Chinese Named Entity Recognition with Character-Level BLSTM and Soft Attention Model

Jize Yin, Senlin Luo, Zhouting Wu and Limin Pan

(Information System and Security & Countermeasures Experimental Center, Beijing Institute of Technology, Beijing 100081, China)

Abstract: Unlike named entity recognition (NER) for English, the absence of word boundaries reduces the final accuracy for Chinese NER. To avoid accumulated error introduced by word segmentation, a deep model extracting character-level features is carefully built and becomes a basis for a new Chinese NER method, which is proposed in this paper. This method converts the raw text to a character vector sequence, extracts global text features with a bidirectional long short-term memory and extracts local text features with a soft attention model. A linear chain conditional random field is also used to label all the characters with the help of the global and local text features. Experiments based on the Microsoft Research Asia (MSRA) dataset are designed and implemented. Results show that the proposed method has good performance compared to other methods, which proves that the global and local text features extracted have a positive influence on Chinese NER. For more variety in the test domains, a resume dataset from Sina Finance is also used to prove the effectiveness of the proposed method.

Key words: Chinese; named entity recognition (NER); character-level; bidirectional long short-term memory; soft attention model

CLC number: TP 391.1 **Document code**: A **Article ID**: 1004-0579(2020)01-0060-12

Named entity recognition (NER) is a subtask of information extraction (IE), one of the main tasks in natural language processing (NLP). NER was established on the sixth message understanding conference (MUC-6)[1] and aims to recognize named entities like person names, location names and organization names in text. Many downstream applications like relation extraction[2] and machine translation[3] are based on NER. Recently, Chinese NER has received attention in light of the elevation of China's international status[4-9].

Unlike English NER, Chinese NER must deal with unstructured text without any word boundaries. Therefore, the processing procedure for Chinese NER usually begins with a word segmentation step[10-13]. Recent research efforts also verify the importance of word segmentation information[14-16]. However, Kuru et al. describe a character-level tagger and evaluate it on different NER tasks. The results across seven languages including Arabic, Czech, Dutch, English, German, Spanish and Turkish show that viewing characters as the representation is better than considering words as the representation[17], which provides a new thought that Chinese NER can be improved by adequately extracting only character-level features. There have also been several works drawing attention to character-level feature extraction for Chinese NER[5-7].

For character- or word-level feature extraction, deep models are helpful[18] and there have recently been several works on deep neural net-

Received 2018-10-15
Supported by 242 National Information Security Projects (2017A149)
✉ Author for correspondence, senior engineer
E-mail: panlimin2016@gmail.com
DOI: 10.15918/j.jbit1004-0579.18161

works applied in NER recently[15-17,19]. As for the methods of model training, although semi-supervised and unsupervised learning algorithms are widely used in NER[4,16,20-21], supervised learning algorithms are still popular in NER because of their simplicity and reliability[9,15,17,19].

In this article, a deep model that combines a bidirectional long short-term memory (BLSTM) with a soft attention model (AM) is proposed to improve character-level feature extraction performance. A Chinese NER method is given based on this model. The experimental results based on the Microsoft Research Asia (MSRA) dataset prove the validity of the character-level feature extraction and show the proposed method's good performance. Further experimental results based on a resume dataset from Sina Finance prove the effectiveness of the proposed method for various test domains.

1 Related Work

1.1 Deep neural networks for NER

Some effective NER methods are shown in Tab. 1. There have been many hand-crafted feature-based models used for NER such as hidden Markov models (HMM)[22], decision trees[23], maximum entropy models (ME)[24], support vector machines (SVM)[25-28] and conditional random fields (CRF)[29]. Recently, there have been several research efforts that have generated improved models[30-31]. To avoid hand-crafted features[11,32-33] and achieve automatic text feature extraction for NER, many new deep models based on neural network models like long short-term memory (LSTM)[15], BLSTM[16-17] and convolutional neural network (CNN)[19] have been proposed and proven to be effective.

Tab. 1 Several existing NER methods

Categorie	Applied model and method	Representative
Using hand-crafted feature-based models	Hidden markov models (HMM)	Bikel et al. (1997)
	Decision trees	Sekine(1998)
	Maximum entropy models (ME)	Borthwick et al. (1998)
	Support vector machines (SVM)	Isozaki and Kazawa(2002); Kazama et al. (2002); Takeuchi and Collier (2002); Asahara and Matsumoto (2003); Patra and Saha (2013)
	Conditional random fields (CRF)	McCallum and Li (2003); Hao et al. (2013)
Using deep models based on neural network models	Long short-term memory (LSTM)	Dong et al. (2016); Peng and Dredze (2016)
	Bidirectional long short-term memory (BLSTM)	Kuru et al. (2016); He and Sun (2017)
	Convolutional neural network (CNN)	Chiu and Nichols (2016); Zheng et al. (2016); Zheng et al. (2017)
	Skip-gram model	Lu et al. (2016)
Using extra knowledge	Word boundary information	Peng and Dredze(2016); Zhang and Yang (2018)

