applied sciences

MDPI

Article

A Local Information Perception Enhancement–Based Method for Chinese NER

Miao Zhang and Ling Lu *

College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China
* Correspondence: ll@cqut.edu.cn; Tel.: +86-139-8379-1161

Abstract: Integrating lexical information into Chinese character embedding is a valid method to figure out the Chinese named entity recognition (NER) issue. However, most existing methods focus only on the discovery of named entity boundaries, considering only the words matched by the Chinese characters. They ignore the association between Chinese characters and their left and right matching words. They ignore the local semantic information of the character's neighborhood, which is crucial for Chinese NER. The Chinese language incorporates a significant number of polysemous words, meaning that a single word can possess multiple meanings. Consequently, in the absence of sufficient contextual information, individuals may encounter difficulties in comprehending the intended meaning of a text, leading to the emergence of ambiguity. We consider how to handle the issue of entity ambiguity because of polysemous words in Chinese texts in different contexts more simply and effectively. We propose in this paper the use of graph attention networks to construct relatives among matching words and neighboring characters as well as matching words and adding left- and right-matching words directly using semantic information provided by the local lexicon. Moreover, this paper proposes a short-sequence convolutional neural network (SSCNN). It utilizes the generated shorter subsequence encoded with the sliding window module to enhance the perception of local information about the character. Compared with the widely used Chinese NER models, our approach achieves 1.18%, 0.29%, 0.18%, and 1.1% improvement on the four benchmark datasets Weibo, Resume, OntoNotes, and E-commerce, respectively, and proves the effectiveness of the model.

Keywords: Chinese named entity recognition; graph attention network; convolutional neural network; lexicon information

Citation: Zhang, M.; Lu, L. A Local Information Perception Enhancement–Based Method for Chinese NER. *Appl. Sci.* **2023**, *13*, 9948. https://doi.org/10.3390/app13179948

Academic Editor: Douglas O'Shaughnessy

Received: 14 August 2023
Revised: 25 August 2023
Accepted: 28 August 2023
Published: 3 September 2023

1. Introduction

From a given unprocessed text, the named entity recognition (NER) task seeks to identify and categorize related entities. Named entity recognition has an essential effect in subsequent natural language processing (NLP) projects. These projects include relation extraction [1], question-answering systems [2], and entity linking [3].

The NER aspect of Chinese usually uses character-level annotation strategies to identify named entities [4]. Several studies have shown that the character-based NER approach avoids errors in the subword stage [5,6]. However, sometimes the lexical boundary is the entity boundary; thus, the lack of boundary information provided by the lexicon may cause the wrong entity to be extracted. Take this one, for instance: "南京市长江大桥 (Nanjing Yangtze River Bridge)"; if there is no lexical knowledge, some wrong information, such as "南京市长 (Mayor of Nanjing)" and "江大桥 (Jiang Daqiao)", may be extracted. Therefore, recent research has focused on improving NER's performance by better integrating lexical information into characters.

To our knowledge, there exist two primary methodologies for integrating character and lexical information. The first is the dynamic framework method. It designs corresponding structural support for lexical typing, such as Lattice-LSTM [7], LR-CNN [8], and FLAT [9]. Lattice-LSTM extends the commonly used character-based long short-term

memory (LSTM) networks to encode character information in sentences while fusing potential word information. The LR-CNN model employs convolutional neural networks (CNNs) to encode both character attributes and probable word features. Additionally, attention mechanisms are utilized to effectively integrate the information from characters and words. However, both RNNs and CNNs have limitations in modeling long-range dependencies [10]. FLAT overcomes this limitation by designing an ingenious positional encoding to fuse the lattice structure at the top of the Transformer [10]. As a result, FLAT can interact immediately with all matching words for characters independent of long-range dependencies. Despite the research progress, the above methods still need to improve the specific structure of neural networks, thus limiting the broader application. Another approach is constructing adaptive embedding based on lexical information, i.e., embedding lexical knowledge in the encoding stage. WC-LSTM uses four encoding strategies to encode the Lattice-LSTM input statically [11]. WC-LSTM, although an adaptive embedding paradigm, suffers from information loss. To incorporate contextual information in the original vector of individual characters, Luo first filters the set of candidate entities for a given character and then constructs a character–entity relationship graph of characters and candidate entities [12]. The character representations in the character–entity relationship adjacency matrix are updated using graph attention networks (GAT). Finally, a character representation incorporating semantic information of contextual entities is obtained. To better utilize the lexical sources, SoftLexicon directly maps word characters to four positions, begin, middle, end, and single, and then uses a static weighting method to weight the word frequency magnitude in the lexical set [13]. SoftLexicon has been shown experimentally to effectively address the underutilization of low-speed inference and matching words, compensating for the shortcomings of the lattice-based model [14]. The unique feature of this approach is that it does not require the development of complex sequence modeling architectures. Therefore, it can be applied to other sequence annotation frameworks.

In addition, some studies have achieved good results without utilizing an external lexicon. Gu found that most types of entities have strong naming regularity. To effectively explore the internal compositional information of entities, a Regularity-Inspired reCOgnition Network (RICON) was designed [15]. The model utilizes a regularity-aware module to capture the internal regularity of each span. Then, a regularity-agnostic module is employed to mitigate the excessive focus on span regularity. RICON achieves the state-of-the-art performance of the year on the four datasets. Liu utilized BERT pretrained language models to replace traditional static word embeddings [16,17]. Employing a context-dependent dynamic generation of semantic vectors improved the representation of word embeddings. It could extract entities more accurately and efficiently than traditional named entity recognition algorithms. It also achieved good results in the named entity recognition task within history and culture. To reduce the dependence on data annotation, Chen developed a new semisupervised model called MAUIL [18]. Compared with other models, MAUIL cleverly integrates multiple levels of attribute embedding, such as character-level and word-level features. This approach enhances the high-level semantic features in text and dramatically improves the reliability of artificial intelligence programs, such as named entity recognition. In addition, Li proposed a new method called W2NER, which can handle three types of NER tasks: planar entities, overlapping entities, and discontinuous entities in a unified manner [19]. The NER task is constructively transformed into predictive word–word relation classification. The model structure effectively simulates the adjacency relationships between entity words using next neighbor word (NNW) and trailing head word-* (THW-*) relations. W2NER has driven unified NER to achieve the most advanced performance. These proposed new frameworks bring new ideas to Chinese NER.

According to our findings, most existing studies focus on entity discovery methods. These methods focus more on detecting entity boundaries and only consider words in the thesaurus that match entity characters. However, they ignore the knowledge of the interaction between entity characters and their neighboring matching characters. Fusing lexical information improves the representation for kanji, and this is necessary for Chinese NER.

However, information on the entity boundary region is essential for entity detection, and existing lexicon-based methods pay less attention to this region. Our proposed boundary region is the adjacent region's front and back zones of the entity boundary, as shown in Figure 1. It is a boundary region of size K, which we call Zone-K.

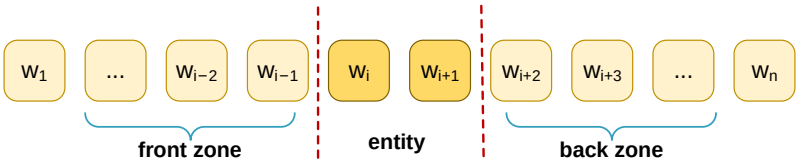

Figure 1. The front zone and back zone of the entity.

On the one hand, the lexical semantics of Zone-K helps to improve the understanding of entities and thus to determine their categories. For example, Figure 2 shows that although "高雄(Kaohsiung)" can be detected in both sentences 1 and 2, it is a challenge to determine its category as "PER" in sentence 1 and "LOC" in sentence 2. This is because there are many polysemous words in Chinese. Thus, even if the boundary of an entity can be detected correctly, determining its category is still a challenge. In this case, we propose considering the semantics of Zone-K characters and their lexical matching words. For example, in sentence 1, the category of "高雄 (Kaohsiung)" can be identified as "PER" by "演员 (performer)" and "饰演 (play)". In contrast, in sentence 2, the category of "高雄 (Kaohsiung)" can be identified as "LOC" by "在(in)" and "住(live)".

