information

Article

CWPC_BiAtt: Character–Word–Position Combined BiLSTM-Attention for Chinese Named Entity Recognition

Shardrom Johnson [1,2,3,*], Sherlock Shen [1] and Yuanchen Liu [4]

[1] School of Optical-electrical and Computer Engineering, University of Shanghai for Science and Technology, Jungong Road 516, Shanghai 200093, China; 172590530@st.usst.edu.cn
[2] XianDa College of Economics and Humanities, Shanghai International Studies University, East Tiyuhui Road 390, Shanghai 200083, China
[3] Information Centre, Shanghai Municipal Education Commission, Dagu Road 100, Shanghai 200003, China
[4] Faculty of Foreign Languages, Ningbo University, Fenghua Road 818, Ningbo 315211, China; 1811051117@nbu.edu.cn
* Correspondence: jshardrom@shec.edu.cn

Received: 28 November 2019; Accepted: 13 January 2020; Published: 15 January 2020

Abstract: Usually taken as linguistic features by Part-Of-Speech (POS) tagging, Named Entity Recognition (NER) is a major task in Natural Language Processing (NLP). In this paper, we put forward a new comprehensive-embedding, considering three aspects, namely character-embedding, word-embedding, and pos-embedding stitched in the order we give, and thus get their dependencies, based on which we propose a new Character–Word–Position Combined BiLSTM-Attention (CWPC_BiAtt) for the Chinese NER task. Comprehensive-embedding via the Bidirectional Llong Short-Term Memory (BiLSTM) layer can get the connection between the historical and future information, and then employ the attention mechanism to capture the connection between the content of the sentence at the current position and that at any location. Finally, we utilize Conditional Random Field (CRF) to decode the entire tagging sequence. Experiments show that CWPC_BiAtt model we proposed is well qualified for the NER task on Microsoft Research Asia (MSRA) dataset and Weibo NER corpus. A high precision and recall were obtained, which verified the stability of the model. Position-embedding in comprehensive-embedding can compensate for attention-mechanism to provide position information for the disordered sequence, which shows that comprehensive-embedding has completeness. Looking at the entire model, our proposed CWPC_BiAtt has three distinct characteristics: completeness, simplicity, and stability. Our proposed CWPC_BiAtt model achieved the highest F-score, achieving the state-of-the-art performance in the MSRA dataset and Weibo NER corpus.

Keywords: entity; comprehensive-embedding; completeness; simplicity; stability; BiLSTM; attention mechanism; Conditional Random Field

1. Introduction

Named Entity Recognition (NER) plays an important role in the field of natural language processing. In recent years, it has gradually become an essential component of information extraction technologies [1]. NER serves many Natural Language Processing (NLP) downstream tasks, for example, event extraction [2], relation extraction [3,4], entity linking [5,6], and question answering [7]. In the field of machine learning, methods like making full use of Support Vector Machine (SVM) to treat the problem of NER [8–10]. As a classic binary classification algorithm, the SVM combined can realize multiple classification. However, the training of each classifier takes all samples as data. When solving the quadratic programming, the training speed will slow down as the training sample increases. So it is necessary to identify multiple classes of NER tasks that NER's training corpus is often very large, and the use of SVM to process NER tasks will be very inefficient, ignoring the connection between contexts in the statement. The article [11,12] proposed to use Hidden Markov Model (HMM) to deal with NER tasks, and good results were achieved, but HMM is limited by Homogeneous Markov and Observation Independence Hypothesis. Homogeneous Markov enables the state at any moment to solely depend on its previous moment (namely only consider the previous character or word of the current character or word), so that only a small amount of connection between contexts is considered, and thus resulting in an error in the final annotation. In [13], the author proposed Conditional Random Field (CRF) to overcome the limitations of HMM and solve the label bias problem. In [14], the author used CRF to handle NER tasks and achieved good results. But the author did not consider whether the connection between characters, words, and positions in the sentence would affect the final result.

In recent years, with the rise of neural networks, it has become synonymous with deep learning. The field of image recognition, natural language processing, etc., witnessed a climax of using neural networks. The initial structure proposed in Recurrent Neural Network (RNN) [15–17] as time goes on has evolved over time into a mature RNN. RNN can store the historical information, so RNN is very qualified to handle long text types of NLP tasks. But RNN has the problem of vanishing gradient. In a given time series, RNN cannot capture the dependency relationship between two text elements staying far away from each other. From the perspective of performance of current parties, the performance of neural networks in NER has surpassed that of traditional machine learning, improving NER's various indicators like the accuracy rate by a great deal. The Long Short-Term Memory (LSTM) was proposed in 1997 by [18], having achieved unprecedented performance in the field of NLP in recent years. In the NER task, the use of Bidirectional Long Short Term Memory-Convolutional Neural Networks (BiLSTM-CNNS) on the CoNLL-2003 data set by [19] achieved a good score of 91.23%. The use of the attention mechanism achieved great performance in machine translation by [20], setting off a new wave in the NLP field. The utilization of attention mechanism by [21] in F1-score reached 90.64% in the SIGHAN data set.

In addition, Out-Of-Vocabulary (OOV) also poses another challenge to NER. If processing NER only considers word-level embedding, that will ignore the connection between characters, and the word-level embedding has difficulty to capture all the details. So, we propose a new comprehensive-embedding as pre-training embedding, which is a stitching of character-embedding, word-embedding, and pos-embedding in the order we give. Taking into account the above considerations, we finally proposed a new model, Character-Word-Position Combined BiLSTM-Attention (CWPC_BiAtt), for Chinese NER task.

2. Related Work

2.1. NER

For the NER task, it can be roughly divided into two ways, one is to use the traditional machine learning method, and the other is the neural network method. Before the neural network method became mainstream, traditional machine learning was the main method in dealing with the NER task. [22] used conditional random fields and maximum entropy models to perform the Chinese NER task. [23] proposed the Multi-Phase Model. Both of them can be regarded as pioneers in fulfilling the Chinese NER task by using traditional machine learning methods. However, the disadvantage of using traditional machine learning to handle NER tasks is very significant because of tremendously huge feature engineering. Even if many features are built, there is no guarantee that each feature can improve the NER task. So, feature selection is another issue. Although feature selection is performed, there is no guarantee that features not selected are meaningless to the final result.

With the rise of neural networks methods, researchers began to adopt deep learning algorithms such as LSTM [18] and Gated Recurrent Unit (GRU) [24] in the field of natural language processing.

Rather than using handcrafted features, neural network method shows extraordinary strengths in areas such as image processing and natural language processing. [21,25–27] used LSTM-CRF as the baseline. Except [27] used one-way LSTM, the others chose BiLSTM to capture historical and future information. From the F1 scores of their experiments, the scores obtained with BiLSTM were higher than those of LSTM. In addition, for the NER task, it is very important to choose a good pre-train representation. In [27], the author conducted combined training of the Chinese word segmentation (CWS) and NER. Considering that there is no natural interval in Chinese sentences, it is not necessarily the same when segmenting words. CWS and NER can be optimized through loss. However, there is a problem that CWS and NER can affect each other, so between the two variables, it cannot be determined which one has a larger impact. Moreover, the author adopted forward LSTM rather than backward LSTM to capture the future information. [26] is based on character without considering whether word and position influence the NER task. The author used GRU to capture the historical and future information. Although the effect was good, the author didn't consider that GRU was not suitable for many parameters. Moreover, the author used character as input, so the parameters would be large. To perform many NLP tasks, researchers will use BiLSTM-CRF or BiGRU-CRF as the baseline, and then adjust the model based on this, to reduce the cost of trial and error for researchers. BiLSTM-CRF has become a classic baseline, contributing a lot in the field of NLP. So, this paper also makes improvement taking BiLSTM-CRF as the baseline. [28] proposed word-embedding, and [29] put forward the global vector. Both methods serve well for many NLP downstream tasks. [30] proposed Embeddings from Language Model (ELMO) to train word embedding and suggest to use different tags for expressing different meanings of the same word. [31] put forward Bidirectional Encoder Representations from Transformers (BERT) method to train word-embedding. During the word-embedding training process, this method will mask a small part of each sentence for training. Interestingly, both ELMO and BERT are roles in Sesame Street, and people familiar with children's programs should know them.

