Chinese Named Entity Recognition with Changed BiLSTM and CRF

Jie Ren(✉), Jing Liang, and Chenkai Zhao

University of Electronic Science and Technology of China, Chengdu, China
2356670185@qq.com

Abstract. This paper is aimed at improving name entity recognition (NER) accuracy. We replace the traditional bidirectional long short-term memory network (BiLSTM) with a changed BiLSTM, and then uses a CRF layer behind the changed BiLSTM layer to add the probabilistic relation of different Chinese characters.

Keywords: NER · CRF · Changed BiLSTM

1 Introduction

Name entity recognition (NER) is also known as proper name recognition. It is a fundamental task in natural language processing (NLP) and has a wide range of applications. Named entity generally refers to the entity with specific meaning or strong reference in the text, usually including person name, place name, organization name, and so on. NER is a foundation key task in NLP. At the same time, NER is also the basis of many NLP tasks such as relationship extraction, event extraction, knowledge mapping, machine translation, and question answering system.

NER has been a research hotspot in the field of NLP. At beginning, NER is based on dictionary and rule-based methods. Later, the traditional machine learning methods, especially probabilistic graph models such as hidden Markov model (HMM) [1,2], maximum entropy Markov model (MEMM), and conditional random field (CRF) [3,4], became the focus of NER's research. In recent years, NER based on deep learning such as long short-term memory (LSTM) has been become popular [5]. However, both probability graph model and deep learning model are incomplete. Probability graph model does not learn the deep hidden information in sentences and deep learning model does not consider the probabilistic relation between parts of speech and the parts of speech. Some scholars found that the combination of two methods deals with the above problems [8]. Huang Z, et al. used bidirectional long short-term memory (BiLSTM) and CRF to do NER, and the results is very good [6,7]. Then, Lample G, et al. proposed that use two LSTM and CRF can obtain deeper hidden information

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 398–406, 2020.
https://doi.org/10.1007/978-981-15-0187-6_47

and a better accuracy [9]. But their method did not consider the reverse meaning of texts. Also, Zheng S, et al. used a BiLSTM network to deal with this problem and a LSTM network to replace CRF to gain the deeper semantic [10], but they did not consider the probabilistic relationship.

To solve these problems, we propose an advanced BiLSTM network to achieve better deep semantics and probability relationships. We use two LSTM to train a sequence and return a sequence. What is more, we use two reverse LSTM to train the same sequence and return a sequence too. Then, we concatenate the two sequences to one sequence. Hence, we can obtain deep semantics and the relationships including the forward meaning and backward meaning together. Finally, we use a CRF to obtain the probabilistic relation.

In this paper, in Sect. 1, we introduce the background and the overall idea. In Sect. 2 , we introduce the important models in our algorithm. In Sect. 3 , we give out our experiment results. In Sect. 4, a conclusion is given.

2 Mathematic Models

2.1 Improved BiLSTM

The Disadvantage of RNN: Recurrent neural networks (RNN) are implemented by reusing a cell structure. Hence, the output of the current moment is related to the past.

The expression function of RNN is:

$$h_t = f(W \cdot x_t + U \cdot h_{t-1} + b), \tag{1}$$

$$y_t = V \cdot h_t, \tag{2}$$

where x_t and y_t are the input and output at time t, respectively. h_t is the memory information at time t, and $f(z)$ is an activation function, which is usually a tanh function. W, U, b, v are the parameters of the network.

Because RNN is implemented by reusing a cell structure, we just train W, U, b, v by iterations and do not need other parameters.

However, there is a disadvantage of RNN. It can not remember long time information because of gradient diffusion. The appearance of LSTM solved this problem.

The Advantage of LSTM: LSTM is a variant of RNN, which can effectively solve the gradient diffusion problem of simple RNN. It mainly improves the following two parts.

First, it adds a new internal state c_t and retains the original external state h_t. Hence, gradient diffusion is suppressed by combining linearity and nonlinearity. Second, it controls the amount of information transmitted through three gates, ensuring that the linear transmission does not lead to too much information. Therefore, its formulas are written as:

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t, \tag{3}$$

$$h_t = o_t \odot \tanh(c_t), \tag{4}$$

where f_t, i_t, o_t are forgotten gate, input gate, output gate, respectively. They are from 0 to 1. $\odot$ means the product of the value of the gate and each element in the vector. $\hat{c}_t$ is the candidate states obtained by nonlinear functions. The details are shown as follows:

$$\hat{c}_t = \tanh(W_c \cdot [x_t, h_{t-1}] + b_c), \tag{5}$$

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i), \tag{6}$$

$$f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f), \tag{7}$$

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o), \tag{8}$$

where $[x_t, h_{t-1}]$ means to concatenate x_t and h_{t-1}, together. σ is sigmoid function:

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}. \tag{9}$$

Because of these three gates and the internal state, LSTM can successfully remember the past information and delete the useless information.