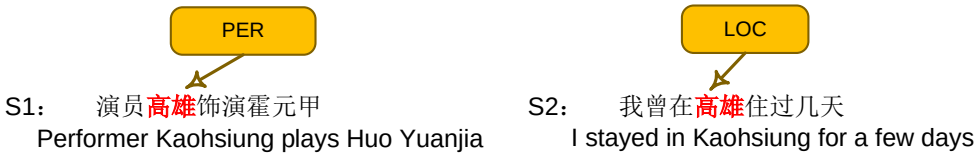

Figure 2. Examples of ambiguous entity words.

Second, there is a significant semantic change between the characters in Zone-K and the boundary characters of the entity. For the example, in Figure 3, for the sentence "我曾在高雄住过几天 (I stayed in Kaohsiung for a few days)", since the two character sequences "在(in)-高 (gao)" and "雄 (xiong)-住(live)" are small probability co-occurrence sequences, we consider "在(in)-高 (gao)" and "雄(xiong)- 住 (live)" as two semantic violators, which means that the semantic distance between "在 (in)" and "高 (gao)" as well as "雄 (xiong)" and "住 (live)" is quite far. Therefore, we consider Zone-K as a semantic mutation zone, which is similar to an image's contour and reflects the text's local feature discontinuity. Semantic changes in Zone-K can help determine the entity's boundary "高雄 (Kaohsiung)".

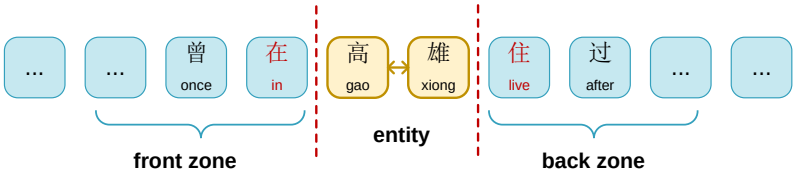

Figure 3. Examples of front and back zones of an entity.

In summary, we propose to use the Zone-K information in two ways.

One is to fuse the lexical knowledge of Zone-K to help determine the category of entities. For this purpose, we propose to use graph attention networks to catch connections among characters with their neighboring character-matching words. For example, based on semantics of the adjacent contextual match "饰演 (play)" for "雄 (xiong)", "高雄 (Kaohsiung)" can be inferentially tagged as "PER".

Second, the semantic transformation of Zone-K is introduced to help determine the boundaries. This is similar to contour detection in images, where local semantic fusion

can detect semantic change boundaries. For this purpose, we introduce CNN, which uses sliding windows to fuse short-sequence information of the text to perceive local sequence features of the text. Furthermore, Chiu and Nichols proposed to combine LSTM and CNN networks to learn character–word level information for English NERs [20]. This inspired us to provide a model that combines short-sequence CNN and LSTM coding. The local features of the text are extracted using short-sequence CNN, resulting in a local contextual representation. The global context representation is obtained by implicitly encoding the character sequence using LSTM, and then the local and global representations are used together for NER.

Our approach bridges the gap in the following aspects compared with previous approaches. First, the model adopts feature representation, context encoder, and tag decoder architecture, which has the property of migrating to other networks. Moreover, it can be combined with BERT [17]. Second, we introduce lexical information from the character representation layer based on the SoftLexicon method, which is simple and direct. The graph neural network is used to directly capture the lexical semantic information of entity neighborhoods without the need to construct dependency parse trees with the help of external NLP tools. It effectively avoids error propagation issues, makes up for information loss, and improves the performance of Chinese NER. Finally, the sequence coding layer of the model effectively balances the acquisition of local and global information and enhances the recognition of entity boundaries. Adding GAT simply and effectively improves the prediction accuracy of entity types.

The present study can be succinctly outlined by considering the following facets:

1. We constructed a Chinese NER method with enhanced local information perception. The method directly utilizes local lexical information to capture the semantic relationship between entity characters and matching lexical entries through graph attention networks. There is no need to build dependency parse trees with the help of external NLP tools. This avoids the problem of error propagation caused by this process, thus effectively improving NER performance.
2. We used a modified short-sequence CNN to fuse local features and achieve encoding of shorter subsequence features by an additional sliding window module. Then we combined it with LSTM to obtain a global representation of the sequence. It compensates for the shortcomings of existing sequence encoders in extracting local and global features.
3. The experiment achieved advanced results on standard Chinese NER datasets in four domains. Moreover, entity-type prediction accuracy improved, indicating that the proposed local information-aware approach is interpretable.

2. Related Work

2.1. Chinese NER

The Chinese language exhibits a distinctive characteristic wherein the demarcation between words within Chinese texts lacks clarity. Further, Chinese has an intricate grammatical structure and numerous synonyms. Due to this rationale, Chinese named entity recognition typically employs character-level annotation strategies [21]. Furthermore, lexical data can provide more boundary information for character-based learning. As a result, some works propose incorporating word information into character sequences to exploit each character's lexical information fully [11,22]. The model's performance is improved by connecting lexical knowledge with relevant characters in the sentence, enabling additional lexical features to improve the extraction of semantic features from sentences. Ma proposed a Chinese NER approach based on SoftLexicon encoding of word information. It encodes character and lexical information into a joint representation of the model's input layer [13]. It first uses the lexicon to find a word for each character corresponding to the four position types "BMES" [23]. B (begin) denotes the beginning position of a word, M (middle) denotes the middle position of a word, E (end) denotes the end position of a word, and S (single) denotes a single word. As shown in Figure 4, "海 (sea)" in "海南海口

(Hainan Haikou)" can match "海口 (Haikou)" in the "B" position, "南海口 (South Seaport)" in the "M" position, "南海 (South China Sea)" in the "E" position, and "海 (sea)" in the "S" position. If a related word does not exist, it is replaced by none, as shown in Figure 5. The word set embeddings are computed using the frequency of word occurrences. Finally, these character embeddings containing lexical information are fed into the sequence encoder, and then the label results are predicted by the conditional random fields (CRF) module [24].

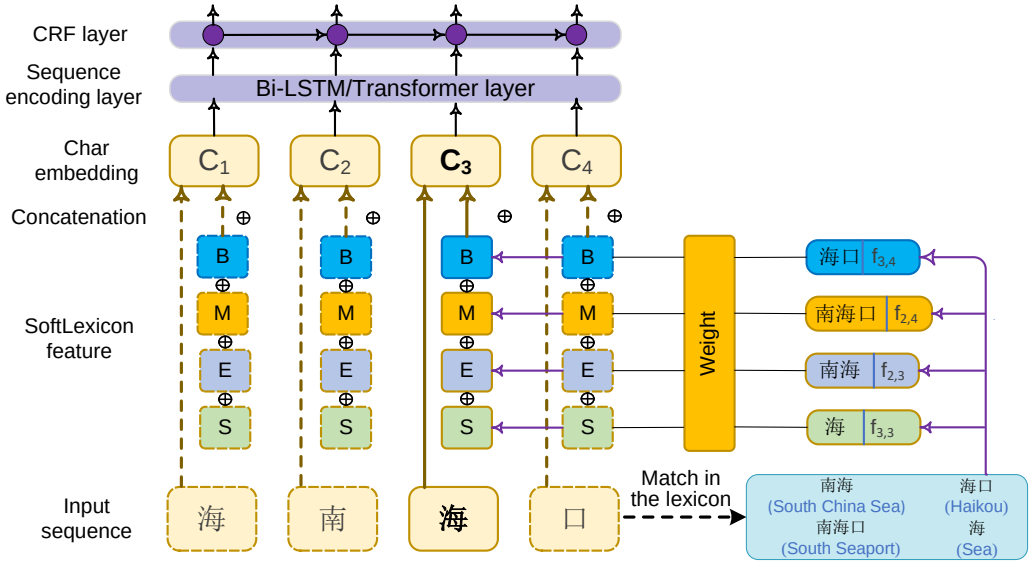

Figure 4. Overall architecture of the SoftLexicon model. By constructing the SoftLexicon features, the model adds the lexicon matches to the begin, middle, and end of each character, as well as single positions. Then it inputs these enhanced character embeddings into the sequence encoding layer and the conditional random field (CRF) layer to obtain the final prediction results.