2.2. Attention Mechanism

In the 1990s, after more than 20 years of silence, until the google mind team [32] used the Attention Mechanism for image classification on the RNN model. this mechanism was first proposed in the field of visual images. Subsequently, [33] adopted Attention Mechanism to simultaneously translate and align in machine translation tasks. Their work was the first try to apply Attention Mechanism to the NLP field. Not until it became the research focus, the Valley Machine Translation Team [20] used the attention mechanism. [21] proposed adversarial transfer learning framework. With usage of the attention mechanism, it can capture global dependency and the internal structural features of the entire sentence. [26] used the attention mechanism to grab the local information at the Convolutional Attention layer, and then retrieved the information at the GRU layer by adopting the global-attention mechanism. It is precisely because of the continuous efforts of the predecessors that the attention mechanism has achieved great success in various fields.

3. Model

This paper utilizes BiLSTM-CRF as the baseline, which seems to have been bound with NLP. Our model considers the multi-level context, divided into 4 layers: (1) Stitching character-embedding, word-embedding, pos-embedding to obtain comprehensive-embedding. (2) Connecting comprehensive-embedding to BiLSTM Neural Network and get the top hidden unit. (3) Connect the obtained hidden unit to the attention mechanism for processing. (4) Use CRF to decode the data of the previous layer. The overall structure of Character-Word-Position Combined Bidirectional Long Short Term Memory-Attention (CWPC_BiAtt) we proposed is shown as Figure 1.

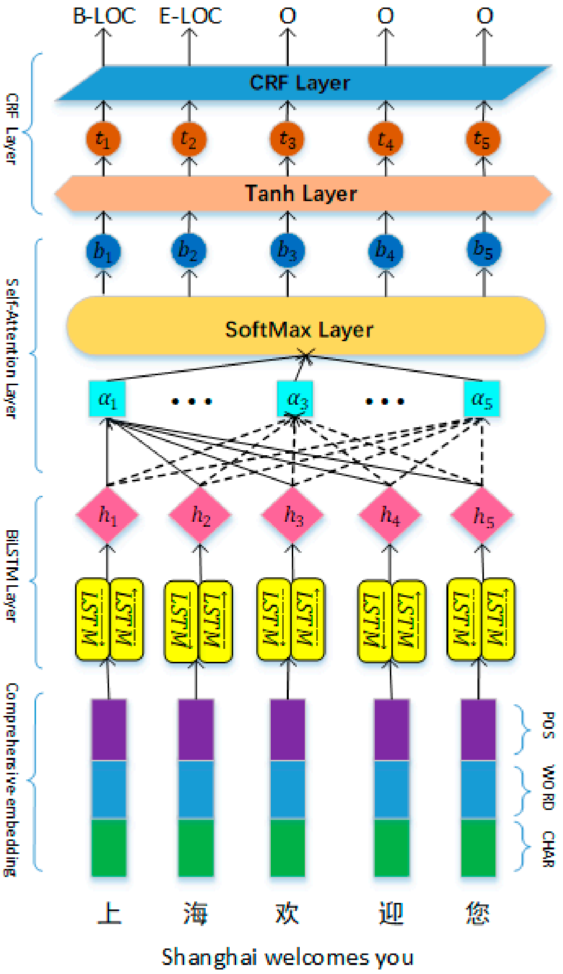

Figure 1. The overall structure of Character-Word-Position Combined Bidirectional Long Short Term Memory-Attention (CWPC_BiAtt) is divided into four layers: (1) comprehensive-embedding, which is composed of character-embedding, word-embedding, and position-embedding. (2) Bidirectional Long Short Term Memory (BiLSTM) layer, the role of this layer is to capture historical information and future information in the sentence of this paper. (3) self-attention layer captures the connection between several arbitrary positions in the sentence, using softmax to normalize the value output by attention-mechanism. (4) conditional random field(CRF) layer first uses tanh function to quickly change the gap between features, and then uses CRF for marking. B in B-LOC means Begin, LOC means location, E in E-LOC means end, and LOC means location.

3.1. Comprehensive-Embedding

In the NER task, we select a sentence $X_i^c = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ from the pre-trained data set, in which x_{in} represents the n-th character-embedding of the i-th sentence in the pre-trained data set, and $x_{in} \in \mathbb{R}^{d_c}$, and d_c is the dimension of character-embedding. We train word-embedding according to "word representations in vector space" proposed by [28], and obtain $X_i^w = \{w_{i1}, w_{i2}, \ldots, w_{is}\}$, in which w_{is} the word-embedding of the s-th in the i-th sentence in the pre-training data, and $w_{is} \in \mathbb{R}^{d_w}$, d_w is the dimension of word-embedding. The comprehensive-embedding stitching in order we proposed also needs to consider pos-embedding. As for pos-embedding, this paper adopts one-hot encoding, set as 1 when the character appears, and otherwise 0. That is, $X_i^p = \{p_{i1}, p_{i2}, \ldots, p_{in}\}$, in which $p_{in} \in \mathbb{R}^{d_p}$, d_p is the dimension of pos-embedding. Although one-hot encoding has the problem of sparseness, in this paper, we do not perform one-hot encoding on the Chinese characters, but encoding the position of the characters. In the Chinese corpus, the average length of each sentence ranges from 25 to 35, so the length is not very sparse for position-embedding. After the Chinese sentence is converted to comprehensive-embedding, it can be found that if only look at the part represented by position-embedding, it must be the symmetric matrices with values on the main diagonal are easy for the calculation. Therefore, comprehensive-embedding is expressed as $\mathrm{comp}_{it} = \mathrm{concatenate}\left[x_{it}, w_{i\tilde{t}}, p_{it}\right]$, $\mathrm{comp}_{it} \in \mathbb{R}^{d_c + d_w + d_p}$. And $w_{i\tilde{t}}$ in comp_{it} is determined according to x_{it}, there is a case that the subscript index $\tilde{t}$ in clause plus word-embedding $w_{i\tilde{t}}$ and the subscript index t of char-embedding x_{it} may not be the same. The reason behind this is that Chinese sentences have no natural interval, and both character and word exist after the split. The structure of comp_{it} is shown as Figure 2. In the following, we will illustrate with an example, so the index i can be omitted, and comp_{it} can be simplified to comp_t, and used as the input by BiLSTM Layer, and $t \in [1, n]$. Because the character is sequential, so it can be seen as n moments, and t is the t-th moment from 1 to n.