Although LSTM is the most popular network in dealing with time sequence problems, it also has a small shortcoming. It just learn the forward information but does not learn the backward information, which has limitations in the language model.

The Meaning of BiLSTM: As we just mentioned above, sometimes, we need a network to study the forward information and backward information together, especially in text problem such as text classification, text translation, and NER.

BiLSTM is simple to understand. It has two LSTMs. The first one's input is from x_1 to x_T and its external state is from h_1^1 to h_T^1. The second one's input is from x_T to x_1 and its external state is from h_1^2 to h_T^2. Later, the concatenation of these two LSTMs' external state is $h_t = [h_t^1, h_t^2]$.

Our Changed BiLSTM: In this paper, in order to get the deeper information of sentences and the forward and backward text connection, we change the BiLSTM structure.

We divided the model into two parts. The first part is a forward LSTM connected to another forward LSTM. The first forward LSTM's input is the input data, and the second forward LSTM's input is the returned sequence from the first forward LSTM. The second part is a backward LSTM connected to another backward LSTM. Also, the first backward LSTM's input is the input data, and the second backward LSTM's input is the returned sequence from the first backward LSTM.

By using this structure, we can firstly get deeper meanings of forward and backward text sequences. Later, we combine the two states together and use the combinative result as the CRF's input.

2.2 Conditional Random Filed

The Model Definition: CRF is a conditional probability distribution model of one set of output variables with another set of input variables that are given. It is characterized by the assumption that the output random variables constitute Markov conditional field.

The Parametric Formalization: If $P(Y|X)$ is linear CRF and the value of random variable X, x is given, the value of random variable Y, y is shown as follows:

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_l, x, i)\right), \tag{10}$$

where t_k and s_l are characteristic functions which are always 0 or 1, λ_k and μ_l are corresponding weights, $Z(x) = \sum_y \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_l, x, i)\right)$ is a normalization factor.

The parameters are learned by BFGS method, which is not described here.

The Prediction Algorithm: The CRF prediction problem is to find the output sequence y^* with the maximum conditional probability given the CRF $P(Y|X)$ and the input sequence x. This problem is usually solved by viterbi algorithm. Hence, the output sequence is written as:

$$y^* = \arg\max_w P_w(y|x) = \arg\max_w \frac{\exp(w \cdot F(y, x))}{Z_w(x)}, \tag{11}$$

and because $Z_w(x)$ is a constant and exp(.) is a monotone increasing function. Therefore,

$$y^* = \arg\max_y (w \cdot F(y, x)) \tag{12}$$

So the CRF prediction problem is called the optimal path problem with the largest probability of denormalization.

$$\max_y (w \cdot F(y, x)), \tag{13}$$

where the path means the tag sequence and

$$w = (w_1, w_2, ..., w_K)^{\mathrm{T}}, \tag{14}$$

$$F(y, x) = (f_1(y, x), f_2(y, x), ..., f_K(y, x))^{\mathrm{T}}, \tag{15}$$

$$f_k(y, x) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, x, i), \quad k = 1, 2, ..., K. \tag{16}$$

To solve the optimal path, (13) can be written as:

$$\max_y \sum_{i=1}^{n} w \cdot F_i(y_{i-1}, y_i, x), \tag{17}$$

where $F_i(y_{i-1}, y_i, x) = (f_1(y_{i-1}, y_i, x, i), f_2(y_{i-1}, y_i, x, i), ..., f_K(y_{i-1}, y_i, x, i))^{\mathrm{T}}$ is local characteristic vector.

Later, use viterbi algorithm. First of all, the denormalized probability of each mark $j = 1, 2..., m$ to position 1 is:

$$\delta_1(j) = w \cdot F_1(y_0 = \text{start}, y_1 = j, x), \quad j = 1, 2, ..., m. \tag{18}$$

In general, the recursive formula is used to find the maximum value of the denormalization probability of each mark $l = 1, 2, ..., m$ to position i:

$$\delta_i(j) = \max_{i \le j \le m} \delta_{i-1}(j) + w \cdot F_i(y_{i-1} = j, y_i = l, x), \quad l = 1, 2, ..., m, \tag{19}$$

$$\Phi_i(l) = \arg \max_{i \le j \le m} \delta_{i-1}(j) + w \cdot F_i(y_{i-1} = j, y_i = l, x), \quad l = 1, 2, ..., m. \tag{20}$$

When $l = n$, the maximum value of the denormalized probability is:

$$\max_y(w \cdot F(y, x)) = \max_{1 \le j \le m} \delta_n(j), \tag{21}$$

and the most optimal terminal:

$$y_n^* = \arg \max_{1 \le j \le m} \delta_n(j) \tag{22}$$

Return from the end of this optimal path:

$$y_i^* = \Phi_{i+1}(y_{i+1}^*), \quad i = n-1, n-2, ..., 1, \tag{23}$$

Finally, the most optimal path is $y^* = (y_1^*, y_2^*, ..., y_n^*)^{\mathrm{T}}$.