Several feature extraction methods based on deep models have led to good results in NER. Chiu et al. provide a neural network architecture that automatically detects word- and character-level features. This architecture includes a BLSTM to extract word-level features and CNNs to extract character-level features[19]. However, Chiu et al. 's method is for English with clear word boundaries. For some languages, character-level features might be better than word-level fea-

tures. To compare character-level features with word-level features, Kuru et al. use a deep model consisting of stacked bidirectional LSTMs to get characters and output tag probabilities. These probabilities are then converted to named entity tags by a Viterbi decoder. The experimental results based on seven languages prove that character-level features are better than word-level features[17]. Similarly, Lu et al. treat Chinese sentences as sequences of characters and present a position-sensitive skip-gram model to learn multi-prototype Chinese character embeddings. The multi-prototype Chinese character embeddings are then used as features in the Chinese NER task, which results in an improvement in recognition[6]. It is a way to extract features by using unsupervised learning, and there are other methods with unsupervised character representations. Dong et al. make their model utilize both character-level and radical-level representations[5]. Zheng et al. transfer the context-specific and multi-prototype character representations into an NER task, which proves the ability of unsupervised character representation to enhance supervised learning[7]. Furthermore, Zheng et al. continue to work on unsupervised character representations and observe an improvement in the supervised learning tasks like NER[8]. However, the methods using unsupervised character representations need an additional process of unsupervised learning, all except basic supervised learning. Moreover, the above methods treat all the characters equally, which ignores the varying importance of characters in text.

Extra knowledge has been used in the NER task. For Chinese, the absence of word boundary information is treated as a problem, so several research efforts on Chinese NER use word boundary information as extra knowledge. Peng et al. use word boundary tags as features to provide richer information for an NER system, which is realized by jointly training an NER model and a Chinese word segmentation (CWS) model based on a LSTM-CRF model[15]. Furthermore, Zhang and Yang propose a good model named lattice LSTM[9]. This model receives a character-level one-hot vector sequence as the input, seeks candidate words and processes them in the form of word embeddings. That means Zhang and Yang still have to rely on the word boundary information like Peng et al.[15]. In general, the NER corpus does not have any other information except named entity information. So, the biggest disadvantage is that this kind of method needs more external labor.

In conclusion, there are still some problems that need to be solved in NER, especially for Chinese. In this article, a Chinese NER method based on a deep model combining a BLSTM with a soft attention model is proposed and character-level features are extracted to identify named entities. Without adding word boundary information, other features named local text features are extracted to make an improvement in Chinese NER. Supervised learning is enough and there is no need for other learning algorithms such as semi-supervised learning[4,16] or unsupervised learning[7-8]. Experimental results present the proposed method's satisfactory performance.

1.2 Attention mechanism

The attention mechanism in neural networks, a selection mechanism that can allocate limited information processing capacity, is based on the visual attention mechanism exhibited in humans. Using an attention mechanism means that some important information will be addressed while other information will be ignored. There have been several applications of attention mechanism in different fields[34-37]. For some special applications in NLP, the important parts of text should be given high weights and the useless parts should be given low weights.

Bahdanau et al. propose an attention model to search for parts of a source sentence automatically so that the sentence-length feature can be useful for neural machine translation[38]. This at-

tention model is generally called a soft attention model, compared to its opposite model called a hard attention model[39]. Shang et al. use the soft attention model to build a local scheme that is combined with a global scheme to be one part of the general encoder-decoder framework taken by the neural responding machine (NRM), a neural network-based response generator for short-text conversation[40]. A soft attention model gives weights to every word or character in the source sentence, but a hard attention model needs only to find the critical word or character in the source sentence and ignore the rest. Luong et al. build a global attention model based on the soft attention model and a local attention model based on the tradeoff between the soft and hard attention models for neural machine translation[41]. Moreover, there are many improved models and new mechanisms based on the attention mechanism for special applications[42-56].

In this article, a deep model combining a soft attention model with a BLSTM like what Bahdanau et al. provide[38] is proposed so that the important parts of the text can be weighted more heavily than the rest. There is a small difference between the proposed model and Bahdanau et al.'s model. The proposed model uses a soft attention model considering the opposite directions of the two LSTMs that comprise a BLSTM, but Bahdanau et al.'s model doesn't. The global and local text features extracted by the proposed model are different from what the global and local attention models extract[41]. The global and local attention models are distinguished by the attention window sizes, but the global and local text features are distinguished by whether or not the soft attention model is used. By using the proposed model, a good performance of the Chinese NER method is obtained.

2 Method

2.1 Framework

As shown in Fig. 1, there are three parts in the proposed Chinese NER method: character vector conversion, feature extraction and named entity tagging. In particular, the feature extraction consists of both global text feature extraction via a BLSTM and local text feature extraction via a soft attention model.