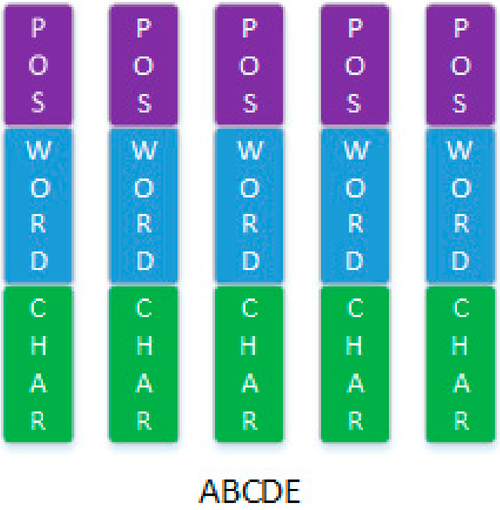

Figure 2. To facilitate the explanation of comprehensive-embedding, we assume sentence "ABCDE" is a Chinese sentence. Suppose sentence "ABCDE" can become "AB/CD/E" after the word segmentation. AB, CD, E are Chinese words after the word segmentation. This sentence consists of five comprehensive-embedding: (1) character (A) + word (AB) + pos (first position); (2) character (B) + word (AB) + pos (second position); (3) character (C) + word (CD) + pos (third position); (4) character (D) + word (CD) + pos (forth position); (5) character (E) + word (E) + pos (fifth position).

3.2. BiLSTM Layer

The continuous maturity of RNN has greatly improved the processing of the sequence tag problem. But RNN has the problem of vanishing gradient, and in a given time series, it is impossible to capture the dependencies between two text elements that far apart from each other.

LSTM [18], as a variant of RNN, can well improve the problem of vanishing gradient. LSTM can well grab the dependencies between several text elements that far apart from each other. Capturing the link between past historical information and future information, LSTM can do a good job of handling sequence tagging. And one-way LSTM can only get historical information, but ignore the future information. The BiLSTM was used by [34] to perform the Relation Classification perfectly. In considering the future information, we also need to access a backward LSTM. The specific model of LSTM is shown as Figure 3. The main structure of LSTM is as follows:

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, \mathrm{comp}_t\right] + b_i\right), \tag{1}$$

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, \mathrm{comp}_t\right] + b_f\right), \tag{2}$$

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, \mathrm{comp}_t\right] + b_o\right), \tag{3}$$

$$\widetilde{c_t} = \tan\left(W_c \cdot \left[h_{t-1}, comp_t\right] + b_c\right), \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t}\,, \tag{5}$$

$$h_t = o_t \odot \tan(c_t). \tag{6}$$

i_t, f_t, o_t represents input gate, forget gate, and output gate respectively. And σ and $\odot$ is respectively sigmoid function and element-wise product. W_i, W_f, W_o is respectively the weight matrix of the input gate, the forget gate, and the output gate. And b_i, b_f, b_o is respectively the bias of the input gate, the forget gate, and the output gate. $comp_t$ represents the input at time t, and h_{t-1} the hidden state at time $t-1$, namely the short-term memory output at time $t-1$. It can be found from equations (1, 2, 3) that three gates all represent a layer of perceptron network. In [18], the appendix of this paper, Hochreiter and Schmidhuber adopted the backpropagation to calculate W_i, W_f, W_o. Therefore, the weight matrix to be learned in this paper can be solved based on back propagation. $\widetilde{c_t}$ represents the current memory at time t, that calculated from the hidden state h_{t-1} at time $t-1$ and the input $comp_t$ at time t, W_c and b_c are the corresponding weight matrix and bias can be calculated respectively. c_t represents the unit state at time t, namely the long-term memory at time t, calculated from the unit state c_{t-1} and forget gate f_t, at time $t-1$. when the value of f_t is 0, it means to forget the previous c_{t-1}, and when the value of f_t is 1, it means that the previous c_{t-1} is remembered and retained to c_t. h_t represents the hidden state at time t, namely the short-term memory at time t. It is calculated by the output gate o_t and unit state c_t. When the value of o_t is 0, it means that there is no short-term memory. When the value of o_t is 1, it means that there is short-term memory. The paper adopted the Bidrectional LSTM network which is composed of forward LSTM and backward LSTM. $\overrightarrow{h_t}$, $\overleftarrow{h_t}$ are used separately to distinguish the hidden states. So, h_t is the concatenate of $\overrightarrow{h_t}$, $\overleftarrow{h_t}$, and h_t is shown as below.

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t}, \tag{7}$$

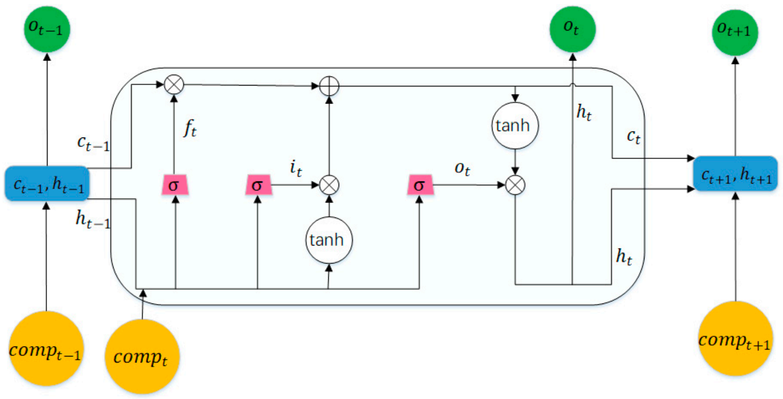

Figure 3. The structure of Long Short Term Memory.

The sequence of the top-level hidden unit of the final output:

$$H = \{h_1, h_2, \ldots, h_n\}. \tag{8}$$

And $\oplus$ represents concatenate, $\overrightarrow{h_t} \in \mathbb{R}^{d_h}$, $\overleftarrow{h_t} \in \mathbb{R}^{d_h}$, $h_t \in \mathbb{R}^{2d_h}$. On the second layer of the model, BiLSTM is used to get the correlation between contexts, but there has certain defects. When the value of forget gate remains at 0, the previous memory will be cleared, so BiLSTM can't capture the relation between the subsequent content and cleared content. To make up the shortcoming, we will use the attention mechanism and grab the connection between any two characters. Therefore, it will improve the accuracy ofrelation between captured contexts. Nowadays, LSTM proposed by Hochreiter and

Schmidhuber has becomeclassic method, and for scholars who study NLP, LSTM is an algorithm that must be learned. For processing NER tasks, using LSTM is currently very popular.

3.3. Attention Layer

Attention Layer is inspired by the attention mechanism by [20] on machine translation. Since Vaswani et al. published the paper [20], Attention-Mechanism has been extensively used in NLP tasks, because it can break the limitations of sentence serialization. We use the attention mechanism to study the dependencies between any two characters in a sentence and capture internal structural information. Currently, attention mechanism is the most effective method that can be used to test if there is dependency relation between the two characters which is captured randomly. And it has greatly improved the NLP task and has made important contributions. In Section 3.2, we take the hidden unit sequence $\mathrm{H} = \{\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_\mathrm{n}\}$ at the top of the BiLSTM layer as the input of the attention layer in this section.

Multi-head self-attention can perform its own duty, for example, some self-attention consider local consultation, and some self-attention considers global consultation. Self-attention is good at capturing the connections between several arbitrary locations in a sequence. Since the multi-head self-attention is a multiple self-attention parallel, the final result needs merging. As to the considerations of global consultation and local consultation, self-attention is divided into global self-attention and local self-attention. The following is exemplified by a single self-attention for explanation.