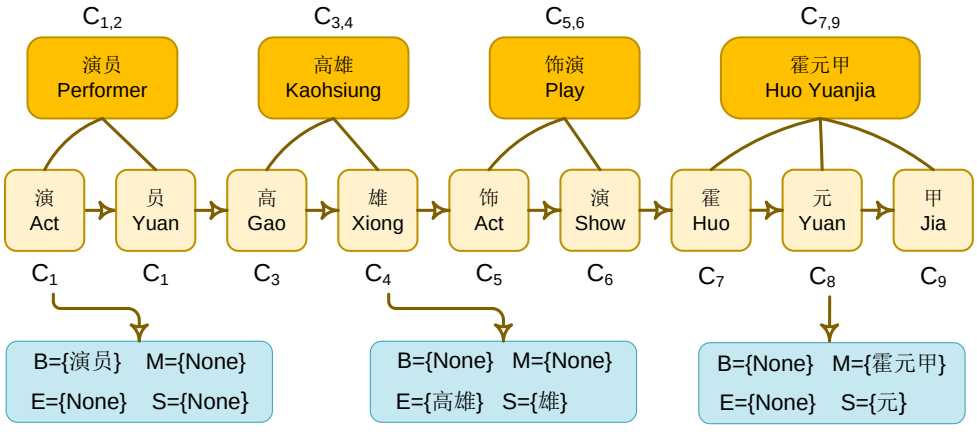

Figure 5. SoftLexicon method.

2.2. Graph Attention Network

Recently, the successful application of lexical information and graph neural networks has demonstrated the effectiveness of graph neural network models in enhancing the performance and sophistication of NER. Gui developed a graph network that utilizes lexical information and constructed the Chinese NER project to classify nodes in a graph problem [25]. The advantage of graph attention networks is the application of a multihead attention mechanism. Hence, they can summarize the features of the graph by allocating respective weights to adjacent nodes or correspondence edges [26]. Wang found that the majority existing methodologies relied on static weighting methods in calculating character-

word set embeddings. This leads to an inaccurate utilization of lexical information, which seriously affects the performance of NER. For this reason, Wang proposes a polymorphic graph attention network (PGAT) [27]. It constructs a graph for each of the four lexical sets of "BMES" [23]. It dynamically captures the relationship between characters and matching words from multiple dimensions, thus more fully utilizing lexical information. Furthermore, by utilizing large-scale grammatical information, neural network models can achieve improved performance [28,29]. However, these methods typically rely on external NLP tools to build dependency parsing trees, which can lead to error propagation issues [7,29,30], since word embeddings already available in lexicon can provide lexical semantic information. In addition, the graph attention mechanism can dynamically adjust the importance of word meanings based on contextual information, thereby achieving word sense disambiguation [26]. On the other hand, GAT establishing a joint representation of the character–word method helps us incorporate the semantic relationship between characters and words in their neighborhood into the model. For this reason, we constructed a graph attention network for Chinese NER to capture local feature information and improve the recognition rate of named entities.

3. Approach

The overarching model framework of our method is shown in Figure 6. The primary composition of this system consists of four distinct network modules. Initially, the encoder layer is employed to acquire contextual information pertaining to sentences and to represent the semantic information associated with the lexicon. Then, it uses SoftLexicon to fuse the lexicon information and use short-sequence CNN and Bi-LSTM to obtain the sequence's local and global hidden details, respectively. Semantic relations between characters and their adjacent characters' matching words were captured using GAT [26]. Finally, the feature representation after fusing the lexicon is summed with the output of the sequence encoder, and the tags are decoded using the standard CRF model [31].

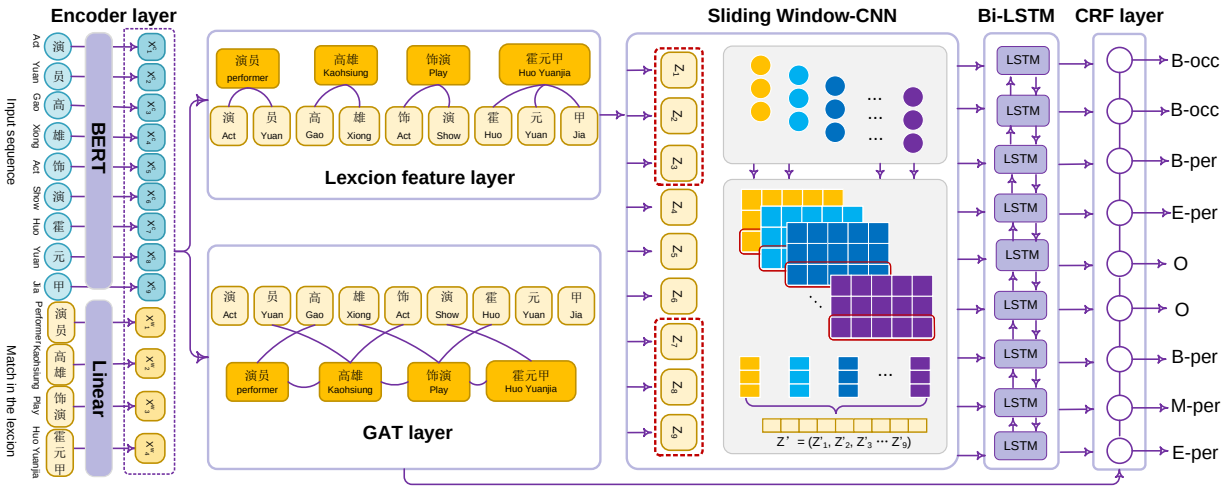

Figure 6. The overall proposed NER architecture. After the encoder layer, first, the sequence hidden state is obtained by short-sequence CNN combined with a Bi-LSTM structure after the fusion of the matching word information using the lexicon. Then the relationship between characters and their adjacent characters' matching words is captured using GAT, and finally, the two outputs are summed to predict the sequence labels using a CRF layer.

3.1. Encoder Layer

The character-level-based approach to Chinese NER treats sentences as sequences of multiple independent Chinese characters: $s = \{c_1, c_2, \ldots, c_n\}$. Each character vector in the sentence is represented as

$$x_i^c = e^c(c_i) \tag{1}$$

where e^c is a character vector's lookup table. We chose BERT as the input to the model, which one of the most advanced pretraining models widely used for natural language tasks [17,32]. The context representation of each character x is obtained after the calculation of BERT. The NER model based entirely on characters has the problem that word information cannot be used to introduce lexical information into the character representation. Regarding the given input sequence $s = \{c_1, c_2, \ldots, c_n\}$, $w_{i,j}$ represents each lexicon matched by its subsequence $\{c_i, c_{i+1}, \ldots, c_j\}$, with matching words defined as $w = \{w_1, w_2, \ldots, w_m\}$. Use dense vectors to represent each word:

$$x_i^w = e^w(w_i) \tag{2}$$

where e^w is a word vector's lookup table.

