Named Entity Recognition with Gating Mechanism and Parallel BiLSTM

Yenan Yi* and Yijie Bian

School of Business, Hohai University, Nanjing, 211106, China
E-mail: yi_yenan@hhu.edu.cn; byj@hhu.edu.cn
**Corresponding Author*

Received 18 March 2021; Accepted 19 May 2021;
Publication 24 June 2021

Abstract

In this paper, we propose a novel neural network for named entity recognition, which is improved in two aspects. On the one hand, our model uses a parallel BiLSTM structure to generate character-level word representations. By inputting character sequences of words into several independent and parallel BiLSTMs, we can obtain word representations from different representation subspaces, because the parameters of these BiLSTMs are randomly initialized. This method can enhance the expression abilities of character-level word representations. On the other hand, we use a two-layer BiLSTM with gating mechanism to model sentences. Since the features extracted by each layer in a multi-layer LSTM from texts contain different types of information, we use the gating mechanism to assign appropriate weights to the outputs of each layer, and take the weighted sum of these outputs as the final output for named entity recognition. Our model only changes the structure, does not need any feature engineering or external knowledge source, which is a complete end-to-end NER model. We used the CoNLL-2003 English and German datasets to evaluate our model and got better results compared with baseline models.

Keywords: Named entity recognition, parallel BiLSTM, gating mechanism, CoNLL-2003.

Journal of Web Engineering, Vol. 20_4, 1157–1176.
doi: 10.13052/jwe1540-9589.20413

1 Introduction

The main task of named entity recognition (NER) is to label each word in sentences with correct entity types, which usually include person, place, organization, etc. NER is a very important part of many natural language processing tasks. A high quality recognition result is very important for the following tasks such as information extraction and automatic question answering.

NER is usually regarded as a sequence labelling task. There are two main methods: traditional statistical machine learning models and neural network models. The statistical models mainly include Hidden Markov Models (HHM) and Conditional Random Fields (CRF). Earlier researches were mostly based on the CRF, and had achieved many good results by manually extracting data features and combining with external knowledge sources in specific fields [1–7]. Although these methods can achieve good results in some specific fields, once it is applied in other languages, text data or named entity types, the previous extracted features cannot be used any more, it needs to manually re-extract data features, which will undoubtedly increase the application cost of the model.

In recent years, neural network models have developed rapidly, and made a lot of achievements in the field of natural language processing. It can automatically extract features from texts and represent them in the form of word embeddings [8, 9]. Therefore, more and more researchers begin to use some powerful neural network models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short term memory (LSTM), gate recurrent unit (GRU) and so on, to study named entity recognition [10–15]. From the experimental results, the performances of these neural network models have gradually been better than the traditional statistical machine learning models, and become the main research direction in the future.

Therefore, we propose a novel neural network model for named entity recognition. Our model does not need any feature engineering and domain specific knowledge source, and can be applied to NER tasks in different languages, which means our model is a end-to-end NER model. First of all, we build a parallel BiLSTM structure to generate character-level word representations. Then, we concatenate the obtained character-level word representations and the pretrained word embeddings, and input them into a two-layer BiLSTM with gating mechanism to model sentences. This gating mechanism enables the model to learn how to select information from these outputs of different layers for calculation. Finally, the weighted sum of these outputs is fed into a CRF layer for named entity recognition. We applied this novel neural network model to the English and German datasets provided by the shared task of CoNLL-2003 [16] and got better results compared with baseline models.

2 Related Work

In recent years, many researchers have used various types of neural network models to study named entity recognition. Huang et al. [11] took pretrained word embeddings as inputs, used BiLSTM and CRF to jointly decode the sequence labels and achieved good results. However, this method did not use character-level word representations, so it cannot make good use of some character-level features (such as prefixes and suffixes, uppercases and lowercases, etc.) contained in words. At the same time, they combined some manually extracted features to improve the final result, which made their model not a truly end-to-end system. Santos and Guimarães [10] proposed to use both character-level and word-level representations to improve the performance. Chiu and Nichols [12] used CNNs to extract character-level word features, and combine with pretrained word embeddings, input them into BiLSTM for named entity recognition. They also used some external knowledge sources. Ma and Hovy [15] adopted similar methods, they used CNNs and BiLSTM to model character-level vectors and pretrained word vectors, add a CRF layer for joint labelling, and achieve good results. Considering the characteristics of LSTM, it is easier to remember the contents close to the current input. Therefore, Lample et al. [13] used BiLSTM to generate character-level word representations. This method can better represent prefixes and suffixes of words through a bi-directional LSTM. In Peters et al.'s researches [17, 18], they used pretrained word embeddings as the input of neural network models, at the same time, combined with the outputs of a bi-directional language model, this method can get different word representations in different contexts, and obtain state-of-the-art NER performance.