3.3.1. Global Self-Attention

Global self-attention is a scaled dot-product attention operation on the hidden unit sequence $\mathrm{H} = \{h_1, h_2, \ldots, h_n\}$ obtained from the BiLSTM Layer in Section 2. as follows:

$$\mathrm{B}^\mathrm{G} = \sum_{\mathrm{j}=1}^{\mathrm{n}} \alpha_\mathrm{j}^\mathrm{G} \mathrm{h}_\mathrm{j}^\mathrm{G}, \tag{9}$$

where $\mathrm{B}^\mathrm{G} = \left\{\mathrm{b}_1^\mathrm{G}, \mathrm{b}_2^\mathrm{G}, .., \mathrm{b}_\mathrm{n}^\mathrm{G}\right\}$, B^G is the dependency extracted by global self-attention. $\mathrm{h}_\mathrm{j}^\mathrm{G}$ is equivalent to h_j, and the adding of a superscript is to distinguish the operation between global self-attention and local self-attention, using different tags. $\alpha_\mathrm{j}^\mathrm{G}$ is the weight of the attention mechanism, we need to calculate the score of the j-th target $\mathrm{h}_\mathrm{j}^\mathrm{G}$ and the current t-th target $\mathrm{h}_\mathrm{t}^\mathrm{G}$, and the specific calculation is as follows:

$$\alpha_\mathrm{j}^\mathrm{G} = \frac{\exp(\mathrm{score}^\mathrm{G}\left(\mathrm{h}_\mathrm{t}^\mathrm{G}, \mathrm{h}_\mathrm{j}^\mathrm{G}\right))}{\sum_{\mathrm{k}\in\{1,2,\ldots,\mathrm{n}\}} \exp(\mathrm{score}^\mathrm{G}\left(\mathrm{h}_\mathrm{t}^\mathrm{G}, \mathrm{h}_\mathrm{k}^\mathrm{G}\right))}, \tag{10}$$

The score function is shown as follows:

$$\mathrm{score}^\mathrm{G} = \mathrm{v}_\mathrm{G}^\mathrm{T} \tan\mathrm{h}\left(\mathrm{W}_{\mathrm{B}^\mathrm{G}}\left[\mathrm{h}_\mathrm{t}^\mathrm{G}, \mathrm{h}_\mathrm{j}^\mathrm{G}\right]\right). \tag{11}$$

We can find that in fact, Equation (11) is a Multilayer perceptron, and $\mathrm{W}_{\mathrm{B}^\mathrm{G}}$ is the weight matrix for the first layer while $\mathrm{v}_\mathrm{G}^\mathrm{T}$ is for the second layer. By feedforward network, current situation $\mathrm{h}_\mathrm{t}^\mathrm{G}$ and code-ended information $\mathrm{h}_\mathrm{j}^\mathrm{G}$ are calculated by activation function tanh, then this data is given to softmax function in Equation (10) to figure up the loss between the output and the answer. Through back propagation two parameters $\mathrm{W}_{\mathrm{B}^\mathrm{G}}$, $\mathrm{v}_\mathrm{G}^\mathrm{T}$ are learned.

3.3.2. Local Self-Attention

The global self-attention of all comprehensive-embedding in the source sentence may be costly and unnecessary. To solve this problem, this paper uses multi-head self-attention, some are local self-attention, and others global self-attention. When using local self-attention, we set the window size at 2w + 1 so that the target h_j only considers each position before and after. The local self-attention of $\{h_{i-w}, \ldots, h_i, \ldots, h_{i+w}\}$ is similar to the global self-attention in Section 3.3.1 with only minor changes. It is expressed as follows:

$$B^L = \sum_{j=1}^{n} \alpha_j^L h_j^L, \tag{12}$$

$$\alpha_j^L = \frac{\exp\left(\text{score}^L\left(\left(h_t^L,\ h_j^L\right)\right)\right)}{\sum_{k \in \{i-w,..,i,...,i+w\}} \exp\left(\text{score}^L\left(\left(h_t^L, h_k^L\right)\right)\right)}, \tag{13}$$

$$\text{score}^L = v_L^T \tan h\left(W_{B^L}\left[h_t^L, h_j^L\right]\right). \tag{14}$$

where $B^L = \left\{b_1^L, b_2^L, .., b_n^L\right\}$, B^L is the dependency extracted by global self-attention. The weight matrix v_L^T and W_{B^L} is the parameter of the model training. The detailed explanation of the score function Equation (14) is similar to the Equation (11) in Section 3.3.1.

3.3.3. Multi-Head Self-Attention

This paper uses Multi-head self-attention, which requires to concatenate the results of all global self-attention and local self-attention. The results are shown as follows:

$$B = \{b_1, b_2, .., b_n\}, \tag{15}$$

where $b_i = \text{concatenate}\left\{b_i^{G1}, b_i^{G2}, .., b_i^{L1}, b_i^{L2}, \ldots\right\}$ concatenating all global and local components.

3.4. CRF Layer

We connect $B = \{b_1, b_2, .., b_n\}$ obtained in Section 3.3.3 into tanh layer for the treatment by the belief function so that all values in the value B fall within (−1, 1) as follows:

$$T = \sum_{i=0}^{n} \tan h(W_t b_i), \tag{16}$$

where $T = \{t_1, t_2, \ldots, t_n\}$, W_t is model training parameter.

Compared with Hidden Markov Model (HMM), CRF's condition is much looser. CRF is not limited to: (1) Homogeneous Markov, (2) Observation Independence Hypothesis. Compared to Maximum entropy Markov models (MEMMs) proposed by [35], CRF means global normalization while MEMMs local normalization. So CRF effectively solves the shortcomings of MEMMs: the label bias problem was proposed by [13]. Therefore, it can be said that CRF is a very effective for dealing with sequence marking problems.

The output value of the Tanh Layer $T = \{t_1, t_2, \ldots, t_n\}$ is used as the input of the CRF layer. In order to follow the representation of the CRF algorithm convention, we rewrite $T = \{t_1, t_2, \ldots, t_n\}$ into $X = \{x_1, x_2, \ldots, x_n\}$, where t_i and x_i are equivalent.

Conditional Random Field (CRF) is a Conditional probability model that will give a set of variable conditions $X = \{x_1, x_2, \ldots, x_n\}$ and another set of outputs $Y = \left\{y_1, y_2, \ldots, y_n\right\}$, and Y is the corresponding label of X. The overall structure of the CRF is as follows:

$$P\left(Y = y \middle| X = x\right) = \frac{1}{Z(x, \theta)} \exp\left(\theta^T \cdot H\left(y_{t-1}, y_t, x\right)\right), \tag{17}$$

where $H\left(y_{t-1}, y_t, x\right)$ is the score function, as shown below.

$$H\left(y_{t-1}, y_t, x\right) = \begin{pmatrix} \sum_{t=1}^{n} f \\ \sum_{t=1}^{n} g \end{pmatrix}_{K+L}, \tag{18}$$

where f is the total transfer function of the score function $\mathrm{H}\left(y_{t-1}, y_t, x\right)$, and g is the total state function of the score function $\mathrm{H}\left(y_{t-1}, y_t, x\right)$. K and L respectively represent the dimension of total transfer function and the total state function.

The total transfer function is as follows:

$$\mathrm{f} = \begin{pmatrix} f_1 \\ f_2 \\ \cdots \\ f_K \end{pmatrix} = f\left(y_{t-1}, y_t, x\right), \tag{19}$$

where f_k refers to the transfer function in the total transfer function, as shown below:

$$f_k = f_k\left(y_{t-1}, y_t, x\right). \tag{20}$$

The total state function is as follows:

$$g = \begin{pmatrix} g_1 \\ g_2 \\ \cdots \\ g_L \end{pmatrix} = g\left(y_t, x\right), \tag{21}$$

where g_l is the state function in the total state function, as shown below:

$$g_l = g_l\left(y_t, x\right). \tag{22}$$

In Equation (17), Z is the normalization factor of softmax and θ is the total parameter learned by the total transfer function f and the total state function g. The dimension of the total parameter θ is K + L and the parameter θ are shown as follows:

$$\theta = \begin{pmatrix} \lambda \\ \eta \end{pmatrix} K + L, \tag{23}$$

where $\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \cdots \\ \lambda_K \end{pmatrix}, \eta = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \cdots \\ \eta_L \end{pmatrix}$. λ_k is the parameter learned by transfer function while η_l learned by state function.