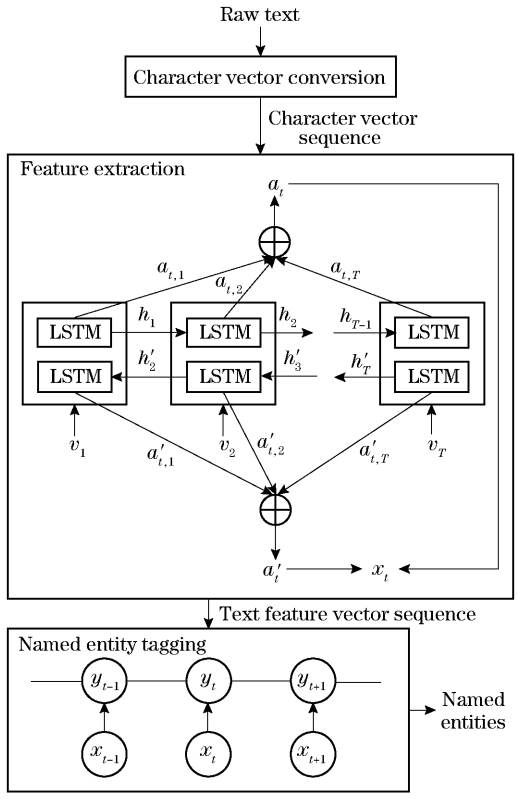

Fig. 1 Proposed method of Chinese NER

2.2 Character vector conversion

For a character-based language like Chinese, each character should be converted to a unique vector so that the raw text can be used for mathematical calculation. To achieve this goal, a one-hot model is used. First, a character dictionary is built by giving sequence numbers to different characters in it. For instance, if A is the first character in the dictionary, No. 1 will be given to A. No. 2 will be given to the second character, and so on. Next, a zero vector whose dimension is the number of different characters is built. Finally, a one-hot vector occurs by replacing zero at the position to which a character's number corresponds with one. After that, the raw text is converted to a character vector sequence.

2.3 Feature extraction

Feature extraction can extract important features from text and store them in the vectors of real numbers which are called feature vectors. In this article, NER is viewed as a sequence labeling problem, so there is a feature vector generated as output for each given character vector.

In the proposed Chinese NER method, the feature extraction consists of global text feature extraction and local text feature extraction.

Global text features can be abstracted into a text sequential feature from a character sequence. In the grammatical sense, the text sequential feature decides the characters' order in text because people write or read a text message in some order. In the semantic sense, there is practical meaning when the characters are arranged in some order. The global text features including sufficient grammatical and semantic information are extracted as a bidirectional text sequential feature by a BLSTM.

Local text features can be regarded as remaining text features that get rid of the unnecessary information and they are determined by the chosen parts of text. In the grammatical sense, the unnecessary information might destroy the common grammatical structures. In the semantic sense, the unnecessary information can be easily replaced or deleted because it has a marginal effect on the core sense of the whole text. The local text features, including sufficient grammatical and semantic information, are extracted by a soft attention model.

2.3.1 Long short-term memory

A long short-term memory is a specially designed neural network model proposed by Sepp Hochreiter and Jürgen Schmidhuber[57]. It obtains the output of the last moment and the input of the present moment like what a recurrent neural network (RNN) does, so it has the ability to process sequential data. But compared with an RNN, a LSTM can remember more information and is usually good at processing long text by adjusting the entire structure with a cell that connects to other functional structures of the LSTM. These specially designed functional structures are called gates, which include an input gate, a forget gate and an output gate.

In the process of Chinese NER, the inputted text contains many character sequences, so it's easy to convert text processing to character sequence processing. A LSTM can extract the unidirectional sequential feature from a character sequence after the character sequence is converted to a vector sequence of real numbers.

First, a LSTM should decide which part of the new information needs to be stored in the cell. The input gate of the LSTM can make it and is shown in

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{\mathrm{vi}}\boldsymbol{v}_t + \boldsymbol{W}_{\mathrm{hi}}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{\mathrm{ci}}\boldsymbol{c}_{t-1} + \boldsymbol{b}_{\mathrm{i}}) \quad (1)$$

where $\boldsymbol{i}_t$ is the present state of the input gate and σ is the logistic sigmoid function. The vector $\boldsymbol{v}_t$ is the present input whose weight is $\boldsymbol{W}_{\mathrm{vi}}$. The vector $\boldsymbol{h}_{t-1}$ is the output of the last moment (hidden state vector) whose weight is $\boldsymbol{W}_{\mathrm{hi}}$. The vector $\boldsymbol{c}_{t-1}$ is the cell state of the last moment whose weight is $\boldsymbol{W}_{\mathrm{ci}}$. The vector $\boldsymbol{b}_{\mathrm{i}}$ is the bias.

Meanwhile some old information in the cell needs to be forgotten. The forget gate of the LSTM can perform this function and is shown in

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{\mathrm{vf}}v_t + \boldsymbol{W}_{\mathrm{hf}}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{\mathrm{cf}}\boldsymbol{c}_{t-1} + \boldsymbol{b}_{\mathrm{f}}) \quad (2)$$

where $\boldsymbol{f}_t$ is the present state of the forget gate. The present input $\boldsymbol{v}_t$ has a weight $\boldsymbol{W}_{\mathrm{vf}}$, the output of the last moment $\boldsymbol{h}_{t-1}$ has a weight $\boldsymbol{W}_{\mathrm{hf}}$, the cell state of the last moment $\boldsymbol{c}_{t-1}$ has a weight $\boldsymbol{W}_{\mathrm{cf}}$, and the vector $\boldsymbol{b}_{\mathrm{f}}$ is the bias.