In the above researches, for word-level embeddings, most researchers use word vectors which are pretrained in large corpus to initialize the lookup table. From Collobert et al.'s research [19], we can see that these pretrained word embeddings contain potential context semantic information, which can significantly improve the NER result compared with randomly initialized word embeddings. For character-level word representations, they contain more fine-grained morphological features of words, which can effectively solve the problem of insufficient training of rare words, and also have a significant impact on the final NER result. However, for character embeddings, the usual approach is still to use random values for initialization. If we only use a single CNNs or BiLSTM to generate character-level word representations, the quality of the initial values may have an impact on the result. Therefore, we refer to the parallel RNN structure proposed in previous researches [20, 21]. They split the data and input them into multiple independent small-scale parallel RNNs, which can reduce the total amount of parameters and improve the final performance. A little different from their methods, we do not split the character sequences of words, but input the complete character sequences into multiple parallel RNNs, that is, each small-scale RNN accepts the same input, and then we use their outputs to generate the character-level word representations, so as to enhance the expression abilities of the final word representations.

In addition, Belinkov et al.'s research [22] shows that in part of speech tagging tasks, the performance of text representations learned in the first layer is better than that learned in the second layer of a two-layer LSTM model. Peters et al.'s research [18] shows that the higher layers of LSTM units mainly capture the context semantic information of words, while the lower layers of LSTM units mainly learn the syntactic information in texts. From the above researches, we can see that the information learned in each layer of a multi-layer LSTM is different. However, most of the existing named entity recognition researches use a single-layer BiLSTM to model sentences. Therefore, the other research content of this paper is to build a two-layer BiLSTM with gating mechanism for named entity recognition, which uses the output of each layer to jointly model sentences, so as to improve the model performance.

3 Model Structure

Before introducing the NER model proposed in this paper, we first review the traditional BiLSTM-CRF model in Section 3.1, and then present the proposed model in following sections and discuss its advantages.

3.1 BiLSTM-CRF

Recurrent neural networks (RNNs) are commonly used neural networks for modeling sequence information, which can process sequence data iteratively along the time dimension. This recurrent structure allows RNNs to effectively use the information from earlier time to calculate the output of the current time. The information flow can be summarized as follows. The output of the current state is calculated by combining the input of the current state with the output of the previous state. Although RNNs can theoretically process sequence data of arbitrary lengths, but as the data length grows, the impact of the earliest input information on the subsequent output becomes less and less. In order to solve this long-term dependence problem, Hochreiter and Schmidhuber [23] proposed the Long Short-Term Memory network (LSTM). The LSTM cell has three gate structures: an input gate, a forget gate and a output gate. Through these three gate structures, the input information can be selectively stored or discarded, which can be used to solve the long-term dependence problem.

For most sequence labelling tasks, it is helpful to use both the information on the left (past) and the information on the right (future) of words for the final result. A single LSTM can model sentences from left to right according to people's normal reading habits, and use the past information to represent the current content. Then, adding another LSTM according to the reverse direction can use the future information to represent the current content. Combining these two independent LSTMs is called bi-directional LSTM (BiLSTM) [24]. In application, the forward and backward outputs are usually concatenated as the final output of the model.

We can simply input word representations into BiLSTM to model sentences, and then the extracted features are fed into a softmax layer, which can be used to predict the label of each word in the sentences. This method can succeed in some relatively simple sequence labelling tasks (such as part of speech tagging). However, in the task of named entity recognition, there are often strong dependencies between labels. For example, in the IOBES labelling scheme, the label after I-LOC cannot be I-PER or B-ORG, it can only be I-LOC or E-LOC. Using BiLSTM and softmax layer can only model each word independently, and cannot use the dependency information between labels. Therefore, while using BiLSTM to extract features, we combine a conditional random field (CRF) [25] to decode sequence labels, because the CRF encourages the model to predict label sequences that conform to the dependencies between labels.