Special attention should be given to $x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix}, y = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}$ in Equation (17).

Substitute Equations (18)–(23) in Equation (17) as follows:

$$P\left(Y = y \middle| X = x\right) = \frac{1}{Z} \exp \sum_{t=1}^{n} \left[\sum_{k=1}^{K} \lambda_k \cdot f_k\left(y_{t-1}, y_t, x_{1:n}\right) + \sum_{l=1}^{L} \eta_l \cdot g_l\left(y_t, x_{1:n}\right) \right], \tag{24}$$

Then, we use Maximum likelihood estimation to solve for Equation (17) or Equation (24). Here we only demonstrate the solution to Equation (17) as follows:

$$\log P\left(Y = y \middle| X = x\right) = \theta^T \cdot H\left(y_{t-1}, y_t, x\right) - \log Z(x, \theta) = \theta^T \cdot H\left(y_{t-1}, y_t, x\right) - \log \sum_{y' \in \mathbb{Y}} \theta^T \cdot H\left(y'_{t-1}, y'_t, x\right), \tag{25}$$

where $\mathbb{Y}$ is all possible tags that represent the sentence *x*.

In summary, we encourage the model to generate valid output labels. During the decoding process, we predict the maximum score of the output sequence. For the algorithm of the solution, we use the dynamic programming algorithm to solve the problem. This paper uses Viterbi-algorithm [36] to solve the maximum score, as shown below.

$$y^{*} = \underset{y' \in \mathbb{Y}}{\operatorname{argmax}} \theta^{T} \cdot H\left(y'_{t-1}, y'_{t}, x\right). \tag{26}$$

4. Experiments and Analyze

4.1. Evaluation Metrics

The evaluation metrics of the NER task mainly include precision, recall, and F1 scores. Precision refers to the proportion of samples that are predicted to be positive. There are two possibilities for predicting positive, (1) predicting positive class as true positive (TP), (2) predicting negative class as false positive (FP). Precision can be defined as follows:

$$P = \frac{TP}{TP + FP}, \tag{27}$$

Recall rate refers to the proportion of the sample being predicted correctly. There are also two possible cases: (1) if the prediction is correct, the positive class is predicted to be true positive (TP). (2) if the prediction is failed, and the positive class is predicted as a false negative (FN), so the recall rate is defined as the following formula:

$$R = \frac{TP}{TP + FN}, \tag{28}$$

However, sometimes precision and recall may contradict, that is, the accuracy rate increases while the recall rate decreases, the accuracy rate decreases while the recall rate increases. The F-measure method needs to be introduced comprehensive measurement. This paper uses the most common calculation method:

$$F1 = \frac{2 \times P \times R}{P + R}. \tag{29}$$

4.2. Datasets

We propose a comparison experiment using CWPC_BiAtt and previous state-of-the-art method on MSRA dataset and Weibo NER corpus. The dataset parameters required for the experiment are shown as Table 1.

Table 1. Microsoft Research Asia (MSRA) dataset and Weibo NER (Named Entity Recognition) corpus detailed parameters.

Dataset	Type	Train	Dev	Test
MSRA dataset	Sentences	46.4 k	-	4.4 k
	Characters	2169.9 k	-	172.6 k
	Entities	74.8 k	-	6.2 k
Weibo NER corpus	Sentences	1.4 k	0.27 k	0.27 k
	Characters	73.8 k	14.8 k	14.5 k
	Entities	1.89 k	0.42 k	0.39 k

4.2.1. MSRA Dataset

We use the MSRA dataset to test our proposed model CWPC_BiAtt. There are three types of named entities in the MSRA dataset: PER (Person), LOC (Location), ORG (Organization). [37] Compared marking method of IOBES (its other name BIOES is commonly used) is better than BIO. In the abbreviated word IOBES, B represents the current word is the beginning of a chunk, I represents the current word in a chunk, O represents the current word is not in any chunk, E represents the end of the current chunk, and S means that the current word is a chunk which has only one character. We take an example for illustration, and we use Chinese Pinyin instead of Chinese sentences. Sentence: zai hu jumin, keyi jingchang canguan shanghaibowuguan (Residents in Shanghai can often visit the Shanghai Museum). In China, hu is the shortened form of Shanghai and refers to location, therefore, according to the three types of named entities, we mark hu as hu-S-LOC. And for shanghaibowuguan (Shanghai Museum), an organization name, it was marked as shang-B-ORG, hai-I-ORG, bo-I-ORG, wu-I-ORG, guan-I-ORG. We relabeled the original MSRA dataset with the marking method of BIOES.

4.2.2. Weibo NER Corpus

We used Weibo NER corpus for comparative experiments. Weibo NER corpus was sorted by [1], 1890 pieces of information were from Sina Weibo from November 2013 to December 2014. This information is tagged as the DEFT ERE entity tagging guide. It mainly includes four main semantic types: person, organization, location, and geo-political entity. [1] annotated both name and nominal mentions. [25,38] revised Weibo NER corpus. After He and Sun gave major cleanup and revision of the annotations, Peng and Dredze revised the Weibo NER corpus. The Weibo NER corpus used in our experiments is the second revision of WeiboNER_2nd_conll by Peng and Dredze and for the dataset details see Table 2.

Table 2. Details of Weibo NER corpus.

	Name	**Nominal**	**Total**
Weibo NER corpus [1]	1276	705	1981
Weibo NER corpus by revised [38]	1321	1322	2643
Weibo NER corpus by revised [25]	1319	1320	2639
WeiboNER_2nd_conll			
Train set	1018	859	1877
Dev set	167	219	386
Test set	216	196	412
Total	1401	1274	2675

4.3. Settings

This section gives the detailed hyper-parameter configurations required for the experiment. We performed several tests on MSRA dataset and Weibo NER corpus and gave fine-tuning to the hyper-parameter as the experiment required. The character-embedding used in the experiment was from [39], who used Skip-Gram with Negative Sampling (SGNG) to train on the Baidu Encyclopedia corpus, and set the window size at 5, iteration 5 times, and the dimension $d_c = 300$.

For word-embedding, we use Wikipedia Chinese corpus on 1, August 2019: zhwiki-20190801-pages-articles- multistream.xml.bz2, the corpus size is 1.86 G. We also used SGNG for training, with window size of 10, iteration of 200, and the dimension $d_w = 400$. Due to the great number of iterations, we applied for the school's High Performance Computer (HPC) for 10-node calculation. See Table 3 for the specific configuration of HPC. The dimension of pos-embedding $d_p = 100$. Regarding the initialization of training weights, we used the deep learning framework Tensorflow to adopt random initialization for weights. The paper uses stochastic gradient descent (SGD) as the optimizer, the learning-rate = 0.01, and the dimension of the hidden layer of the BiLSTM $d_h = 600$. The number of Multi-head self-attention is 3, global self-attention 1, local self-attention 2, window for local self-attention 2w + 1, respectively set tow = 3, w = 5. We trained the CWPC_BiAtt model and each data set was iterated 25 times.

Table 3. The specific configuration of High Performance Computer.