2.3 Our Model Structure

This part totally explains what we do and how to combine improved BiLSTM and CRF together.

Figure 1 is the whole structure of our network.

Firstly, because all the words in corpus are in the form of one-hot encoder and the dimension is very high, we use an embedding layer to find the words' low-dimension representation to reduce the learning time. Also, the low-dimension dense representation has better representation ability than the high-dimension sparse representation.

Secondly, we put the embedded words sequence into our improved BiLSTM to learn the forward and backward deep semantic meaning. Then, use a merge layer to concatenate the external state of the two-layer-forward LSTM and the external state of the two-layer-backward LSTM.

Thirdly, the combined external state is regarded as the input of CRF layer. CRF layer learns its parameters.

Finally, the output sequence of CRF layer is the name entities.

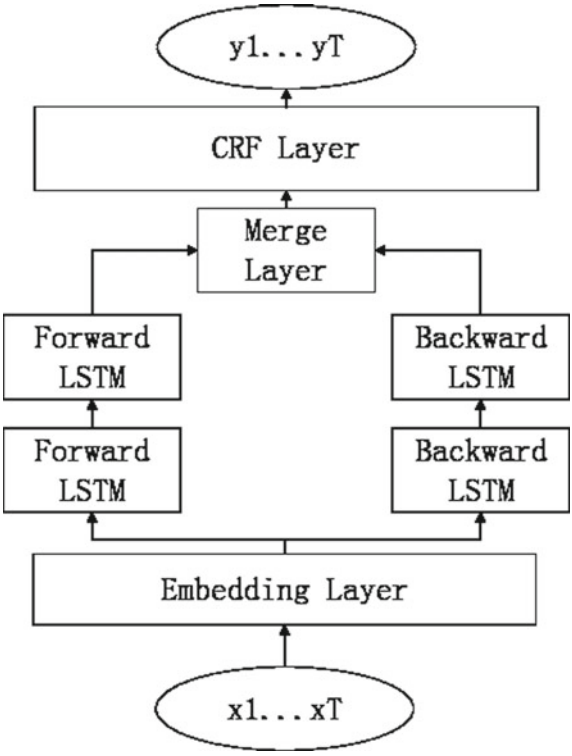

Fig. 1. Whole structure of our model

3 Experiment and Results

3.1 The Form of Experiment Data

In order to show whether our algorithm is good or not, we do an experiment with real data.

We download an open-source data of Chinese NER from the Internet. The data form is as following. Each sentence is separated by "\n\n". Each pair of Chinese character and NER label is separated by "\n". Each element (Chinese character and NER label) in the pair is separated by "\t". For example, "中\tB-LOC\n国\tI-LOC\n很\tO\n大\tO\n\n". And the labels are "B-PER", "I-PER", "B-LOC", "I-LOC", "B-ORG", "I-ORG", "O".

3.2 Data Preprocessing and Model Compile

Obviously, this data can not be used directly. There must be some preprocessing.

First of all, we combine the characters in each sentence as an input sequence, and we combine the NER labels in each sentence as an output sequence. However, the characters can not be used in a mathematic model. Hence, we count

