest information or advanced technology information. A character-level bidirectional BiLSTM-based conference named entity recognition method is proposed. And viterbi algorithm is used to decode the output of BiLSTM. The character-level-based method solves the problem that unregistered word can not be handled. The BiLSMT+Viterbi method solves the problem of ignoring contextual information. And this conference named entity recognition is suitable for open-domain data. The experimental results show the effectiveness of the proposed method.

Acknowledgements

This paper is sponsored by Beijing Science and Technology Research Institute sprouting plan GS201813.

References

[1]. H.F. He and X. Sun, “A unified model for cross-domain and semi-supervised named entity recognition in Chinese social media,” Proceedings of 31st AAAI Conference on Artificial Intelligence, San Francisco, pp3216-3222, February, 2017.

[2]. I.J. Unanue, E.Z. Borzeshi, M. Piccardi, “Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition,” Journal of Biomedical Informatics, vol.76, pp.102-109, 2017.

[3]. J.Z. Xu, J. Zhu, R. Zhao, L. Zhang and J.J. Li, “Astronautics named entity recognition based on CRF algorithm,” Electronic design engineering, vol.25, no.20, pp42-46, Oct.2017.

[4]. G. alph,"The NYU system for MUC—6 or where’s the syntax,” Proceedings of the Sixth Message Understanding Conference, Morgan Kaufmann, November 1995.

[5]. S.Y. Sun and Q.X. Miao, “Text information extraction algorithm based on improved SVM and HMM,” Computer application and software, vol.32, no.11, pp.281-284, Nov. 2015.

[6]. X.Y. Zhang, T. Wang and H.W. Chen, “A mixed statistical model-based method for Chinese named entity recognition,” Computer engineering & science, vol.28, no.6, pp.135-139, June, 2006.

[7]. C.S. Zhang, J.Y. Guo, Y.T. Xian, Z.T. Yu, C.Y. Lei and H.X. Wang, “Named entity recognition of the products with English based on conditional random fields,” Computer engineering & science, vol.32, no.6, pp115-117, June, 2010.

[8]. L.H. Li, Y.K. Guy, “Biomedical Named Entity Recognition with CNN-BLSTM-CRF,” Journal of chinese information processing, vol.32, no.1, pp.116-121, Jan.2018.

[9]. C. Jin, W.H. Li, C. Ji, X.Z. Jin and Y.B. Guo, “Bi-directional long short memory neural networks for Chinese segmentation,” Jounal of Chinese information processing, vol.32, no.2, pp.29-37, Feb. 2018.