Function	Node Name	Model	CPU
Login node	1–2	I620-G30	two Intel Xeon Gold 5118 CPU @ 2.30 Hz
Calculate node	1–110	CX50-G30	two Intel Xeon Gold 6132 CPU @ 2.60 GHz

4.4. Experiment and Analyze on MSRA Dataset

Observing the results obtained by our CWPC_BiAtt model on the MSRA dataset, what surprised us was that the CWPC_BiAtt model reached the new state-of-the-art performance, P = 93.71%, R = 92.29%, and F1 = 92.99%, respectively.

At the time, [22] working at Yahoo used two conditional probabilistic models, namely conditional random fields and maximum entropy models for Chinese NER performed on the MSRA dataset. In the same year, [23] from the State Key Laboratory for Novel Software Technology, Nanjing University proposed the Multi-Phase Model to conduct experiments on the MSRA dataset. The Multi-phase model is mainly divided into two steps: First, the text is segmented using the character-level CRF model; then, three word-level CRF models are applied to tag PER (person), LOC (location), and ORG (organization) of the segmentation results. Han et al. [40] from the Institute for Logic, Language, and Computation, University of Amsterdam proposed the Graph-based Semi-supervised Learning Model (GBSSL). They used unlabeled corpus (unlabeled corpus) to enhance the conditional random field learning model, and it is possible to improve the correctness of the tagging, even if the edge is a bit weak or unsatisfactory in the current experiment. Cao et al. [21] proposed the adversarial transfer learning framework to perform experiments on the MSRA dataset, which used the word boundary information provided by the Chinese word segmentation (CWS) to perform the Chinese NER task. This year, Nankai University and [26] from Microsoft Research Asia proposed the convolutional attention network to perform the Chinese NER task on the MSRA dataset, achieving quite good results.

From the results shown in Table 4, we found that [22,23] all used traditional machine learning methods to perform NER tasks, and their final F1 values were 86.20% and 86.51%. Although the latter has a slight increase, the increased range is only 0.31%. Recent study [21,26] and our model have a more significant increase than [22,23], the values are 4.44%,6.77%, 6.79% and 4.13%, 6.46%, 6.48% respectively, here the increase is more than 4%. Through the analysis, we found that [22,23] used traditional machine learning for NER (Named Entity Recognition) tasks, which required complex feature engineering, and that training results would not be so effective if they had fewer features. The neural networks [21,26] and our model used are not very dependent on complex features. In other areas, neural networks are far more accurate than traditional machine learning algorithms.

Table 4. Performance of previous state-of-the-art methods on MSRA dataset.

Models	P (%)	R (%)	F1 (%)
Conditional Probabilistic Models [22]	91.22	81.71	86.2
Multi-Phase Model [23]	88.94	84.2	86.51
Graph-based Semi-supervised [40]	90.62	77.84	83.74
Adversarial Transfer Learning [21]	91.73	89.58	90.64
CAN-NER [26]	93.53	92.42	92.97
CWPC_BiAtt (ours)	93.71	92.29	92.99

In addition, we also analyzed why the GBSSL-CRF [40] proposed was not effective in the F1 value. [40] put forward the Graph-based Semi-supervised. Semi-supervised learning requires establishing hypotheses to predict the relationship between samples and learning objectives. There are three common assumptions: (1) Smoothness Assumption; (2) Cluster Assumption; (3) Manifold Assumption. Semi-supervised learning is demanding for assumptions, and there is no guarantee that generalization performance will necessarily improve when using unlabeled samples, which can lead to performance degradation. But what's positive is that their innovative use of graph-based for Chinese NER missions is an innovative attempt. Our model got a little better results than [26]. There are three reasons: (1) Zhu et al. [26] only consider the character-embedding, while we take a more comprehensive consideration by using comprehensive-embedding. (2) Zhu et al. [26] did a local self-attention operation on the data firstly, and the second global self-attention operation was done on the basis of the first local-attention, so the second global self-attention's scope of action has been limited and significantly diminished. The global self-attention and local self-attention of our model are processed in parallel on the same basis, so there is no scope limited. (3) Zhu et al. [26] adopted Bidirectional Gated Recurrent Unit (BiGRU), which was proposed by [24]. Though gated recurrent unit (GRU) converges faster than LSTM [18], the GRU is not suitable for the more parameters and large data sets. Considering the large data set, BiLSTM is used in this article. And our model is much succinct than [26], so the model we proposed has an advantage.

It can be found from Table 4 that our proposed model is 0.13% lower in the recall than the recall in [26]. We found that in [26], the author uses character as the basis, so there is no problem of word segmentation. The comprehensive-embedding we use has word-embedding, because there is no natural interval between words in Chinese sentences. If a complete word is incorrectly divided into two words, the positive class will be divided into negative class to affect recall. But [26] used the convolution operation, there will be a certain loss of the original value, and our comprehensive-embedding has no lose, so this point will make up for the shortcomings of our model, which shows that comprehensive-embedding has completeness. The position part of Comprehensive-embedding is a main diagonal matrix, so the main calculation part is character-embedding and word-embedding. From the perspective of calculation volume and composition structure, Comprehensive-embedding has simplicity. In [26], the author uses BiGRU, and our CWPC_BiAtt uses BiLSTM. BiGRU performs poorly when there are many parameters, and the author uses character as the basis, so the parameters will be large. In order to verify this, we used different length sentences to replace BiLSTM in our model with BiGRU and tested it. It can be seen from Figure 4 that our CWPC_BiAtt model can stabilize the F-score value from 0.9298 to 0.9303 when testing sentences of different lengths. But when BiLSTM is replaced with BiGRU, as the sentence length increases, the F-score decreases, but the F-score can still stabilize above 0.9290. We enter into this analysis because comprehensive-embedding carries sufficient information to make up for the shortcomings of BiGRU in processing long sentences.

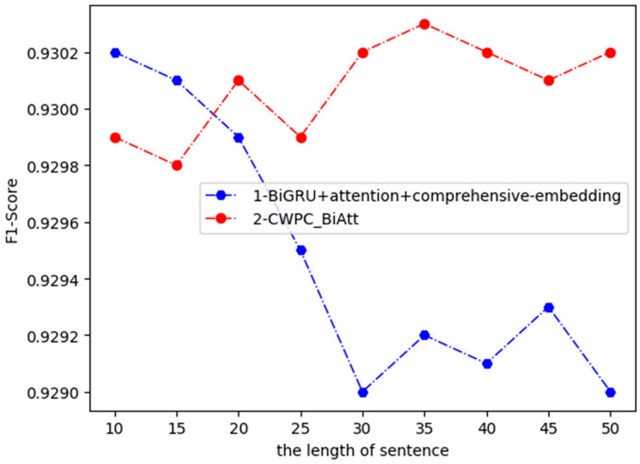

Figure 4. Experiments were performed on sentences of different lengths. As the sentence length increased, the results obtained by bidirectional gated recurrent unit (BiGRU) became worse, but its F-score was stable above 0.9090. Comprehensive-embedding carries sufficient information to make up for the shortcomings of BiGRU in processing long sentences. Our Character–Word–Position Combined bidirectional long short-term memory-attention (CWPC_BiAtt) model is very stable, and the F-score remains at 0.9298–0.9303.

Although our CWPC_BiAtt model obtained F-score = 92.99%, which is 0.22% higher than [26], it cannot be said that our method is better than [26]. Our CWPC_BiAtt model and [26] are from different directions. They are from the perspective of character convolution, and we are from the three integrated perspectives of character, word, and position. So, our model is also a new attempt.