3.2 Generating Character-level Word Representations by Parallel BiLSTM

Many previous researches used CNNs to extract morphological information of words [12, 15, 26], but CNNs cannot make full use of the location information in the input data. For some important character features of words, such

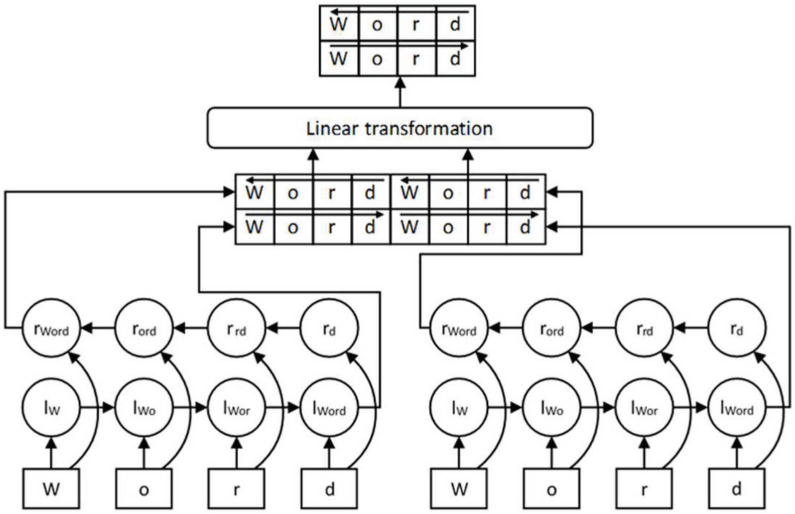

Figure 1 The character sequence of “Word” is input into a parallel structure with two groups of BiLSTM. Their outputs are concatenated and fed into a linear transformation layer to obtain the final character-level word representation.

as prefixes and suffixes, the location information of characters is particularly important. LSTM is very good at modeling sequence data, and the input close to the current time step has a greater impact on the output. Therefore, we refer to Lample et al.’s research [13], use BiLSTM to generate character-level word representations.

At the same time, we also use parallel BiLSTM to enhance the character-level word representations. Similar to previous researches [20, 21], the parallel BiLSTM structure is composed of several independent small scale BiLSTM, each BiLSTM receives the same character sequences of words and is trained simultaneously. We concatenate the output of each BiLSTM and feed them into a linear transformation layer. This layer is mainly to keep the dimension of the final character-level word representations fixed. Figure 1 describes how to generate character-level word representations with parallel BiLSTM.

Using this parallel BiLSTM structure to generate character-level word representations has the following advantages:

1. Enhance the expression abilities of character-level word representations. It can be seen from most of the existing researches that adding character-level word representations can significantly improve the final

performances of NER models, but unlike pretrained word embeddings, we cannot get ideal initial values for character embeddings through pre-training. We can only use random values for initialization, which will lead to the final results affected by the initial values. The parallel BiLSTM structure is composed of several independent BiLSTM, each BiLSTM has different initial parameters, so the input character sequences can be mapped into multiple different representation subspaces, so that a word can obtain multiple character-level word representations with different information. Then we use a linear transformation layer to combine these different outputs to get the final word representations. The character-level word representations generated in this way contains more information from different subspaces, so their expression abilities are better than those generated by a single BiLSTM or CNNs.

2. Compared with the same size BiLSTM, the parallel BiLSTM has better performance and lower calculation cost. For a basic RNN cell with *n* hidden units, we assume that the dimension of input *x* is *m*, The calculation equation of RNN is,

$$h_t = tanh(W[h_{t-1}, x_t] + b) \tag{1}$$

Where *W* and *b* are the weight matrix and bias, *tanh* is the hyperbolic tangent function, then the dimension of *W* is $[n, n + m]$, the total number of parameters need to be calculated is equal to $n * (n + m)$ (the bias is not considered here). If we adopt the parallel structure composed of *k* groups of RNN with n/k hidden units, we can reduce the number of parameters to $n * (n/k + m)$ while keeping the output dimension unchanged. In addition, Zhu et al.'s research [20] shows that the parallel RNN can reduce the negative effects from unrelated features in recurrent connections, thus improving the model performance.

3.3 Modeling Sentences by BiLSTM with Gating Mechanism

We get final word representations by concatenating character-level word representations and pretrained word embeddings, and feed them into BiLSTM to model sentences.