3.1.1. Lexicon Feature Layer

The lexicon feature uses the SoftLexicon method to fuse matching words with characters. According to the input sentence $s = \{c_1, c_2, \ldots, c_n\}$, SoftLexicon generates matching word sets for characters using the word set labeled by the four positional labels "BMES" [23]. $B(c_i)$ denotes the set of matching words starting with c_i on L. Similarly, $M(c_i)$ and $E(c_i)$ represent the set of words matching the middle and end of c_i, respectively. The set of words represented by a single character c_i only is denoted as $S(c_i)$.

$$B(c_i) = \{w_{i,k}, \forall w_{i,k} \in L, i < k \leq n\} \tag{3}$$

$$M(c_i) = \{w_{j,k}, \forall w_{j,k} \in L, 1 \leq j \leq i < k \leq n\} \tag{4}$$

$$E(c_i) = \{w_{j,k}, \forall w_{j,i} \in L, 1 \leq j < i\} \tag{5}$$

$$S(c_i) = \{c_i, \exists c_i \in L\} \tag{6}$$

where L denotes the lexicon, $w_{i,k}$ denotes a subsequence from character c_i to character c_j. The word set vectors are then aligned and connected to the character vectors. Then each character can be represented as

$$v^S(S) = \frac{4}{Z} \sum_{w \in S} z(w) e^w(w), Z = \sum_{w \in B \cup M \cup E \cup S} z(w) \tag{7}$$

$$e^S(c_i) = \left[v^S(B(c_i)); v^S(M(c_i)); v^S(E(c_i)); v^S(S(c_i)) \right] \tag{8}$$

$$Y(c_i) = \left[e^w(c_i); e^S(c_i) \right] \tag{9}$$

where S denotes the set of words, and $z(w)$ denotes the frequency of occurrence of the lexicon word w in the dataset.

3.1.2. GAT Layer

We first build the required graph. The GAT layer analyzes the correlations among matching words and their adjacent characters and adjacent matching words. Thus, each character and its matching word serve as the graph's vertex, represented as

$G_h = [x_1^c, x_2^c, \ldots x_n^c, x_1^w, x_2^w, \ldots, x_m^w]$, where n and m denote the count of characters and matching words, respectively. The adjacency matrix is shown in Figure 7. If matching word i or character v is associated with adjacent pre- and postmatching words of character j, the correspondence of (i, j) or (v, j) of the corresponding position of the adjacency matrix M is then filled with 1. Similarly, if matching word i and another word k are adjacent pre- and postmatching words, sign "$M_{ik} = 1$". Since this adjacency matrix is symmetric, only its upper triangular region is stored to achieve compressed storage.

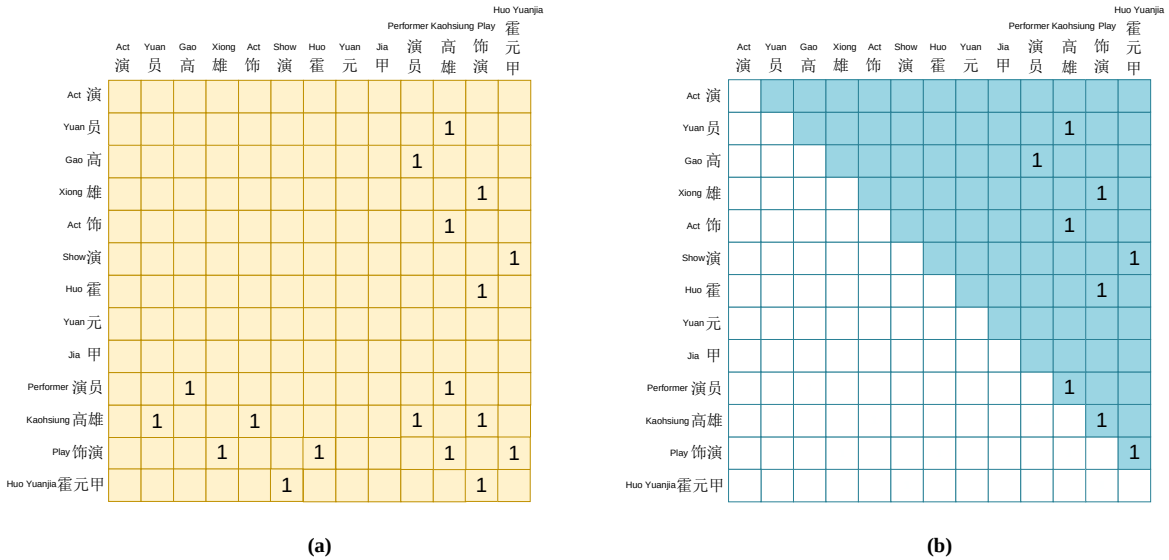

Figure 7. Optimal storage is used to hold the adjacency matrix, which depicts the connections between graph nodes: (**a**) original adjacency matrix and (**b**) optimized adjacency matrix.

We take the node features $GF = \{f_1, f_2, \ldots, f_N\}$ and the adjacency matrix M feed into the GAT layer. $f_i \in \mathbb{R}^F$, $M \in \mathbb{R}^{N \times N}$, where F and N are the characteristic dimension and number of nodes, respectively. Then a new set of node characteristics can be obtained, that is,

$$GF' = \left\{ f_1', f_2', \ldots, f_N' \right\} \tag{10}$$

The following output feature representation is obtained using K-independent attention mechanisms:

$$f_i' = \Big\Vert_{K=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k W^k f_j \right) \tag{11}$$

$$\alpha_{ij}^k = \frac{\exp\left(Leaky\mathrm{ReLU}\left(a^T \left[W^k f_i || W^K f_j \right] \right) \right)}{\sum_{k \in N_i} \exp\left(Leaky\mathrm{ReLU}\left(a^T \left[W^k f_i || W^K f_k \right] \right) \right)} \tag{12}$$

where $||$ indicates a connection operation, σ is the nonlinear activation function, N_i in the network is the node next to node i, α_{ij}^k is an attention factor, and $W^k \in \mathbb{R}^{F' \times F}$, $a \in \mathbb{R}^{2F'}$ is a single-layer feedforward neural network. At this layer, the output dimension of f_i is KF'. The averaging operation obtains the final output feature F' in the last layer.

Through the computations on the i-th vertex and its associated vertices, we can achieve the terminal representation for the i-th vertex. The corresponding coefficients calculated according to the attention mechanism are

$$f_i^{final} = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N_i} \alpha_{ij}^k W^k f_j \right) \tag{13}$$

GAT's inputs are the vertex set G_h and the adjacency matrix M, then the characteristics of the node can be obtained:

$$G = GAT(G_h, M), G \in \mathbb{R}^{F' \times (n+m)} \tag{14}$$

$$G' = G[:, 0, n] \tag{15}$$

Here, we retain the top n columns of these matrices since only the character representation serves to decode the labels. It is shown in Equation (15). Here is a new sentence representation that integrates semantic knowledge between characters and their adjacent character-matching words.

3.2. Sequence Encoding Layer

3.2.1. Short-Sequence CNN with Sliding Window

Chiu and Nichols considered that by relying solely on word embeddings, it is unable to utilize explicit character-level features, such as prefixes and suffixes [20]. They, therefore, proposed a hybrid model of bidirectional LSTMs and CNNs that learns both character- and word-level features. This approach is of particular interest because it both inputs word-level embeddings and handles the characters of each word. This approach inspired us to use LSTMs to process entire sequences and extract features simultaneously for shorter subsequences. To capture character-based nuances in short sequences, we designed a sliding window module that sequentially divides the sentence into multiple shorter subsequences. At the same time, CNN is good at capturing local features (such as n-grams or short sequences). Combined with LSTM's ability to solve the long-term dependency problem, our sequence encoding layer can effectively balance the need for global and local sequence information.

Convolutional neural networks have a natural advantage for local feature extraction due to their sliding convolutional computation. The detection of entity boundaries can be considered as detecting local semantic mutations in text, similar to image contour detection. Thus, we propose the short-sequence CNN module (SSCNN) to enhance local information extraction. As illustrated in Figure 8, we construct a sliding window of front zone, zone, and back zone to generate a subsequence $\{z_{i-1}, z_i, z_{i+1}\}$ of length 3 for each character and the left and right adjacent characters, and then apply it to a convolutional neural network. The antecedent and consequent relationships between each character in this subsequence and its neighboring characters will contain z_i-rich local contextual information. This operation is characterized by the fact that the local features of each character can be highlighted more clearly without being corrupted by long-distance information.