Next, the cell needs to be updated and its present state $\boldsymbol{c}_t$ is shown in

$$\boldsymbol{c}_t = \boldsymbol{f}_t\boldsymbol{c}_{t-1} + \boldsymbol{i}_t\tanh(\boldsymbol{W}_{\mathrm{vc}}\boldsymbol{v}_t + \boldsymbol{W}_{\mathrm{hc}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\mathrm{c}}) \quad (3)$$

where tanh is the hyperbolic tangent function. The present input $\boldsymbol{v}_t$ has a weight $\boldsymbol{W}_{\mathrm{vc}}$, the output of the last moment $\boldsymbol{h}_{t-1}$ has a weight $\boldsymbol{W}_{\mathrm{hc}}$, and the vector $\boldsymbol{b}_{\mathrm{c}}$ is the bias.

The LSTM needs to decide which part of the information should constitute the output. The output gate of the LSTM can make it and is shown

in

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{\mathrm{vo}}\boldsymbol{v}_t + \boldsymbol{W}_{\mathrm{ho}}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{\mathrm{co}}\boldsymbol{c}_t + \boldsymbol{b}_{\mathrm{o}}) \quad (4)$$

where $\boldsymbol{o}_t$ is the present state of the output gate. The present input $\boldsymbol{v}_t$ has a weight $\boldsymbol{W}_{\mathrm{vo}}$, the output of the last moment $\boldsymbol{h}_{t-1}$ has a weight $\boldsymbol{W}_{\mathrm{ho}}$, the present cell state $\boldsymbol{c}_t$ has a weight $\boldsymbol{W}_{\mathrm{co}}$, and the vector $\boldsymbol{b}_{\mathrm{o}}$ is the bias.

Finally, the present hidden state vector $\boldsymbol{h}_t$ output by the LSTM is shown in

$$\boldsymbol{h}_t = \boldsymbol{o}_t \tanh \boldsymbol{c}_t \quad (5)$$

The hidden state vector sequence output by the LSTM is the unidirectional sequential feature vector sequence.

2.3.2 Bidirectional long short-term memory

A bidirectional long short-term memory is a neural network model proposed by Alex Graves and Jürgen Schmidhuber[58]. This model consists of two LSTMs named forward LSTM and backward LSTM. The forward LSTM processes the sequential data in the forward direction and the backward LSTM processes the sequential data in the opposite direction. The BLSTM's structure is shown in Fig. 2. For each moment, the hidden state vector from the forward LSTM and the hidden state vector from the backward LSTM are combined to form the final output.

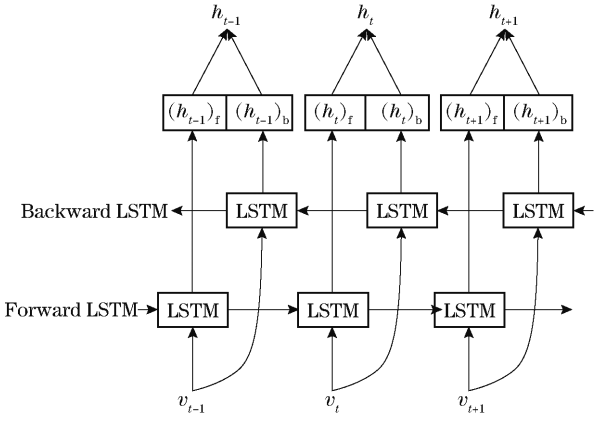

Fig. 2 BLSTM's structure

A BLSTM can output bidirectional text sequential feature vector sequences $\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{t-1}, \boldsymbol{h}_t, \boldsymbol{h}_{t+1}, \boldsymbol{h}_{t+2}, \cdots, \boldsymbol{h}_T\}$ (T being the length of the output) after getting the computed text input $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{t-1}, \boldsymbol{v}_t, \boldsymbol{v}_{t+1}, \boldsymbol{v}_{t+2}, \cdots, \boldsymbol{v}_T\}$ (T being the length of the input).

2.3.3 Soft attention model

For Chinese text processing, the attention mechanism works by assigning different mathematical weights to different characters. In this article, the soft attention model proposed by Bahdanau et al.[38] is used to extract local text features, whose structure is shown in Fig. 1.

The soft attention model based on the BLSTM that obtains the character vector sequence $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_T\}$, where T is the length of the character vector sequence, consists of two parts: one for the forward LSTM and the other for the backward LSTM. The two parts have the same structure and function. Considering the part for the forward LSTM, all the hidden state vectors from the forward LSTM $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_T\}$, where T is the length of the output from the forward LSTM, are given corresponding weights $\{\alpha_{t,1}, \alpha_{t,2}, \cdots, \alpha_{t,T}\}$, where t represents the present moment and T is the length of the output from the forward LSTM. The sum of the weights is the attention state $\boldsymbol{a}_t$ of the present moment.

$$\boldsymbol{a}_t = \sum_{j=1}^{T} \alpha_{tj}\boldsymbol{h}_j \quad (6)$$

The weight α_{tj} can be calculated by

$$\alpha_{tj} = \frac{\exp e_{tj}}{\sum_{k=1}^{T} \exp e_{tk}} \quad (7)$$

And the variable e_{tj} can be calculated by

$$e_{tj} = \boldsymbol{V}^{\mathrm{T}} \tanh(\boldsymbol{W}_1\boldsymbol{a}_{t-1} + \boldsymbol{W}_2\boldsymbol{h}_j + \boldsymbol{b}_t) \quad (8)$$

where tanh is the hyperbolic tangent function. e_{tj} represents the associated energy that is used to measure the relationship between the input in the position j and the output in the position t. The associated energy is based on the attention state of the last moment $\boldsymbol{a}_{t-1}$ with the weight $\boldsymbol{W}_1$ and the hidden state vector $\boldsymbol{h}_j$ with the weight $\boldsymbol{W}_2$. The vector $\boldsymbol{b}_t$ is the bias. The vector $\boldsymbol{V}$ is the weight for the combination of all the features.