Firstly, we concatenate the bi-directional outputs of the first layer to obtain the *output_layer1*. Then, we feed the *output_layer1* into the second layer, with the same operation, we can get the *output_layer2*. Finally, in order to make better use of these outputs, we use gating mechanism to obtain the

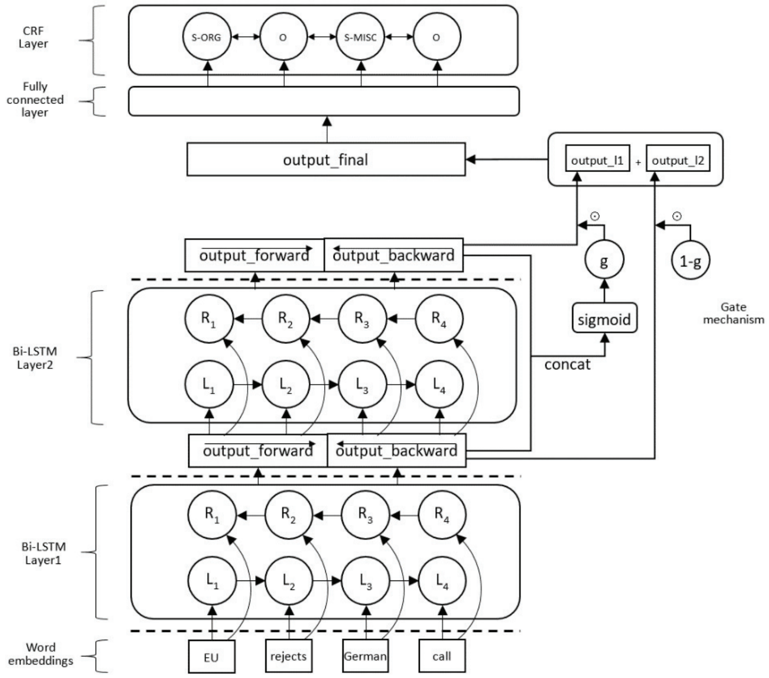

Figure 2 The structure of our two-layer BiLSTM with gating mechanism.

final outputs. The equations are as follows.

$$output\_concat = concat(output\_l1,\ output\_l2) \tag{2}$$

$$gate = sigmoid(W * output\_concat + b) \tag{3}$$

$$output\_final = gate \odot output\_l1 + (1 - gate) \odot output\_l2 \tag{4}$$

We concatenate the *output_layer1* and the *output_layer2* to get the *output_concat*, Equation (2). Then feed it into a fully connected layer with sigmoid activation function to calculate the vector *gate*, Equation (3). Finally, we use the *gate* and (*1-gate*) to calculate the *output_final*, Equation (4). Similarly, we also add a CRF layer for decoding, as shown in Figure 2.

Compared with the previous researches, our model has the following improvements:

1. Most of the previous researches use single-layer BiLSTM, or stacked BiLSTM similar to that used in Chiu and Nichols's research [12]. In their stacked BiLSTM, the forward and backward LSTM are independent

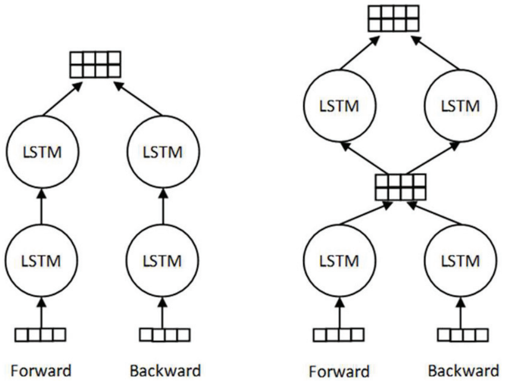

Figure 3 Difference between the stacked BiLSTM (left) and our two-layer BiLSTM used in this paper (right).

when transmitting information, and the bi-directional outputs of the top layer are concatenated as the final output. In our model, the forward and backward outputs of the first layer are concatenated as the input of the second layer for calculation. The difference between these two structures is shown in Fig. 3. From the experiment results, our two-layer BiLSTM structure used in this paper is better than the stacked BiLSTM.