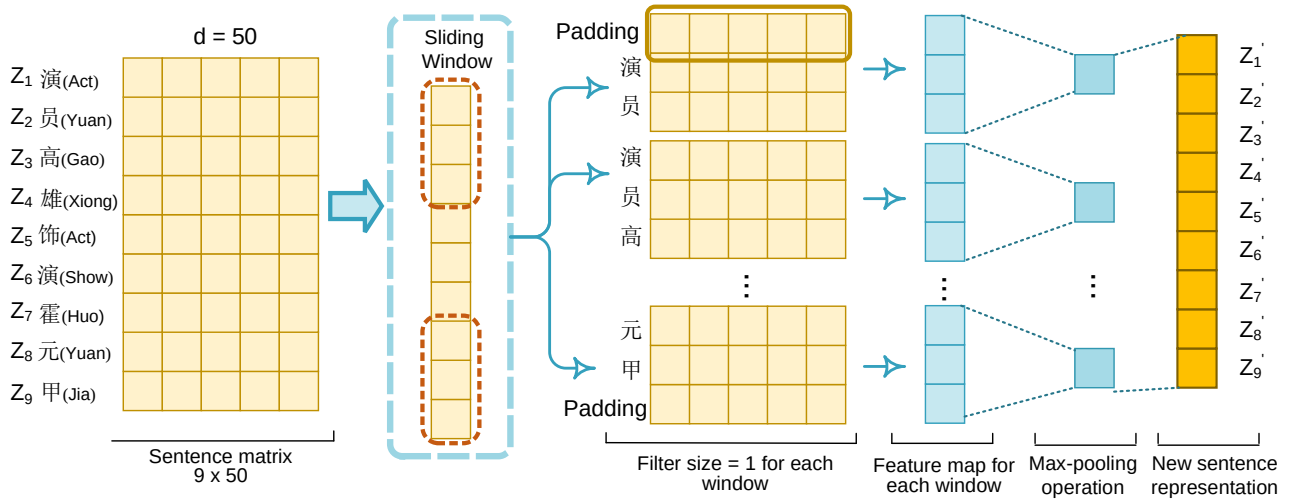

Figure 8. Short-sequence CNN with sliding window.

When the sentence length is n, assume that $z_i \in \mathbb{R}^d$ is the corresponding d-dimensional vector of the i-th character. Furthermore, some sentences need to add padded operations. The convolution operation involves the filter $w \in \mathbb{R}^{hd}$. Then filters are applied to a single character (h = 1) in each subsequence $\{z_{i-1}, z_i, z_{i+1}\}$ generated by the sliding window (front zone, zone, back zone) to generate new features. For instance, feature x_i is generated from the window of character z_i, by $x_i = f(w \cdot z_i + b)$, where f is the hyperbolic function and $b \in \mathbb{R}$ is the bias. Use this filter operation for all the windows within the sentence. Each sub-sequence $\{z_{i-1}, z_i, z_{i+1}\}$ generates a feature map of $x = \{x_{i-1}, x_i, x_{i+1}\}$ and $x \in \mathbb{R}^{i-1,i+1}$. Then, we used a maximum pooling operation on each feature map to capture the essential features [33]. They finally obtained a brand-new sequence $Z^{'} = \left\{Z_1^{'}, Z_2^{'}, \ldots, Z_n^{'}\right\}$ containing local feature information.

3.2.2. LSTM for Global Feature Extraction

After obtaining the character sequence's local feature information, we feed it into a single-layer Bi-LSTM to obtain global contextual information [34]. Forward LSTM is defined as follows:

$$I_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{16}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{17}$$

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{18}$$

$$\widetilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{19}$$

$$C_t = f_t * C_{t-1} + I_t * \widetilde{C}_t \tag{20}$$

$$h_t = O_t * \tanh(C_t) \tag{21}$$

where I_t is input gates, f_t is forget gates, and O_t is output gates. W is each weight matrix, h_t is the hidden state at step t, b is the bias to be applied, and σ is the sigmoid function. In forward LSTM, the input sequence is processed from left to right, whereas in reverse LSTM, it is processed from right to left, and connecting the LSTMs in both directions is represented as the output of the final Bi-LSTM.

3.3. Conditional Random Field (CRF) Layer

At the end of the model, we used the sequential CRF layer to make label inferences on a whole sentence [24].

$$p(y|s;\theta) = \frac{\prod_{t=1}^{n} \phi_t(y_{t-1}, y_t|s)}{\sum_{y' \in \gamma_s} \prod_{t=1}^{n} \phi\left(y'_{t-1}, y'_t|s\right)} \tag{22}$$

where y_s denotes all possible tag sequences s; then

$$\phi_t = \left(y', y|s\right) = \exp\left(\omega_{y',y}^{T} h_t + b_{y',y}\right) \tag{23}$$

where θ denotes the model parameters. $\omega_{y',y}$ and $b_{y',y}$ are each pair of label (y', y) corresponding to the network training parameters. In the final label prediction stage, the Viterbi algorithm filters the labels with the maximum probability as possible labels [35]. The following equation shows:

$$y^* = yp(y|s;\theta) \tag{24}$$

3.4. Implementation Details

The size of the model's character embedding and lexicon embedding is 50. A 2-layer GAT network is employed, with 3 attentions. A dropout rate of 0.5 is used to mitigate model overfitting. CNN and LSTM layers have 1 layer, with a dropout rate of 0.1 for CNN and 0.5 for LSTM. Additionally, ensure that the shortest subsequence provides locally localized information in Zone-K while being noise-free. The sliding window size consists of the character and its left and right neighboring characters; i.e., window size is taken as 3 and kernel size is 1. The model uses an Adamax optimization network, and the decay rate is set to 0.5. Additional specific training details can be found in the training section in Section 4 of the paper.

4. Experimental Results and Comparison

4.1. Experimental Settings

4.1.1. Datasets

Four commonly used Chinese datasets served as the basis for our experiments: Weibo [22], Resume [7], OntoNotes [36], and E-commerce [37]. Weibo is from Sina Weibo, and Resume is from Sina Finance. Weibo is labeled with four entities: personal names, places, organizations, and geopolitics, and contains a certain amount of noisy data; Resume is labeled with eight entities, such as educational institution, occupation, and title, and datasets have a limited number of entities but a high range of types. OntoNotes is a dataset derived from newswires and broadcasts and contains four named entity categories: personal names, places, organizations, and geopolitics. The E-commerce dataset is a dataset from the manually labeled e-commerce domain and includes both brand and product, two types of entities. Table 1 shows the details of these datasets.

Table 1. Statistics on the benchmarking dataset for the experiments.

Dataset	Type	Train	Dev	Test
E-commerce	Char	119.1 K	14.9 K	14.7 K
Weibo	Char	73.8 K	14.5 K	14.8 K
OntoNotes	Char	491.9 K	200.5 K	208.1 K
Resume	Char	124.1 K	13.9 K	15.1 K

4.1.2. Baseline Methods

Aiming to measure and analyze the proposed method, we compared its performance on various datasets with mainstream models of recent years.

- Lattice-LSTM [7] is a modified LSTM structure that can take as model input the characters in a sentence along with the lattice of all its potential matching words.
- LR-CNN [8] is based on a CNN model combined with the rethinking mechanism and using attention mechanism to the integration of character–word feature information.
- PLTE [38] is an expansion of the Transformer model that batch parallelizes the processing of characters and their matching words.
- LGN [25] is using graph neural networks to handle named entity recognition as a graph node categorization operation.
- Multi-Graph [37] is a graph neural network–based approach to named entity recognition combined with a multigraph that can automatically learn the features of gazetteers.
- FLAT-BERT [9] is a model based on the lattice structure using Transformer's location encoding to capture lexical information.
- SoftLexicon-BERT [13] is a method of incorporating lexical information directly into character representation at the character representation layer.
- PGAT-BERT [27] is a polymorphic graph attention network that can capture fine-grained character and matching word relationships more efficiently and dynamically.