Finally, an attention state sequence $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_T\}$, where T is the length of the output from the forward LSTM, is output. Each attention state has local text features based on the sequential

feature. In the same way, there is an attention state sequence for the backward LSTM. By combining both of the attention state sequences, there is a global and local text feature vector sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$, where T is the length of the input character vector sequence, as the output of feature extraction.

2.4 Named entity tagging

Named entity tagging is the process of giving a tag to each text feature vector. In this article, a linear chain CRF is used, which is shown in Fig. 1.

CRFs, a type of discriminative undirected probabilistic graphical model, are proposed by Lafferty et al.[59]. If $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ is the observed variable sequence and $\boldsymbol{y} = \{y_1, y_2, \cdots, y_n\}$ is the corresponding tag variable sequence, a conditional probability model $P(\boldsymbol{y} \mid \boldsymbol{x})$ can be obtained. If the model $P(\boldsymbol{y} \mid \boldsymbol{x})$ meets the Markov property, as is shown

$$P(y_i \mid \boldsymbol{x}, y_1, y_2, \cdots, y_n) = P(y_i \mid \boldsymbol{x}, y_{i-1}, y_{i+1}) \tag{9}$$

then the model $P(\boldsymbol{y} \mid \boldsymbol{x})$ is a linear chain CRF.

The conditional probability function $P(\boldsymbol{y} \mid \boldsymbol{x})$ is shown in

$$P(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_j \sum_{i=2}^{n} \lambda_j t_j(y_{i-1}, y_i, \boldsymbol{x}, i) + \sum_k \sum_{i=1}^{n} \mu_k s_k(y_i, \boldsymbol{x}, i) \right) \tag{10}$$

using exponential potential functions and feature functions.

The function $t_j(y_{i-1}, y_i, \boldsymbol{x}, i)$ is a transition feature function whose weight is λ_j, which is used to represent the relationship between the adjacent tag variables and the influence from the observed variable sequence. The function $s_k(y_i, \boldsymbol{x}, i)$ is a status feature function whose weight is μ_k, which is used to represent the influence from the observed variable sequence. The function $Z(\boldsymbol{x})$ shown in

$$Z(\boldsymbol{x}) = \sum_y \exp\left(\sum_j \sum_{i=2}^{n} \lambda_j t_j(y_{i-1}, y_i, \boldsymbol{x}, i) + \sum_k \sum_{i=1}^{n} \mu_k s_k(y_i, \boldsymbol{x}, i) \right) \tag{11}$$

is a normalizing factor used to make sure that Eq. (10) is defined correctly.

Sequential log-likelihood loss is used to train the model. After finishing model training, the linear chain CRF outputs a tag sequence based on the text feature vector sequence by using the Viterbi algorithm. Final named entities are determined according to the tag sequence.

3 Experiments and Analysis

3.1 Purpose

To prove the validity of the feature extraction model which combines a BLSTM with a soft attention model, some experiments based on the MSRA dataset are designed and implemented. The contrast models include a LSTM-CRF model, a BLSTM-CRF model, a LSTM-AM-CRF model and a BLSTM-AM-CRF model. Other research results are also referenced to prove that the proposed Chinese NER method in this article performs well. Further experiments based on a resume dataset from Sina Finance attempt to prove the effectiveness of the proposed method for more variety in test domains.

3.2 Experimental settings

The experimental datasets shown in Tab. 2 include the MSRA dataset[60] and a Chinese resume dataset annotated by Zhang and Yang[9]. While the MSRA dataset is in the news domain, the resume dataset from Sina Finance consists of resumes of senior executives from listed companies in the Chinese stock market[9].

Tab. 2 Statistics of datasets

Dataset	Type	Train	Dev	Test
MSRA	Sentence	46.4k	–	4.4k
	Char	2 169.9k	–	172.6k
Resume	Sentence	3.8k	0.46k	0.48k
	Char	124.1k	13.9k	15.1k

Hyper-parameters for the proposed method are shown in Tab. 3. The optimization method in this article is adaptive moment estimation with a learning rate of 0.01.

Tab. 3 Hyper-parameter values

Parameter	Value
BLSTM layer	1
Learning rate	0.01
LSTM hidden	50
Batch size	1
Epoch	50

The experimental software environment is shown in Tab. 4.

The experimental hardware environment is shown in Tab. 5.

Tab. 4 Experimental software environment

Software	Source	Instruction
Python 3.6.6	Python software foundation	A high-level programming language
TensorFlow 1.10.0	Google brain team	An open-source software library for dataflow programming across a range of tasks①
Numpy	Community project	A library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays②

① https://en.wikipedia.org/wiki/TensorFlow. ② https://en.wikipedia.org/wiki/NumPy.