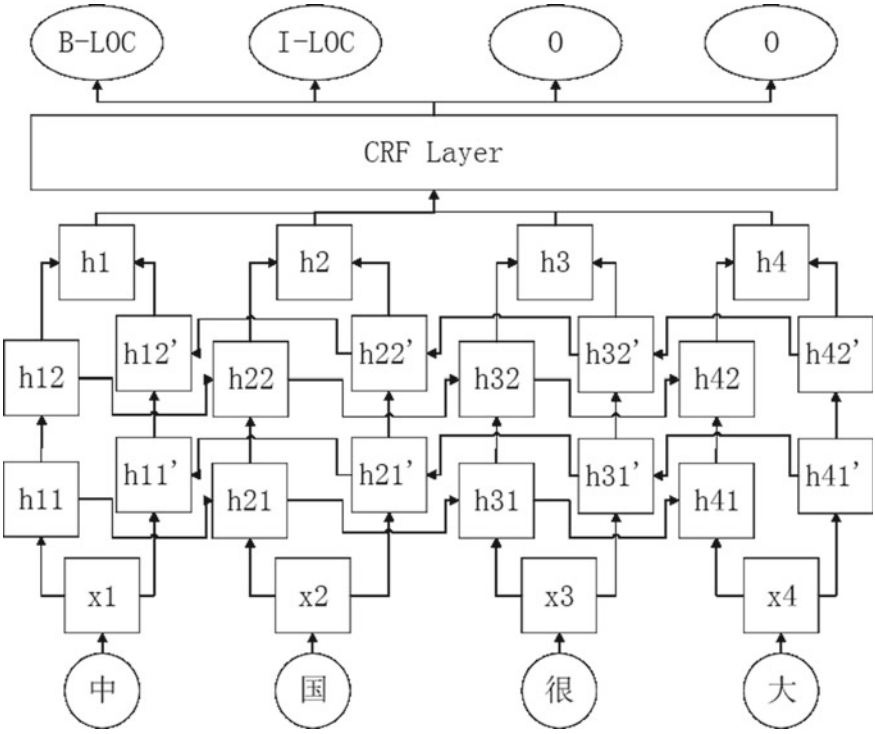

Fig. 2. Simple example of our experiment

the number of characters that appear and change each character to its one-hot encoder form.

Later, LSTM should have a fixed input dimension, which means that the input sequences should have the same length. So, we find the longest sequence and use zero-padding to make the shorted sequence has the same length as the longest one.

Finally, we use Keras to set up our model, and Fig. 2 is a simple example of our model.

In our experiment, we use 50658 training data and 4631 test data. The output embedding dimension is 200, and LSTM has 128 units. Also, there are batch normalization layer and dropout layer. The "categorical cross entropy" is the training loss and metrics, and we use "adam" as our optimizer.

3.3 Experiment Result

After training our model with, we firstly use a new text to predict whether our model has the ability to successfully analyze name entities. Here, we use a sentence "中华人民共和国国务院总理周恩来在外交部长陈毅的陪同下，连续访问了埃塞俄比亚等非洲10国以及阿尔巴尼亚", and the result is that "B-PRE" + "I-PER": ["周恩来","陈毅"] "B-LOC" + "I-LOC": ["埃塞俄比亚","非洲","阿尔巴尼亚"]; "B-ORG" + "I-ORG": ["中华人民共和国国务院","外交部"]. We can see that the result is totally correct.

Table 1. Comparison result of different algorithms.

Model	CRF viterbi accuracy (%)
Embedding + LSTM	95.71
Embedding + CRF	96.57
Embddubg + BiLSTM	96.61
Embddubg + BiLSTM+CRF	97.41
Our model	97.87

Then, we compared our algorithm with the traditional algorithms by using all test data, and the result is shown in Table 1.

From Table 1, we can find that our model has the best accuracy than other models.

However, because our model has deeper structure than traditional models, our training time is longer than other methods. Therefore, we add batch normalization layers and dropout layers to reduce training time. Also, we reduce the amount of units in LSTM layers appropriately. Finally, the comparison of training time is shown in Table 2.

Table 2. Comparison of training time

Model	Training time (mins)	CRF viterbi accuracy (%)
BiLSTM + BN + Dropout	126	96.59
BiLSTM + CRF + BN + Dropout	142	97.46
Our model	164	97.87
Our model + BN + Dropout	148	97.91

Table 2 shows that our model with BN and dropout layers can use the same training time as the traditional models to get a better CRF viterbi accuracy.

4 Conclusion

In order to solve the problem that traditional BiLSTM can not obtain deeper latent semantics, we change the BiLSTM model, and successfully combine it with a probabilistic graphic model (CRF layer) to gain a better prediction accuracy in NER problem. After experiments, we can find that it is useful in Chinese NER question, and it has the highest CRF viterbi accuracy.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (61731006, 61671138), and was partly supported by the 111 Project No. B17008.

References

1. Zhao S (2004) Named entity recognition in biomedical texts using an HMM model. In: International joint workshop on natural language processing in biomedicine and its applications. Association for Computational Linguistics
2. Ponomareva N , Pla F , Molina A et al (2007) Biomedical named entity recognition: a poor knowledge HMM-based approach
3. Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT Press
4. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: International conference of machine learning
5. Zhang Y , Yang J (2018) Chinese NER using lattice LSTM
6. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. Comput Sci
7. Xu C, Xie L, Xiao X (2018 July) A bidirectional LSTM and conditional random fields approach to medical named entity recognition 90(7):1063–1075
8. Ma X, Hovy E (2016) End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF
9. Lample G, Ballesteros M Subramanian S et al (2016) Neural architectures for named entity recognition
10. Zheng S, Wang F, Bao H et al (2017) Joint extraction of entities and relations based on a novel tagging scheme