Two local self-attentions use windows of different sizes. Considering that the length of the naming body corresponding to the PER, ORG, LOC tag is generally no more than 11, we fix a local self-attention 2k + 1 window and set k = 5. The second local self-attention is supplemented with details on a local self-attention. As Figure 5 shows, precision and recall get the maximum value when the second local self-attention window is set to 2k + 1, k = 3. As the windows get bigger and bigger, so does the precision and recall. More information is grabbed, so that gradually starts to overfitting.

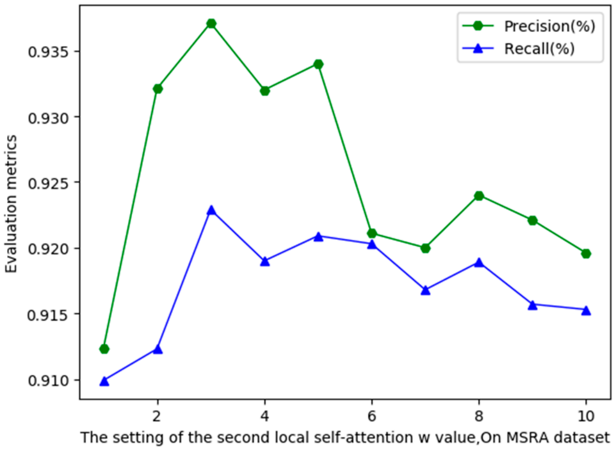

Figure 5. The second local self-attention window setting to 2w + 1, w = 3 can obtain the maximum value. When w > 3, the grasp of information increased, there gradually appears over-fitting, precision, and recall value also begin to decrease.

4.5. Experiment and Analyze on Weibo NER Corpus

Similarly, we experimented with CWPC_BiAtt model we proposed at Weibo NER corpus. The CWPC_BiAtt model achieved new state-of-the-art performance, with an overall F1 = 59.5%. See Table 5 for details. The results shown in Table 5.

Table 5. In first block, perimental results of previous models on Weibo NER corpus. In second block, the result of our CWPC_BiAtt model on Weibo NER corpus.

Models	Name			Nominal			Overall
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	F1 (%)
Joint (cp) without fine [1]	61.9	33.5	43.5	65.9	28.1	39.4	-
Joint (cp) with fine tuning [1]	58	35.6	44.1	63.8	29.5	40.4	-
Jointly Train LSTM + Emb [27]	63.3	39.2	48.4	58.6	37.4	45.7	-
F-Score II Driven (proposal) [25]	66.9	40.7	50.6	66.5	53.6	59.3	54.8
Unified Model (proposal) [38]	61.7	48.8	54.5	74.1	53.5	62.2	58.2
Adversarial Transfer Learning [21]	59.5	50	54.3	71.4	47.9	57.4	58.7
Lattice LSTM [41]	-	-	53	-	-	62.3	58.8
CAN-NER [26]	-	-	55.4	-	-	63	59.3
CWPC_BiAtt (Ours)	63.4	49.7	55.7	74.1	54.5	62.8	59.5

Peng et al. [1] marked the social media data set Weibo NER corpus, and proposed jointly training objective to perform Chinese NER task in Weibo NER corpus. They finally found that the experiment of character + position on CRF got very good results. On the basis of previous trial, [27] experimented on Weibo NER corpus using word segmentation with an LSTM-CRF. The final experimental results were greatly improved. The F1 values of Name and Nominal were respectively 4.3% and 5.3%

higher. Refs. [25,38] proposed a Unified Model, which can study out-of-domain corpora and in-domain unannotated text, and the F1 value of Name and Nominal was increased by 6.1% and 16.5%, respectively on the basis of the experiment by [27]. Zhang and Yang [41] proposed the Lattice LSTM model, which considers all possible word segmentation information and character information, and has also achieved very good results. Moreover, the maximum Nominal F1 score reached 63.0%, and F1 in our model Nominal obtained was 0.2% smaller than theirs. We have already covered the in Section 4.4 the model presented by [21,33], and here we omit the detailed description.

Peng and Dredze [27] made much progress based on their research in 2015. They considered word segmentation information and used LSTM to obtain historical information before accessing CRF. In spite of a significant improvement, the F1 values of Name and Nominal compared with others' like [21,25,38,41], and ours were quite different. Model learning of out-of-domain corpora and in-domain unannotated text proposed by [38] well adapted to the corpus of social media like Weibo NER corpus, which can be in any form. It was seen from the comparison between [21,26] and our model and [27] that the use of attention mechanism greatly improved the processing of NER task. We have conducted in-depth analysis and our model was slightly better than [21], because their model was relatively complex and they used word boundary information provided by Chinese word segmentation (CWS) to carry out the Chinese NER task. Any errors in word boundary information provided by CWS would directly affect the NER task. The comparison between [26] and our model has been analyzed in Section 4.4, which was not repeated here. Although the model proposed by [41] did not use the attention mechanism, their model has taken into account all possible participle and character information. For social media corpus like Weibo NER corpus, we should consider as many situations as possible to reduce the occurrence of incorrect identification. But, because of thinking about as many cases as possible, this not only increased the complexity of calculation, but also produced many alternative answers with only a correct one, and because the alternative answers were so close to the correct one, the machine would be confused.

As can be seen from Table 5, our total F-score = 59.5%. On Weibo NER corpus, our F-score is 0.2% higher than the previous best F-score. Our CWPC_BiAtt model is compared with [26], see Section 4.4. Although some of the precision and recall obtained by our CWPC_BiAtt model are not the best, our model is very stable, even on the more complex data set of Weibo NER corpus. F-score is a comprehensive measurement of precision and recall in order to resolve the contradiction between precision and recall. Comprehensive-embedding provides character, word, and position information. The combination of the three is as stable as a triangle. Character makes up for the shortcomings of word being misclassified. When attention-mechanism captures the dependencies between arbitrary characters, the order of the sequence is broken. But position-embedding makes up for the shortcomings of attention-mechanism. The position part of Comprehensive-embedding is a main diagonal matrix, so the main calculation part is character-embedding and word-embedding. From the perspective of calculation volume and composition structure, Comprehensive-embedding has simplicity. Therefore, comprehensive has two characteristics of completeness and simplicity, which is why we named it Comprehensive-embedding. From Tables 4 and 5, we can see that our precision, recall, and F-score are relatively stable, and they are in the top two positions. Our model has a stable advantage when dealing with different data sets. Looking at the entire model, it has three distinct characteristics: completeness, simplicity, and stability.

To sum up, we summarized the advantages of our model. Comprehensive-embedding could avoid all situations considered by [41]. Multi-head self-attention was used to conduct parallel processing of global self-attention and local self-attention on the same basis, which avoided the functional range constraints of local self-attention being conducted first and global self-attention later in [26]. Five experiments were conducted with different embedding methods in BiAtt-CRF. As can be seen from Figure 6, the CWPC-BiAtt we proposed obtained the state-of-the-art performance. Compared with chart-embedding or word-embedding only, the effect of using comprehensive-embedding was greatly improved. Also, experiment 2 was better than experiment

1 because the problem of OOV (out-of-vocabulary) could be overcome by character-embedding. As shown in experiment 4, character-embedding + word-embedding has greatly improved the results. Word-embedding considered the complete word information from the perspective of words, making up for character-embedding, which only considered the separated word information and thus, meaning was destroyed completely. As shown in experiments 3 and 5, pos-embedding has certain improvements on the experimental results. The experiment showed that comprehensive-embedding had great advantages and simple structure. Finally, the CWPC_BiAtt model we proposed got the best performance on Weibo NER corpus and excellently completed the task of Chinese NER.