2. In previous researches, when using BiLSTM to model sentences, most of them only use the output of single-layer or top-layer for sequence labelling. In this paper, we use all the output of each layer in the two-layer BiLSTM, and the added gating mechanism enables the model to learn how to select information from these outputs to generate the final text representations. According to the pervious researches [18, 22], the features extracted from texts in different layer of LSTM contain different types of information. The lower layer mainly contains the syntactic information in text, while the higher layer mainly contains semantic information of the word contexts. Therefore, our model will make use of all the output of each layer, so that we can combine different information to improve the result of named entity recognition.

4 Experiments

4.1 Parameter Setting and Training Algorithm

In this paper, we use TensorFlow [27] deep learning framework to build our models. Because we also use BiLSTM to generate character-level word representations, the parameter settings used in this paper refer to most of the settings in Lample et al.'s [13] research.

In order to obtain a fair comparison, we use the same data preprocessing method and English pretrained word embeddings published by Lample (since Lample did not publish the German pretrained word embeddings they used, so we use Gensim to train German corpus to obtain the pretrained word embeddings). The character-level embeddings are randomly initialized according to the uniform distribution of $[-0.5, 0.5]$, the dimension is 25. These initialized word-level and character-level embeddings will be continuously updated during the training process. The rest parameters are initialized by the method of Glorot and Bengio [28].

We use the stochastic gradient descent (SGD) as the parameter optimization algorithm, the initial learning rate is set to 0.1, and it decays at a rate of 0.97 per epoch. The gradient clipping parameter is set to 5.0, and we apply 50% dropout on the output of each layer (The locations are shown in the dotted line of Figure 2). We train our model for 60 epochs, during the training process, we monitor the model performance in the development dataset, and save the model with the best performance in the development dataset for the final test.

In the final model, the parallel structure used to generate character-level word representations consists of 6 groups of BiLSTM with the same size (see Section 5.2 for this parameter setting), and the hidden units of each forward and backward LSTM is 25. In the following linear transformation layer, the concatenated output dimension is reshaped to 50, that is, the dimension of the final character-level word representations is 50. The BiLSTM for modeling sentences has two layers, each LSTM has 100 hidden units.

4.2 Experiment Data Set

In order to verify the effectiveness of our model, we use the English and German datasets provided in the shared task of CoNLL-2003 for experiments. These datasets contain four different types of named entities: person (PER), organization (ORG), location (LOC) and miscellaneous (MISC), which are labelled in the IOB (Inside, Outside, Beginning) format. According to the previous researches [3, 13], the IOBES format has a stronger expression ability and can improve the final performance. Therefore, we use the IOBES format to re-label the datasets. For each model configuration, we conduct five experiments with different random seed and report the mean and standard deviation of the test dataset results.

5 Results and Discussion

5.1 Effectiveness of Each Part in our Model

Our model is improved in two aspects: generating character-level word representations and modeling sentences. In this section, we verify the impact of these two parts on our model performance. We select the BiLSTM-CRF model used in Lample et al.'s research [13] as the baseline model, on this basis, we gradually add the improved method proposed in this paper, report the F1 score in the test dataset to measure the model performance. The experiment results are shown in Tables 1 and 2, we use P-BiLSTM(k) to represent the parallel structure with k groups of BiLSTM and use BiLSTM(GM) to represent the BiLSTM with gating mechanism.

From the results in Table 1, when we use the same structure as the baseline model, the English test dataset F1 score is 90.95. On this basis, we first replace the original BiLSTM which generates character-level word representations in the baseline model with the parallel BiLSTM structure, and we can see that the performance of our model on the test dataset is significantly improved, the F1 score is 91.24. Then, we replace the single-layer BiLSTM used for modeling sentences in the baseline model with the two-layer BiLSTM with gating mechanism. Our model also outperforms the baseline model, and the F1 score reaches 91.23. Finally, we add all these two improvements mentioned above to the baseline model, it gets the best result in the English test dataset, the F1 score is 91.27. Table 2 shows the

Table 1 English NER results (CoNLL-2003 test dataset)

Model	Precision	Recall	F1 score
BiLSTM-CRF	91.03 (±0.27)	90.87 (±0.14)	90.95 (±0.11)
BiLSTM-CRF + P-BiLSTM(6)	91.11 (±0.21)	91.37 (±0.09)	91.24 (±0.12)
BiLSTM(GM)-CRF	91.08 (±0.19)	91.37 (±0.22)	91.23 (±0.20)
BiLSTM(GM)-CRF + P-BiLSTM(6)	91.18 (±0.15)	91.36 (±0.02)	**91.27(±0.08)**