4.2. Training

We applied the same pretrained character and lexical embedding as in Lattice-LSTM [7]. The character embedding size and lexicon embedding size are both 50. The multiple heads used in the graph network are 3. We trained our network using the Adamax optimization for all datasets. The decay rate is 0.05. All experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU. Weibo has an optimal learning rate of 0.05, while OntoNotes, Resume, and E-commerce are set to 0.03. Additionally, we set the batch size to 1 for the E-commerce dataset and 8 for the Weibo, Resume, and OntoNotes datasets. OntoNotes' hidden size is set to 300, while Weibo, Resume, and E-commerce are set to 200. These important hyperparameters can be seen in Table 2. Besides, to avoid overfitting the model, we used a parameter setting of a dropout rate of 0.5 in the character embedding layer, word embedding layer, and sequence encoding layer. See Appendix A Table A1 for more hyperparameter settings. Our model uses precision rate (P), recall rate (R), and F1 score as the evaluation metrics of performance.

Table 2. The hyperparameters for best models that we have experimented on the given datasets.

Hyper	Weibo	Resume	OntoNotes	E-Commerce
Batch Size	8	8	8	1
Decay Rate	0.05	0.05	0.05	0.05
Learning Rate	0.005	0.003	0.003	0.003
Hidden Size	200	200	300	200

4.3. Overall Results

We compare the recently widely used lexicon-based character enhancement model and the Chinese NER method, applying graph neural networks as baseline methods with our proposed methods. Comparisons on the Weibo, Resume, and OntoNotes datasets are shown in Table 3.

Table 3. F1 score statistics on the Weibo, Resume, and OntoNotes dataset.

Models	Weibo	Resume	OntoNotes
Lattice-LSTM [7]	58.79	94.46	73.88
LR-CNN [8]	59.92	95.11	74.45
PLTE [38]	59.92	95.40	74.60
LGN [25]	60.15	95.41	74.85
Multi-Graph [37]	59.50	-	76.00
BERT-LSTM-CRF	67.33	95.51	81.82
FLAT-BERT [9]	68.55	95.86	81.82
SoftLexicon-BERT [13]	70.50	96.11	82.81
PGAT-BERT [27]	70.63	96.53	81.87
ours	71.81	96.40	82.99

Table 3 shows the experimental results of NER obtained by our model and other baseline models on the Weibo dataset. We compare our model with three traditional lexicon-based NER models, Lattice-LSTM [7], LR-CNN [8], and PLTE [38], and two graph neural network–based models, LGN [25] and Multi-Graph [37]. In addition, three recent mainstream lexicon-based models are also compared with our model. We can observe that the SoftLexicon-BERT [13] model achieves the best F1 score of 70.63% among all baseline models. On the other hand, our model improves the total F1 score to 71.81%, an increase of 1.18%.

The results of the Resume dataset are shown in Table 3. The F1 score of Lattice-LSTM [7], which released the dataset, is 94.46%; the F1 score of LR-CNN [8] is 95.11%; and the F1 score of the improved PLTE [38] is 95.40%. In addition, PGAT-BERT [27] uses the polymorphic graph attention network based on SoftLexicon [13], which obtained the

highest F1 score of 96.53% among all baseline models. Our model obtained the second-highest F1 score of 96.40%.

Table 3 also displays the experimental results of various models on the OntoNotes dataset. BERT-LSTM-CRF and FLAT-BERT [9] both achieved an F1 score of 81.82%. The most recent polymorphic graph–based attention network, PGAT-BERT [27], achieves an F1 score of 81.87%. SoftLexicon-BERT [13] has the highest F1 score among the baseline models at 82.81%. We can see that our model obtained the highest F1 score of 82.99%.

We also compared the E-commerce dataset with Multi-Graph [37], PGAT-BERT [27], and Bi-LSTM-CRF as the baseline. In addition, the Bi-LSTM-CRF methods were compared by adding three ground name table features, with the separate addition of N-gram features, position-independent entity type (PIET features), and position-dependent entity type (PDET features). The results are shown in Table 4.

Table 4. Precision rate, recall rate, and F1 score statistics on the E-commerce dataset.

Models	P	R	F1
Bi-LSTM-CRF	71.1	76.1	73.6
(+N-gram features)	71.2	75.9	73.5
(+PIET features)	71.7	75.8	73.7
(+PDET features)	72.6	75.1	73.8
Multi-Graph [37]	74.3	76.2	75.2
PGAT-BERT [27]	79.7	81.7	80.7
ours	81.6	82.1	81.8

Table 4 shows the results on the E-commerce dataset. The classical BERT-LSTM-CRF obtains an F1 score of 73.6%. In addition, the F1 scores in Multi-Graph [37] and PGAT-BERT [27] are 75.2% and 80.7%, respectively. Consistent with the observations in the Weibo and OntoNotes datasets, our model obtained the highest F1 score of 81.8%.

By observation, our model performs best on the Weibo dataset. This shows that our model can deal with data with a certain amount of noise. On the E-commerce dataset in the e-commerce domain, the model also achieves a significant improvement (1.1%). Since the E-commerce dataset only contains two types of entity labels, our model is more suitable for processing datasets with fewer types of entities. Compared with the Resume dataset with eight types of entities, our model's F1 value is 0.13% lower than PGAT-BERT [27], ranking second among all baseline models. Finally, the performance on the OntoNotes dataset is also better than the baseline model, which means that the model can also handle larger datasets. Our model gives the Chinese NER project fresh insights while improving the entity identification performance simply and effectively.

To evaluate the model's performance on real-world datasets, we also consider the Youku dataset, which consists of video titles from Youku.com [39]. We use the same hyperparameter settings as for the Weibo dataset. Jie crawled these data from the Youku video site and manually annotated such data with named entities. They present a novel but easy-to-implement method for identifying named entities with incomplete data annotations. We choose the model in which all entities are retained as the baseline. The results of the experiment can be seen in Table 5. The experimental results in the above table show that the algorithm proposed in this paper obtains better performance than the baseline algorithm. This indicates that our method equally applies to real industrial application scenario datasets.

Table 5. Precision rate, recall rate, and F1 score statistics on the Youku dataset.

Models	P	R	F1
Baseline	83.0	81.7	82.4
Ours	87.0	85.1	86.1

4.4. Ablation Study

To further validate the benefit of the GAT layer and short-sequence CNNs in terms of their respective gains on the model, the outcomes of Weibo, E-commerce, and OntoNotes–based ablation experiments are recorded in Table 6.

Table 6. Ablation study.

Models	Weibo	OntoNotes	E-Commerce
Complete model	71.81	82.99	81.83
w/o SSCNN	71.66	82.66	81.61
w/o GAT	70.69	82.64	81.03
w/o SSCNN and GAT	70.51	82.58	81.21

The ablation studies were designed as follows:

- w/o SSCNN: without short-sequence CNN module;
- w/o GAT: no GAT layer module;
- w/o SSCNN and GAT: there is no short-sequence CNN and GAT layer module.

Removing any module leads to significant performance degradation. Specifically, the performance difference between "w/o GAT" and "Complete model" on Weibo and E-commerce is enormous, especially in the Weibo dataset, where F1 decreases the most (1.12%). This indicates that without the GAT module, the model cannot capture the semantic interaction information between characters and their neighbors. Since dialectal slang and irregular phrases are prevalent in social domains, we must rely on GAT to more accurately capture data correlations to utilize more local information for complex contexts. Similarly, model performance deteriorates when the SSCNN module is removed. The local details captured by the SSCNN module can provide a balanced view of the global sequence details from Bi-LSTM. It is shown that SSCNN can better capture the local semantics of the text, which is crucial for capturing character-based nuances in Chinese text. On the OntoNotes dataset, removing the performance gap between the modules similarly validates the usefulness of the GAT layer and SSCNN for capturing local information. The ablation study demonstrates that the GAT layer and SSCNN are crucial for enhancing the perception of local information in the model. Both character and word neighborhood information helps to enhance the performance of NER, while the combination of the two gives the best results.