Tab. 5 Experimental hardware environment

Hardware	Number	Configuration
Server	1	CPU: Intel (R) Xeon (R) CPU E5-2680 v4 @ 2.40 GHz RAM: 64 GB Operating System: Ubuntu 16.04.5 LTS Graphic Card: GTX 1080 Ti

3.3 Evaluation method

Precision (P), recall (R), and F-measure (F_1) are used to evaluate the experimental results. P is the ratio of the number of named entities correctly identified (true positive, TP) to the number of named entities found in text (true positive & false positive, $TP+FP$), which is represented as the equation $P=TP/(TP+FP)$. R is the ratio of the number of named entities correctly identified (true positive, TP) to the number of named entities actually existing in text (true positive & false negative, $TP+FN$), which is represented as the equation $R=TP/(TP+FN)$. F_1 is the harmonic mean based on P and R, which is represented as the equation $F_1=2PR/(P+R)$.

3.4 Experimental procedure

In this article, four models are trained and tested based on the MSRA dataset and the Chinese resume dataset. The four models use a LSTM, a BLSTM, a model combining a LSTM with a soft attention model, and a model combining a BLSTM with a soft attention model to extract text features respectively. All of them use a linear chain CRF to tag named entities. There have been several research publications on Chinese NER based on the MSRA dataset[4-9,11,32-33].

3.5 Results and discussion

The main results on the MSRA dataset are shown in Tab. 6. Compared to previous methods described in Section 1.1, the proposed method in this article has shown competitive results. Compared to method 1, method 2 obtains a good performance with an 86.84% F1-score. It is obvious that a BLSTM performs better than a LSTM in text feature extraction, which confirms the theory of bidirectional text sequential feature in this article. As for method 3, adding a soft attention model to the LSTM-CRF model increases the F1-score of Chinese NER to 85.14%. The soft attention model also works with a BLSTM-CRF model, as shown by comparing method 2 with method 4. Therefore, extracting local text features is effective. Moreover, the final proposed method receives an 88.27% F1-score, better than other three methods.

Main results on resume NER are shown in Tab. 7. The proposed method performs better than the state-of-the-art method[9] with a 95.24% F1-score. It can be seen that the proposed method works well in the domain of Sina Finance. There is only one pattern different from that shown in Tab. 6. Based on the resume NER dataset, method 4 seems worse than method 2. Method 2

Tab. 6 Main results on MSRA

Method	Information	P/%	R/%	F_1/%
Chen et al. (2006)	Hand-crafted features + CRF + ME	91.22	81.71	86.20
Zhang et al. (2006)	Hand-crafted features + ME	92.20	90.18	91.18
Zhou et al. (2013)	Hand-crafted features + boundary identification + entity categorization	91.86	88.75	90.28
Han et al. (2015)	CRF + graph-based semi-supervised learning	77.84	90.62	83.74
Lu et al. (2016)	Position-sensitive skip-gram model + multi-prototype Chinese character embeddings	–	–	87.94
Dong et al. (2016)	Character-level and radical-level representations	91.28	90.62	90.95
Zheng et al. (2016)	Context-specific and multi-prototype character	86.70	83.49	85.06
Zheng et al. (2017)	Representations	–	–	85.59
Zhang and Yang (2018)	Lattice LSTM + CRF	93.57	92.79	93.18
Method 1	LSTM + linear chain CRF	86.95	82.84	84.85
Method 2	BLSTM + linear chain CRF	88.05	85.67	86.84
Method 3	LSTM + AM + linear chain CRF	87.79	82.65	85.14
Method 4	BLSTM + AM + linear chain CRF	89.83	86.76	88.27

Tab. 7 Main results on resume NER

Method	P/%	R/%	F_1/%
Zhang and Yang (2018)	94.81	94.11	94.46
Method 1	94.80	94.03	94.41
Method 2	95.89	95.34	95.61
Method 3	94.37	95.03	94.70
Method 4	95.86	94.62	95.24

and method 4 both use a BLSTM and a linear chain CRF with the same hyper-parameters. However, an attention model is used in method 4, but not used in method 2. Therefore, the whole deep model used in method 4 is more complex than that used in method 2. Because the resume NER dataset is small, the deep model used in method 2 with given hyper-parameters shown in the article can perfectly fit the training corpus, which leads to a 100% F1-score during model training as we have already seen. Generally, under the circumstance of perfectly fitting the training dataset, the more complex a model is, the worse its generalization ability is. That's why method 4 has worse performance on the test dataset compared with method 2.

In other words, if the development dataset of the resume NER corpus is used to adjust the common hyper-parameters in method 2 and method 4, both the deep models respectively used in method 2 and method 4 could be simpler, but the one used in method 4 is still more complex. So, if the deep model used in method 2 cannot fit the training dataset perfectly, but the one used in method 4 can, then method 4 is likely to perform better on the test dataset compared with method 2.

4 Conclusions

In this article, a deep model extracting character-level features is carefully built to avoid the accumulated error introduced by word segmentation. This problem is the basis for the new proposed Chinese NER method. This method converts the raw text to a character vector sequence with the help of a one-hot model, extracts global text features that are abstracted into a bidirectional text sequential feature with a BLSTM and extracts local text features without the unnecessary information using a soft attention model. Because NER can be viewed as a sequence labeling problem, a linear chain CRF is used to label all the characters with the help of the global and local text features. Results on the MSRA dataset prove the fact that the global and local text features have a positive influence on Chinese NER. Results on a resume NER dataset from Sina Finance prove the robustness of the proposed method in various test domains.