Table 2 German NER results (CoNLL-2003 test dataset)

Model	Precision	Recall	F1 score
BiLSTM-CRF	79.50 (±0.25)	69.60 (±0.10)	74.22 (±0.17)
BiLSTM-CRF + P-BiLSTM(6)	79.65 (±0.24)	72.29 (±0.29)	75.79 (±0.18)
BiLSTM(GM)-CRF	79.49 (±0.34)	69.97 (±0.17)	74.42 (±0.16)
BiLSTM(GM)-CRF + P-BiLSTM(6)	80.26 (±0.36)	72.52 (±0.24)	**76.18(±0.39)**

Table 3 Comparison of character features between the CoNLL-2003 English and German datasets

	Average Number of Characters Per Word	Total Number of Characters
English	7.20	79
German	10.33	89

performances of different models in the German test dataset. Similarly, our models also outperform the baseline model, and the degree of performance improvement is higher than that of the English dataset.

As we can see from the results, these two methods proposed in this paper can improve the Recall scores of the model without reducing the Precision, so as to obtain better NER results. Further observation shows that our method is more effective for the German dataset, especially when the parallel BiLSTM is added to the baseline model to generate character-level word representations. The F1 score of English dataset is increased by 0.29 and that of German dataset is increased by 1.57. The main reason may be that the complexity of character features in German dataset is higher than that in English dataset, as shown in Table 3. Therefore, the traditional methods will face more challenges in generating German character-level word representations, while our method can better extract character features and improve the expression abilities of character-level word representations.

5.2 Best Group Number of the Parallel BiLSTM

In this paper, we use the parallel BiLSTM structure to generate character-level word representations, so the best group number of BiLSTM in the parallel structure also needs to be further discussed. Tables 4 and 5 show the impacts on the NER results when using the parallel BiLSTM structure with different group numbers to generate character-level word representations. It can be seen that when the group number increases, the performances of our models gradually improve. When the group number is 6, the models obtain the best results in the test dataset, the F1 score of the English dataset is 91.24 and the F1 score of the German dataset is 75.79. When the group number increase to 8, the performances of the models decrease. Therefore, the group number of parallel BiLSTM is not the more the better. Too many groups may have a negative impact on the model performance due to the information redundancy. We need to find an optimal group number to achieve the best performance.

Table 4 English NER results of the parallel BiLSTM structure with different group numbers

Model	Precision	Recall	F1 score
BiLSTM-CRF	91.03 (±0.27)	90.87 (±0.14)	90.95 (±0.11)
BiLSTM-CRF + P-BiLSTM(2)	91.05 (±0.13)	91.07 (±0.15)	91.06 (±0.13)
BiLSTM-CRF + P-BiLSTM(4)	91.05 (±0.29)	91.30 (±0.22)	91.18 (±0.17)
BiLSTM-CRF + P-BiLSTM(6)	91.11 (±0.21)	91.37 (±0.09)	**91.24(±0.12)**
BiLSTM-CRF + P-BiLSTM(8)	91.01 (±0.11)	91.37 (±0.18)	91.19 (±0.07)

Table 5 German NER results of the parallel BiLSTM structure with different group numbers

Model	Precision	Recall	F1 score
BiLSTM-CRF	79.50 (±0.25)	69.60 (±0.10)	74.22 (±0.17)
BiLSTM-CRF + P-BiLSTM(2)	80.32 (±0.25)	70.32 (±0.42)	74.99 (±0.13)
BiLSTM-CRF + P-BiLSTM(4)	79.31 (±0.22)	71.56 (±0.18)	75.24 (±0.15)
BiLSTM-CRF + P-BiLSTM(6)	79.65 (±0.24)	72.29 (±0.29)	**75.79(±0.18)**
BiLSTM-CRF + P-BiLSTM(8)	80.08 (±0.37)	71.48 (±0.45)	75.54 (±0.28)

Figure 4 shows the total number of parameters in the parallel BiLSTM structure with different group numbers (the size of hidden units is set to 200) and the training time of 1 epoch (CoNLL-2003 English dataset). It can be seen that when the group number increases, the total number of parameters in the parallel BiLSTM structure decreases, as discussed in Section 3.2. At the same time, the model takes less time to train in one epoch. Therefore, the parallel BiLSTM used in this paper can reduce the computational cost and improve the training speed.