4.5. Case Study

To more intuitively demonstrate the rich local information that our model can provide in relation to adjacent terms, Figure 9 shows a case study with and without GAT layers. In Case 1, with the GAT layer, the model can correlate the information of the adjacent matching words and accurately identify the entity "理想 (Li Auto)". Similarly, in Case 2, the label of the entity "俞兆林 (Yu Zhaolin)" is tagged as "PER" as a personal name term. The GAT layer accurately predicts the label of "俞兆林 (Yu Zhaolin)" as "ORG" by integrating the semantic information of the adjacent matching words "品牌 (brand)" and "成立 (found)" of "俞兆林 (Yu Zhaolin)".

To investigate the effect of the local perception enhancement method of this paper on the detection quality of entities, we further observe the performance of extracting entities with fused local information. We selected 300 pieces of data from the Weibo dataset as test samples to test the performance of the prediction of entity types with and without the GAT module. Figure 10 displays the experiment's outcomes. As shown in diagram (a), the values of the model's P, R, and F1 scores are improved after the use of the GAT layer. This indicates that the number and accuracy of entities predicted by our model were enhanced after adding the GAT layer.

In the meantime, we save the output of the model prediction results in the presence and removal of GAT. Filter for differences in the prediction results of entity labels in

each sentence sample under the two conditions. We used tools to successfully select 37 sentence samples from the test set samples and made careful observations and statistics manually. As shown in diagram (b), we found that after adding the GAT module, the model accurately corrected 64 (approximately 16% of total entities) entity label types that the model did not or incorrectly predicted when the GAT layer was removed. We added a total of 28 (approximately 7% of total entities) nonentity false prediction labels. Although sometimes the model incorrectly predicts words as an entity, higher recall can ensure the efficiency of entity lookup in some specific scenarios. In addition, we found from the output sample that the model's processing capacity for a single entity is still insufficient after adding the GAT module, and the model can easily mispredict the consecutive single entity labels "S" and "S" as the whole entities "B" and "E".

Case 1	
Sentence	比亚迪和理想发布了车型. BYD and Li Auto have released their models.
Lexical word	比亚迪(BYD), 理想(Li Auto), 发布(release), 车型(model)
Gold label	***比***, ***亚***, ***迪***, 和, ***理***, ***想***, 发, 布, 了, 车, 型 ***B-ORG***, ***M-ORG***, ***E-ORG***, O, ***B-ORG***, ***E-ORG***, O, O, O, O, O
Without GAT	***B-ORG***, ***M-ORG***, ***E-ORG***, O, ***O***, ***O***, O, O, O, O, O
With GAT	***B-ORG***, ***M-ORG***, ***E-ORG***, O, ***B-ORG***, ***E-ORG***, O, O, O, O, O
Case 2	
Sentence	品牌俞兆林成立于上海. Brand Yu Zhaolin was founded in Shanghai.
Lexical word	品牌(brand), 俞兆林(Yu Zhaolin), 成立(found), 上海(Shanghai)
Gold label	品, 牌, ***俞***, ***兆***, ***林***, 成, 立, 于, ***上***, ***海***. O, O, ***B-ORG***, ***M-ORG***, ***E-ORG***, O, O, O, ***B-LOC***, ***E-LOC***
Without GAT	O, O, ***B-PER***, ***M-PER***, ***E-PER***, O, O, O, ***B-LOC***, ***E-LOC***
With GAT	O, O, ***B-ORG***, ***M-ORG***, ***E-ORG***, O, O, O, ***B-LOC***, ***E-LOC***

Figure 9. Case study. We use bold and italics to identify entities. Characters are split using commas between them: (1) lexical word row: words matched in the lexicon; (2) gold label row: label correct entities as italicized and bolded representations; (3) without GAT row: entities without GAT effect predictions are labeled as italicized and bolded representations; (4) with GAT row: entities with GAT modules are labeled as italicized and bolded.

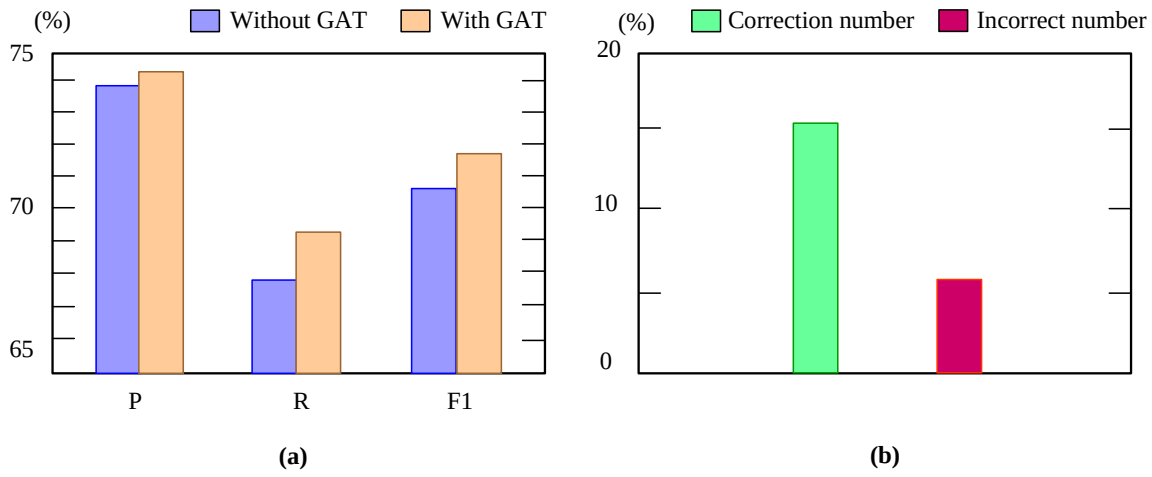

Figure 10. Performance of entity type predictions with and without the GAT module on the Weibo dataset: (**a**) comparison of results with/without the GAT layer; (**b**) ratio of correction/incorrect new entities to total entities.

5. Constraints and Future Work

Integrating lexical information into Chinese character representations effectively improves the performance of Chinese named entity recognition. A continuously accumulated and improved entity lexicon in the vertical field steadily improves NER performance. However, finding a suitable and extensive lexicon and ensuring that the lexicon's content is adapted to the task at hand lead to limitations when using an external lexicon. At the same time, building a lexicon is time-consuming, and the quality of the lexicon may need improvement. In addition, our proposed two-stream network consisting of a lexicon feature layer and a GAT layer affects the interaction between various feature information, complicating the model.

In our future research, we will further investigate how to efficiently integrate interaction knowledge from more distant entity neighborhoods in a single-stream network. Additionally, we will explore how to integrate local features by discovering patterns from the internal composition of entities, thereby enhancing the performance of Chinese NER without relying on external resources.

6. Conclusions

This paper proposes a local information perception enhancement–based method for Chinese NER, through the graph attention network fusion of entity characters with matching words, as well as information about matching words and entity adjacent contextual matching words, thereby enhancing the perception of the entity neighborhood information. Moreover, local text features in short-sequence CNN sliding windows are encoded. Combining local details from CNNs and global sequence details from Bi-LSTMs gives the model a balanced perspective. Chinese datasets from four distinct domains were employed in experiments, and the results demonstrate that our method performs better than the currently used baseline model and significantly enhances Chinese NER performance. A real-world dataset indicates that our method is equally applicable for real industrial application scenario datasets. The case study experiment also indicates that the proposed method can better use the entity's neighborhood information and enhance the precision of entity boundary and type labeling predictions.

Author Contributions: Presenting algorithmic ideas and reviewing and revising the first draft, L.L.; implementing the computer code, writing and revising the first draft, and managing and visualizing data, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Funding through Chongqing Natural Science Foundation (cstc2021jcyj-msxmX0594); funding through Action Plan for High-Quality Development of Graduate Education of Chongqing University of Technology (gzlcx20233205).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This manuscript utilizes publicly available datasets, which can be accessed at the following links: Weibo dataset (https://github.com/quincyliang/nlp-public-dataset/tree/master/ner-data/weibo (accessed on 16 October 2021)), Resume dataset (https://github.com/jiesutd/LatticeLSTM (accessed on 16 October 2021)), OntoNotes dataset (https://catalog.ldc.upenn.edu/LDC2013T19 (accessed on 18 May 2022)), E-commerce dataset (https://github.com/PhantomGrapes/MultiDigraphNER3 (accessed on 13 August 2022)), and Youku dataset (https://github.com/allanj/ner_incomplete_annotation (accessed on 23 August 2023)).