The global text features are abstracted into a text sequential feature from a character sequence in this article, but people might read and organize the words or characters not one by one but with a special structure like a tree structure. Therefore, other text structures need to be considered to replace the sequential structure in the future.

References:

[1] Grishman R, Sundheim B M. Message understanding conference-6: A brief history[C]//Proceedings of the 16th International Conference on Computational Linguistics-Volume 1, San Mateo, CA, USA, 1996.

[2] Bunescu R C, Mooney R J. A shortest path dependency kernel for relation extraction[C]//Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 2005.

[3] Babych B, Hartley A. Improving machine translation quality with automatic named entity recognition [C]//Proceedings of the 7th International EAMT Workshop on MT and other Language Technology Tools, Improving MT through Other Language Technology Tools: Resources and Tools for Building MT, Stroudsburg, PA, USA,2003.

[4] Han A L F, Zeng X D, Wong D F, et al. Chinese named entity recognition with graph-based semi-supervised learning model[C]//Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing (SIGHAN-8), Stroudsburg, PA, USA, 2015.

[5] Dong C H, Zhang J J, Zong C Q, et al. Character-based LSTM-CRF with radical-level features for Chinese named entity recognition[C]//International Conference on Computer Processing of Oriental Languages, New York City, NY, USA, 2016.

[6] Lu Y N, Zhang Y, Ji D H. Multi-prototype Chinese character embedding[C]//Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia, 2016.

[7] Zheng X Q, Feng J T, Lin M X, et al. Context-specific and multi-prototype character representations [C]//Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), Palo Alto, CA, USA,2016.

[8] Zheng X Q, Feng J T, Chen Y, et al. Learning context-specific word/character embeddings[C]//Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA,2017.

[9] Zhang Y, Yang J. Chinese NER using Lattice LSTM [C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), Stroudsburg, PA, USA, 2018.

[10] Gao J F, Li M, Wu A, et al. Chinese word segmentation and named entity recognition: a pragmatic approach[J]. Computational Linguistics, 2005, 31 (4):531 – 574.

[11] Zhang S X, Qin Y, Wen J, et al. Word segmentation and named entity recognition for SIGHAN Bakeoff3[C]//Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, Stroudsburg, PA, USA, 2006.

[12] Mao X N, Dong Y, He S, et al. Chinese word segmentation and named entity recognition based on conditional random fields[C]//Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing, Stroudsburg, PA, USA, 2008.

[13] Luo W C, Yang F. An empirical study of automatic Chinese word segmentation for spoken language understanding and named entity recognition[C]//Proceedings of NAACL-HLT 2016, Stroudsburg, PA, USA,2016.

[14] Peng N Y, Dredze M. Named entity recognition for Chinese social media with jointly trained embeddings[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA,2015.

[15] Peng N Y, Dredze M. Improving named entity recognition for Chinese social media with word segmentation representation learning[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA,2016.

[16] He H F, Sun X. F-score driven max margin neural network for named entity recognition in Chinese social media[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Stroudsburg, PA, USA, 2017.

[17] Kuru O, Can O A, Yuret D. CharNER: Character-level named entity recognition[C]//Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Stroudsburg, PA, USA, 2016.

[18] Collobert R, Weston J. A unified architecture for

natural language processing: Deep neural networks with multitask learning[C]// Proceedings of the 25th International Conference on Machine Learning, Stroudsburg, PA, USA, 2008.

[19] Chiu J P C, Nichols E. Named entity recognition with bidirectional LSTM-CNNs[J]. Transactions of the Association for Computational Linguistics, 2016, 4: 357 – 370.

[20] Liu X H, Wei F R, Zhang S D, et al. Named entity recognition for tweets[J]. ACM Transactions on Intelligent Systems and Technology (TIST)-Special Section on Twitter and Microblogging Services, Social Recommender Systems, and CAMRa2010: Movie Recommendation in Context, 2013, 4(1): 3.

[21] Chou C L, Chang C H, Huang Y Y. Boosted web named entity recognition via tri-training[J]. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)-TALLIP Notes and Regular Papers, 2016, 16(2): 10.

[22] Bikel D M, Miller S, Schwartz R, et al. Nymble: a high-performance learning name-finder[C]// Proceedings of the Fifth Conference on Applied Natural Language Processing, Stroudsburg, PA, USA, 1997.

[23] Sekine S. NYU: Description of the Japanese NE system used for MET-2[C]// Proceedings of the Seventh Message Understanding Conference (MUC-7), Stroudsburg, PA, USA, 1998.

[24] Borthwick A, Sterling J, Agichtein E, et al. NYU: Description of the MENE named entity system as used in MUC-7[C]// Proceedings of the Seventh Message Understanding Conference (MUC-7), Stroudsburg, PA, USA, 1998.

[25] Isozaki H, Kazawa H. Efficient support vector classifiers for named entity recognition[C]// Proceedings of the 19th International Conference on Computational Linguistics - Volume 1, Stroudsburg, PA, USA, 2002.

[26] Takeuchi K, Collier N. Use of support vector machines in extended named entity recognition[C]// Proceedings of the 6th Conference on Natural Language Learning-Volume 20, Stroudsburg, PA, USA, 2002.