5.3 Comparison with Previous Researches

In order to compare with the previous researches, Table 6 lists the F1 scores of the previous researches on the English test dataset of CoNLL-2003. Due to the small size of the dataset, we refer to the method used in researches [12, 17], combine the training dataset and development dataset for training. From the results in Table 6, it can be seen that the F1 score of our model is 91.66 in the test dataset. Compared with some other models which obtained high F1 scores, Chiu and Nichols [12] used some external knowledge sources (We define that the external knowledge source does not include the pretrained word embeddings), Peters et al. [17, 18] used extra large corpus to train the bi-directional language model, which is used to enhance the effect of pretrained

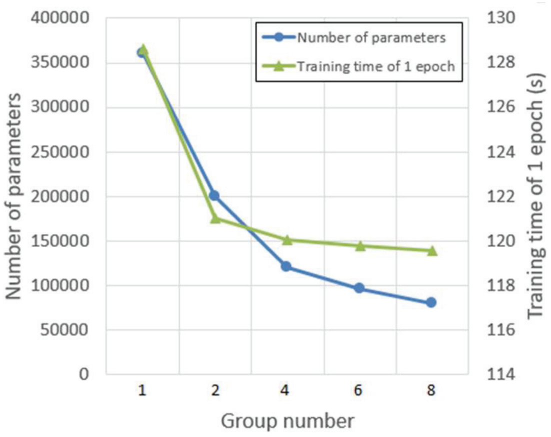

Figure 4 The number of parameters and the training time of 1 epoch with different group numbers.

Table 6 F1 scores of previous researches, where ‡ indicates that the research uses the neural network model, * indicates that the research uses the external knowledge source

Model	F1 score
Lample et al. [13] ‡	90.94
Luo et al. [6]	91.20
Ma and Hovy [15]‡	91.21
Sano et al. [29]	91.28
Žukov-Gregorič et al. [30]‡	91.48
Chiu and Nichols [12] ‡*	91.62
Peters et al. [17]‡*	91.93
Peters et al. [18]‡*	92.22
Our model	**91.66 (±0.15)**

word embeddings and improve the final performance. However, our method only changes the model structure and does not use any external resources. It is a complete end-to-end named entity recognition model.

6 Conclusion

This paper proposes a novel neural network model for named entity recognition. The model is improved from two aspects. One is to use the parallel

BiLSTM structure to generate character-level word representations. This method can combine the information from multiple representation subspaces to generate character-level word representations, so as to improve the expression abilities of word representations. The other is to use the BiLSTM with gating mechanism to model sentences. The outputs of different layers of LSTM contain different types of useful information, this gating mechanism enables the model to learn how to select information from these outputs to calculate, so as to improve the final performance. The experimental results on CoNLL-2003 English and German datasets show that our model is effective and its performance is significantly improved compared with the baseline model. Future research can start from the following aspects: how to improve the diversity of each BiLSTM in the parallel structure to further enhance the expression abilities of word representations, how to find the best group number more effectively and try to use the multi-layer outputs of the model in a more efficient way. In addition, using pretrained models, such as BERT, is currently a very prevalent trend, it is necessary to add them to the subsequent model research to further discuss the improvement of model performance.

Acknowledgement

The authors would like to thank Lample et al. for releasing their pretrained word embeddings.

Funding Statement

The authors received no specific funding for this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by gibbs sampling,” in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, Michigan, USA, 2005, pp. 363–370.

[2] J. Kazama and K. Torisawa, “Exploiting Wikipedia as external knowledge for named entity recognition,” in Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 2007, pp. 698–707.

[3] L. Ratinov and D. Roth, “Design challenges and misconceptions in named entity recognition,” in Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), Boulder, Colorado, USA, 2009, pp. 147–155.

[4] A. Passos, V. Kumar, and A. McCallum, “Lexicon infused phrase embeddings for named entity resolution,” in Proceedings of the Eighteenth Conference on Computational Natural Language Learning, Baltimore, Maryland, USA, 2014, pp. 78–86.

[5] W. Radford, X. Carreras, and J. Henderson, “Named entity recognition with document-specific KB tag gazetteers,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 2015 pp. 512–517.

[6] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie, “Joint entity recognition and disambiguation,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 2015, pp. 879–888.