Acknowledgments: We gratefully acknowledge the support of the Department of Computer Science and Technology, Chongqing University of Technology.

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

Abbreviations

The following abbreviations are used in this manuscript:

SSCNN	short-sequence CNN
RNN	recurrent Neural Network
GAT	graph attention networks
CRF	conditional random field

Appendix A

There are some details about the experimental process using systems such as LSTMs, CNNs, and graph attention networks that require fine-tuning of the hyperparameters, and some details about this process are given in the following appendix.

Table A1. Some other hyperparameter settings about the experiment.

Hyperparameter	Value
LSTM model	
Num-Layer	1
Dropout Rate	0.5
Optimizer	ReLU
CNN model	
Num-Layer	1
Dropout Rate	0.1
Window Size	3
Kernel Size	1
Padding	1
GAT model	
Num-Layer	2
Dropout Rate	0.5
Alpha	0.1
Nheads-K	3
Optimizer	LeakyReLU

References

1. Li, J.; Xu, K.; Li, F.; Fei, H.; Ren, Y.; Ji, D. Mrn: A locally and globally mention-based reasoning network for document-level relation extraction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 1359–1370.
2. Diefenbach, D.; Lopez, V.; Singh, K.; Maret, P. Core techniques of question answering systems over knowledge bases: A survey. *Knowl. Inf. Syst.* **2018**, *55*, 529–569.
3. Hou, F.; Wang, R.; He, J.; Zhou, Y. Improving entity linking through semantic reinforced entity embeddings. *arXiv* **2021**, arXiv:2106.08495.
4. Meng, Y.; Wu, W.; Wang, F.; Li, X.; Nie, P.; Yin, F.; Li, M.; Han, Q.; Sun, X.; Li, J. Glyce: Glyph-vectors for chinese character representations. In *Advances in Neural Information Processing Systems, Proceedings of the Annual Conference on Neural Information Processing Systems 2019, Vancouver, BC, Canada, 8–14 December 2019*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2019; Volume 32.
5. Li, H.; Hagiwara, M.; Li, Q.; Ji, H. Comparison of the impact of word segmentation on name tagging for chinese and japanese. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014; pp. 2532–2536.
6. Liu, Z.; Zhu, C.; Zhao, T. Chinese named entity recognition with a sequence labeling approach: Based on characters, or based on words? In Proceedings of the International Conference on Intelligent Computing, Changsha, China, 18–21 August 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 634–640.
7. Zhang, Y.; Yang, J. Chinese NER using lattice LSTM. *arXiv* **2018**, arXiv:1805.02023.
8. Gui, T.; Ma, R.; Zhang, Q.; Zhao, L.; Jiang, Y.G.; Huang, X. CNN-based Chinese NER with lexicon rethinking. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, 10–16 August 2019; pp. 4982–4988.

9. Li, X.; Yan, H.; Qiu, X.; Huang, X. FLAT: Chinese NER using flat-lattice transformer. *arXiv* **2020**, arXiv:2004.11795.
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems, Proceedings of the 2017 Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2017; Volume 30.
11. Liu, W.; Xu, T.; Xu, Q.; Song, J.; Zu, Y. An encoding strategy based word-character LSTM for Chinese NER. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (Long and Short Papers), pp. 2379–2389.
12. Luo, H.; Lu, L. Character embedding method for chinese named entity recognition. *J. Chin. Comput. Syst.* **2023**, *7*, 1434–1440. [CrossRef]
13. Ma, R.; Peng, M.; Zhang, Q.; Huang, X. Simplify the usage of lexicon in Chinese NER. *arXiv* **2019**, arXiv:1908.05969.
14. Zhao, S.; Hu, M.; Cai, Z.; Chen, H.; Liu, F. Dynamic modeling cross-and self-lattice attention network for chinese NER. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 14515–14523.
15. Gu, Y.; Qu, X.; Wang, Z.; Zheng, Y.; Huai, B.; Yuan, N.J. Delving Deep into Regularity: A simple but effective method for chinese named entity recognition. *arXiv* **2022**, arXiv:2204.05544.
16. Liu, S.; Yang, H.; Li, J.; Kolmani, S. Chinese named entity recognition method in history and culture field based on BERT. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 163. [CrossRef]
17. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
18. Chen, B.; Chen, X. MAUIL: Multilevel attribute embedding for semisupervised user identity linkage. *Inf. Sci.* **2022**, *593*, 527–545.
19. Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; Li, F. Unified named entity recognition as word-word relation Classification. *arXiv* **2021**, arXiv:2112.10070.
20. Chiu, J.P.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
21. Yan, H.; Deng, B.; Li, X.; Qiu, X. TENER: Adapting transformer encoder for named entity recognition. *arXiv* **2019**, arXiv:1911.04474.
22. 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, Lisbon, Portugal, 17–21 September 2015; pp. 548–554.
23. Xue, N. Chinese word segmentation as character tagging. *Int. J. Comput. Linguist. Chin. Lang. Process.* **2003**, *8*, 29–48.
24. Lafferty, J.; Mccallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning, Williamstown, MA, USA, 28 June–1 July 2001.
25. Gui, T.; Zou, Y.; Zhang, Q.; Peng, M.; Fu, J.; Wei, Z.; Huang, X.J. A lexicon-based graph neural network for Chinese NER. In Proceedings of the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 1040–1050.
26. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
27. Wang, Y.; Lu, L.; Wu, Y.; Chen, Y. Polymorphic graph attention network for Chinese NER. *Expert Syst. Appl.* **2022**, *203*, 117467. [CrossRef]
28. Nie, Y.; Tian, Y.; Song, Y.; Ao, X.; Wan, X. Improving named entity recognition with attentive ensemble of syntactic information. *arXiv* **2020**, arXiv:2010.15466.
29. Zhu, P.; Cheng, D.; Yang, F.; Luo, Y.; Huang, D.; Qian, W.; Zhou, A. Improving chinese named entity recognition by large-scale syntactic dependency graph. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 979–991. [CrossRef]
30. Cetoli, A.; Bragaglia, S.; O'Harney, A.D.; Sloan, M. Graph convolutional networks for named entity recognition. *arXiv* **2017**, arXiv:1709.10053.
31. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. *arXiv* **2016**, arXiv:1603.01360.
32. Wang, Y.; Yu, B.; Zhu, H.; Liu, T.; Yu, N.; Sun, L. Discontinuous named entity recognition as maximal clique discovery. *arXiv* **2021**, arXiv:2106.00218.
33. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
34. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
35. Forney, G.D. The viterbi algorithm. *Proc. IEEE* **1973**, *61*, 268–278. [CrossRef]
36. Weischedel, R.; Pradhan, S.; Ramshaw, L.; Palmer, M.; Xue, N.; Marcus, M.; Taylor, A.; Greenberg, C.; Hovy, E.; Belvin, R.; et al. *Ontonotes Release 4.0*; LDC2011T03; Linguistic Data Consortium: Philadelphia, PA, USA, 2011.
37. Ding, R.; Xie, P.; Zhang, X.; Lu, W.; Li, L.; Si, L. A neural multi-digraph model for Chinese NER with gazetteers. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Fortezza da Basso, FL, USA, 28 July–2 August 2019; pp. 1462–1467.

38. Mengge, X.; Bowen, Y.; Tingwen, L.; Yue, Z.; Erli, M.; Bin, W. Porous lattice-based transformer encoder for Chinese NER. *arXiv* **2019**, arXiv:1911.02733.
39. Jie, Z.; Xie, P.; Lu, W.; Ding, R.; Li, L. Better modeling of incomplete annotations for named entity recognition. In Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Minneapolis, MN, USA, 6 June 2019.