[27] Kazama J, Makino T, Ohta Y, et al. Tuning support vector machines for biomedical named entity recognition[C]// Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain-Volume 3, Stroudsburg, PA, USA, 2002.

[28] Asahara M, Matsumoto Y. Japanese named entity extraction with redundant morphological analysis[C]// Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, Stroudsburg, PA, USA, 2003.

[29] McCallum A, Li W. Early results for named entity recognition with conditional random fields, features induction and web-enhanced lexicons[C]// Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, Stroudsburg, PA, USA, 2003.

[30] Hao Z F, Wang H F, Cai R C, et al. Product named entity recognition for Chinese query questions based on a skip-chain CRF model[J]. Neural Computing and Applications, 2013, 23(2): 371 – 379.

[31] Patra R, Saha S K. A Kernel-based approach for biomedical named entity recognition[J]. The Scientific World Journal, 2013, 2013: 950796.

[32] Chen A T, Peng F C, Shan R, et al. Chinese named entity recognition with conditional probabilistic models[C]// Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, Stroudsburg, PA, USA, 2006.

[33] Zhou J S, Qu W G, Zhang F. Chinese named entity recognition via joint identification and categorization[J]. Chinese Journal of Electronics, 2013, 22(2): 225 – 230.

[34] Chorowski J, Bahdanau D, Cho K, et al. End-to-end continuous speech recognition using attention-based recurrent NN: First results[EB/OL]. [2014-12-04]. https://arxiv.org/abs/1412.1602.

[35] Hermann K M, Kočiský T, Grefenstette E, et al. Teaching machines to read and comprehend[C]// Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, Cambridge, MA, USA, 2015.

[36] Vinyals O, Kaiser L, Koo T, et al. Grammar as a foreign language[C]// Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, Cambridge, MA, USA, 2015.

[37] Bapna A, Chen M X, Firat O, et al. Training deeper neural machine translation models with transparent attention[EB/OL]. [2018-09-04]. https://arxiv.org/abs/1808.07561.

[38] Bahdanau D, Cho K H, Bengio Y. Neural machine translation by jointly learning to align and translate[EB/OL]. [2016-05-19]. https://arxiv.org/abs/

1409.0473.

[39] Xu K, Ba J L, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention[C]//Proceedings of the 32nd International Conference on Machine Learning (ICML'15), Lille, France, 2015.

[40] Shang L F, Lu Z D, Li H. Neural responding machine for short-text conversation[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Stroudsburg, PA, USA, 2015.

[41] Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 2015.

[42] Graves A. Generating sequences with recurrent neural networks[EB/OL]. [2014-06-05]. https://arxiv.org/abs/1308.0850.

[43] Graves A, Wayne G, Danihelka I. Neural turing machines[EB/OL]. [2014-12-10]. https://arxiv.org/abs/1410.5401.

[44] Graves A, Wayne G, Reynolds M, et al. Hybrid computing using a neural network with dynamic external memory[J]. Nature, 2016, 538: 471 – 476.

[45] Gu J T, Lu Z D, Li H, et al. Incorporating copying mechanism in sequence-to-sequence learning[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2016.

[46] Gulcehre C, Ahn S, Nallapati R, et al. Pointing the unknown words[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2016.

[47] Cao Z Q, Luo C W, Li W J, et al. Joint copying and restricted generation for paraphrase[C]//Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA, 2017.

[48] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.

[49] Choi H, Cho K, Bengio Y. Fine-grained attention mechanism for neural machine translation[J]. Neurocomputing, 2018, 284: 171 – 176.

[50] Elbayad M, Besaier L, Verbeek J. Pervasive attention: 2D convolutional neural networks for sequence-to-sequence prediction[EB/OL]. [2018-08-11]. https://arxiv.org/abs/1808.03867.

[51] Hu M H, Peng Y X, Huang Z, et al. Reinforced mnemonic reader for machine reading comprehension[C]//Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 2018.

[52] Shen T, Zhou T Y, Long G D, et al. DiSAN: directional self-attention network for RNN/CNN-Free language understanding[EB/OL]. [2017-11-20]. https://arxiv.org/abs/1709.04696.

[53] Shen T, Zhou T Y, Long G D, et al. Bi-directional block self-attention for fast and memory-efficient sequence modeling[EB/OL]. [2018-04-03]. https://arxiv.org/abs/1804.00857.

[54] Strubell E, Verga P, Andor D, et al. Linguistically-informed self-attention for semantic role labeling [EB/OL]. [2018-08-28]. https://arxiv.org/abs/1804.08199.

[55] Tang G B, Müller M, Rios A, et al. Why self-attention? A targeted evaluation of neural machine translation architectures[EB/OL]. [2018-08-28]. https://arxiv.org/abs/1808.08946.

[56] Yu A W, Dohan D, Luong M T, et al. QANET: Combining local convolution with global self-attention for reading comprehension[EB/OL]. [2018-04-23]. https://arxiv.org/abs/1804.09541.

[57] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735 – 1780.

[58] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures[J]. Neural Networks, 2005, 18(5 – 6): 602 – 610.

[59] Lafferty J D, Mccallum A, Pereira F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[C]//Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), San Mateo, CA, USA, 2001.

[60] Levow G A. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition[C]//Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, Stroudsburg, PA, USA, 2006.

(**Edited by** Jianying Cai)