[7] D. Benikova, S. Muhie, Y. Prabhakaran, et al. “GermaNER: free open german named entity recognition tool,” Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology, University of Duisburg-Essen, Germany, 2015, pp. 31–38.

[8] T. Mikolov, G. Corrado, K. Chen, et al. “Efficient estimation of word representations in vector space,” in Proceedings of the International Conference on Learning Representations (ICLR 2013), Scottsdale, Arizona, USA, 2013.

[9] P. Bojanowski, E. Grave, A. Joulin, et al., “Enriching word vectors with subword information,” Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, Jun. 2017.

[10] C. dos Santos and V. Guimarães, “Boosting named entity recognition with neural character embeddings,” in Proceedings of the Fifth Named Entity Workshop, Beijing, China, 2015, pp. 25–33.

[11] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” arXiv preprint arXiv:1508.01991, 2015.

[12] J. P. Chiu and E. Nichols, “Named entity recognition with bidirectional LSTM-CNNs,” Transactions of the Association for Computational Linguistics, vol. 4, pp. 357–370, 2016.

[13] G. Lample, M. Ballesteros, S. Subramanian, et al., “Neural architectures for named entity recognition,” in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 2016, pp. 260–70.

[14] Z. Yang, R. Salakhutdinov, and W. Cohen, “Multi-task cross-lingual sequence tagging from scratch,” arXiv preprint arXiv:1603.06270, 2016.

[15] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF,” in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 2016, pp. 1064–1074.

[16] E. F. T. K. Sang and F. De Meulder, “Introduction to the CoNLL-2003 shared task: language-independent named entity recognition,” in Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL, Edmonton, Canada, 2003, pp. 142–147.

[17] M. E. Peters, W. Ammar, C. Bhagavatula, et al., “Semi-supervised sequence tagging with bidirectional language models,” in Meeting of the Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1756–1765.

[18] M. Peters, M. Neumann, M. Iyyer, et al., “Deep contextualized word representations,” in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, USA, 2018, pp. 2227–2237.

[19] R. Collobert, J. Weston, L. Bottou, et al., “Natural language processing (almost) from scratch,” Journal of Machine Learning Research, vol. 12, pp. 2493–2537, 2011.

[20] D. Zhu, S. Shen, X.-Y. Dai, et al., “Going wider: recurrent neural network with parallel cells,” arXiv preprint arXiv:1705.01346, 2017.

[21] F. Gao, L. Wu, T. Qin, et al., “Efficient sequence learning with group recurrent networks,” in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, USA, 2018, pp. 799–808.

[22] Y. Belinkov, N. Durrani, F. Dalvi, et al., “What do neural machine translation models learn about morphology?,” in Proceedings of the

55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 861–872.

[23] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM networks,” in Proceedings. 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 2005, pp. 2047–2052.

[25] J. D. Lafferty, A. McCallum, and F. C. Pereira, “Conditional random fields: probabilistic models for segmenting and labeling sequence data,” in Proceedings of the Eighteenth International Conference on Machine Learning, Williamstown, MA, USA, 2001 pp. 282–289.

[26] C. D. Santos and B. Zadrozny, “Learning character-level representations for part-of-speech tagging,” in Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, 2014, pp. 1818–1826.

[27] M. Abadi, P. Barham, J. Chen, et al., “Tensorflow: large-scale machine learning on heterogeneous distributed systems,” in Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2016.

[28] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 2010, pp. 249–256.

[29] M. Sano, H. Shindo, I. Yamada, et al., “Segment-level neural conditional random fields for named entity recognition,” in Proceedings of the Eighth International Joint Conference on Natural Language Processing, Taipei, Taiwan, 2017, pp. 97–102.

[30] A. Žukov-Gregorič, Y. Bachrach, and S. Coope, “Named entity recognition with parallel recurrent neural networks,” in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 69–74.

Biographies

Yenan Yi received his B.S. degree from Nanjing University of Science and Technology, Nanjing, China, in 2012; M.S. degree from Hohai University, Nanjing, China, in 2017. He is currently a Ph.D. student at Hohai University, and his research interests are information management and intelligent question answering.

Yijie Bian received his B.S., M.S. and Ph.D. degrees from Hohai University, Nanjing, China. He is a professor and doctoral supervisor of Hohai University. His research interests include information management and e-commerce, financial engineering and investment management.