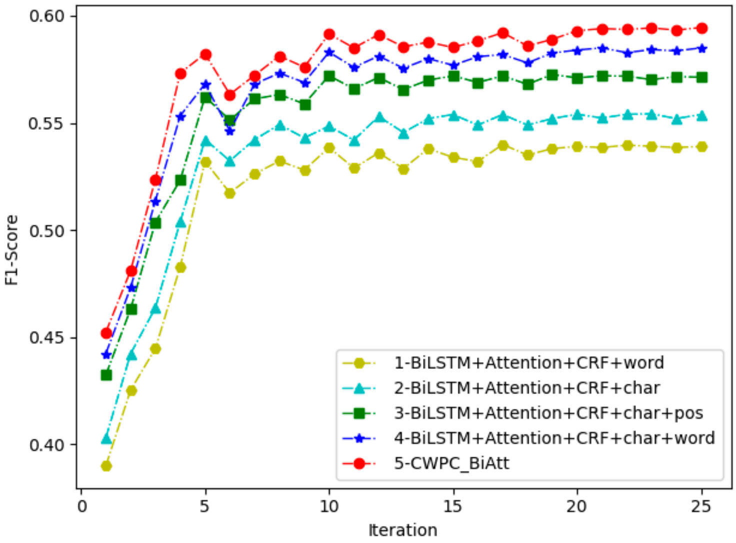

Figure 6. As shown in CWPC_BiAtt and other experiments, the results gained by using comprehensive-embedding are better than that selectively utilizing char-embedding, word-embedding, and pos-embedding. The F1-score of CWPC_BiAtt has increased by 1.2% than that of (char + word), in that comprehensive-embedding provides an index for sentences, which records the positional relation among embedding.

5. Conclusions

In this paper, the new model, CWPC_BiAtt, we put forward achieved state-of-the-art performance in the MSRA dataset and Weibo NER corpus. The comprehensive-embedding we proposed, which can take character, word, and position into account, has valid structure and can seize effective information. By utilizing BiLSTM, we can get information from the past and the future. We can also grab local and global information through Multi-head self-attention. Position-embedding in comprehensive-embedding can compensate for attention-mechanism to provide position information for the disordered sequence, which shows that comprehensive-embedding has completeness. The position part of Comprehensive-embedding is a main diagonal matrix, so the main calculation part is character-embedding and word-embedding. From the perspective of calculation volume and composition structure, Comprehensive-embedding has simplicity. Our model achieved the highest F-score on the MSRA dataset, and also performed well on the more complex Weibo NER Corpus, which was 0.2% higher than the previous best F-score. Looking at the entire model, our proposed CWPC_BiAtt has three distinct characteristics: completeness, simplicity, and stability. Experiments above show that the model we put forward can be qualified for Chinese NER.

In the future, we will do research into the possibility of applying the model to other NLP fields, such as sentiment analysis.

Author Contributions: Conceptualization, S.J.; Data curation, S.J.; Formal analysis, S.S. and Y.L.; Funding acquisition, S.J.; Investigation, S.J.; Methodology, S.S. and Y.L.; Project administration, S.J.; Resources, S.J.; Software,

S.J.; Supervision, S.S.; Validation, S.J.; Visualization, S.S.; Writing—original draft, S.S. and Y.L.; Writing—review & editing, S.S. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: The theory of this research work is support by the National Science Fund Project (61303097), the Shanghai Science Fund Project (17ZR1428400), the Second (2016) Shanghai Research Project for Private University (2016-SHNGE-08ZD) the support of this paper work.

Acknowledgments: Herein, I would like to extend my sincere gratitude to my supervisor, Professor Shardrom Johnson, for his patient guidance and detailed explanation during the process of completing the paper. I also want to express my heartfelt thanks to my junior fellow apprentice for their help. I also want to express my heartfelt thanks to my junior fellow apprentice for their help, and Haiqing Zhang, Xingwang Zhuang, Lingze Qin and Meixia Shan for their valuable advice.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Peng, N.; Dredze, M. Named entity recognition for Chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 548–554.
2. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 167–176.
3. Bunescu, R.C.; Mooney, R.J. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 724–731.
4. Miwa, M.; Bansal, M. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1105–1116.
5. Ratinov, L.; Roth, D.; Downey, D.; Anderson, M. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; Volume 1, pp. 1375–1384.
6. Gupta, N.; Singh, S.; Roth, D. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 2681–2690.
7. Yao, X.; Van Durme, B. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 956–966.
8. Isozaki, H.; Kazawa, H. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th International Conference on Computational Linguistics—Volume 1*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 1–7.
9. Kazama, J.; Makino, T.; Ohta, Y.; Tsujii, J. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain—Volume 3*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 1–8.
10. Ekbal, A.; Bandyopadhyay, S. Named entity recognition using support vector machine: A language independent approach. *Int. J. Electr. Comput. Eng.* **2010**, *4*, 589–604.
11. Zhou, G.; Su, J. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 473–480.
12. Florian, R.; Ittycheriah, A.; Jing, H.; Zhang, T. Named entity recognition through classifier combination. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003—Volume 4*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; pp. 168–171.
13. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2001; pp. 282–289.

14. Sobhana, N.; Mitra, P.; Ghosh, S.K. Conditional random field based named entity recognition in geological text. *Int. J. Comput. Appl.* **2010**, *1*, 119–125. [CrossRef]
15. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef] [PubMed]
16. Jordan, M.I. Serial order: A parallel distributed processing approach. In *Neural-Network Models of Cognition*; Advances in Psychology Series Volume 121; Donahoe, J.W., Dorsel, V.P., Eds.; North-Holland Publishing: Amsterdam, The Netherlands, 1997; pp. 471–495.
17. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
18. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
19. Chiu, J.P.C.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
21. Cao, P.; Chen, Y.; Liu, K.; Zhao, J.; Liu, S. Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 182–192.
22. Chen, A.; Peng, F.; Shan, R.; Sun, G. Chinese named entity recognition with conditional probabilistic models. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2006; pp. 173–176.
23. Zhou, J.; He, L.; Dai, X.; Chen, J. Chinese named entity recognition with a multi-phase model. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2006; pp. 213–216.
24. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1724–1734.
25. He, H.; Sun, X. F-score driven max margin neural network for named entity recognition in Chinese social media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 713–718.
26. Zhu, Y.; Wang, G.; Karlsson, B.F. CAN-NER: Convolutional attention network for Chinese named entity recognition. *arXiv* **2019**, arXiv:1904.02141.
27. Peng, N.; Dredze, M. Improving named entity recognition for Chinese social media with word segmentation representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 149–155.
28. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
29. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1532–1543.
30. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
31. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
32. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent Models of Visual Attention. In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 2204–2212.
33. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
34. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 207–212.

35. McCallum, A.; Freitag, D.; Pereira, F.C.N. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2000; pp. 591–598.
36. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **1967**, *13*, 260–269. [CrossRef]
37. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. 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*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 260–270.
38. He, H.; Sun, X. A unified model for cross-domain and semi-supervised named entity recognition in Chinese social media. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 3216–3222.
39. Li, S.; Zhao, Z.; Hu, R.; Li, W.; Liu, T.; Du, X. Analogical reasoning on Chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 138–143.
40. Han, A.L.-F.; Zeng, X.; Wong, D.F.; Chao, L.S. Chinese named entity recognition with graph-based semi-supervised learning model. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 15–20.
41. Zhang, Y.; Yang, J. Chinese NER using lattice LSTM. *arXiv* **2018**, arXiv:1805.02023.