Exp-SoftLexicon Lattice Model Integrating Radical-Level Features for Chinese NER

Lijie Li[1], Shuangyang Hu[1], Junhao Chen[1], Ye Wang[1,*], Zuobin Xiong[2]
[1] College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[2] Department of Computer Science, Georgia State University, Atlanta, USA
{lilijie, sunyonge, junhaochen, wangye2020}@hrbeu.edu.cn, zxiong2@student.gsu.edu

***Abstract*—The Lattice series model using potential words information has been proved to be effective in Chinese Named Entity Recognition (NER). The recently proposed Simplified Lattice not only brings new baseline results, but also improves the inference speed of Lattice models. However, the Simplified Lattice fails to fully explore the rich information contained in the radical-level features of the character sequences. Moreover, the performance of the Simplified Lattice decreases dramatically as the length of entity increases. In this paper, we propose the SLRL-NER model that integrates word, character, and radical-level information to alleviate the above problems. Specifically, text Convolutional Neural Network (CNN) is used to extract the radical-level features. The original SoftLexicon set is expanded to refine the relative position information of characters in the candidate words to cope with the challenge of increasing entity length. Experiments on three datasets show SLRL-NER outperforms the state-of-the-art comparison methods.**

Keywords- Chinese NER; radical-level features; SoftLexicon

I. Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that aims at identifying entities from plain text segments and tagging them with types, such as person, location, organization, etc. Many high level tasks, such as question answering [1], knowledge graph construction [2], and information retrieval [3] are all inseparable from NER.

Due to the implicit word boundary of Chinese sentences, Chinese NER is more difficult than English NER. The existing Chinese NER methods are mainly classified into word-based and character-based models. In the word-based model, a Chinese Word Segmentation (CWS) system is firstly used to segment the original input sequence. However, the performance of word-based model has been proved to be worse than character-based model because of the word segmentation errors from the CWS system [4]. In the character-based model, segmentation errors are naturally avoided. But character-based model does not exploit word-level information in input sequence at all. To integrate words information into character-based model, many works have attempted to use external gazetteers to extract word information [5-7]. Yue Zhang and Jie Yang [4] proposed the Lattice-LSTM model which encodes the matched lexical words information into character sequence with a gate mechanism and achieved a great success. Next, Ma et al. [8] proposed the Simplified Lattice model, where they constructed a SoftLexicon set for each word and used a word-weighting strategy to fuse word-level information. However, the Simplified Lattice model still faces two challenges.

Firstly, to the best of our knowledge, existing lattice models do not explore the radical-level information inherent in the Chinese characters. Radicals originate from pictograms [9], which are the minimum semantic units for Chinese characters. Different from the character representations obtained by the pretrained language model, radical-level features are context-independent [10] and relates only to the character itself. Thus, they have additional intrinsic information that is not present in the pretrained character embeddings. Dong et al. [11] have shown that the radical-level features can effectively improve the performance of Chinese NER. However, they ignored the word information in the sequence which has been demonstrated to be very important for Chinese NER by many works.

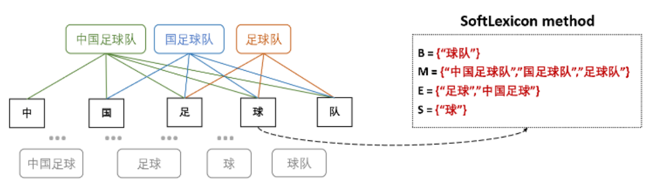

Fig. 1. The SoftLexicon method. “BMES” indicates the position of the character in the corresponding candidate word is: begin, middle, end and single.

Secondly, although the Simplified Lattice utilizes word information in a straightforward and efficient way, there is still a loss of position information for long words. As shown in Fig. 1, for the input sentence “中国足球队 (Chinese football team)”, according to the position of the character in the word, the “middle” group word set of the character “球 (ball)” includes: “足球 (football team)”, “国足球队 (Chinese football team)” and “中国足球队 (Chinese football team)”. The character “球 (ball)” is the second, third and fourth character in three different words. However, the SoftLexicon method indistinguishably places the three different words of the character “球 (ball)” in the “Middle” group, which does not properly identify their specific relative positions within the words. As a result, this method loses a large amount of relative position information, which is proved to be essential for Chinese NER [12], and the problem becomes more serious as the length of entity grows.

DOI reference number: 10.18293/SEKE2022-037.
*Corresponding authors

In this paper, we consider all these issues systematically and present a novel exp-SoftLexicon Lattice Model Integrating Radical-Level Features for Chinese NER (SLRL-NER) to deal with these issues. **For radical-level information**: we elaborate a text CNN to extract radical-level information of characters from three different perspectives: radical, structural and unique position features. **To further exploit the position information of characters in candidate words**: we expand the original SoftLexicon module to enrich the relative position information of the "Middle" group to cope with the challenge of increased length of entity. Finally, we aggregate the radical-level and word-level information into the character representations to predict the named entity tags.

In summary, this paper makes the following contributions: (1) We propose a novel lattice structure to incorporate character-level, word-level and radical-level information of sentences for Chinese NER. Our model can capture both the inherent information and the rich context information for Chinese characters. (2) We introduce radical-level information into Chinese NER lattice model and design a text CNN module to exploit the radical-level features of characters from three different perspectives: radical, structural and unique position features. In addition, we apply an exp-SoftLexicon module to refine the relative position information of characters in the candidate words to deal with the challenge of increasing entity length. (3) Experimental results on three public Chinese NER datasets show that our model achieves better performance compared with the state-of-the-art methods.

II. Background

In this section, we introduce some methods relevant to this paper, including radical-level features, character pre-trained methods and SoftLexicon module.

A. Radical-Level Features

Different from English, Chinese characters are pictographic, and most of them still retain the original pictogram meaning. In particular, the morphological information is mainly reflected in the radical, structural and unique position features of the characters [11]. Furthermore, the radical is the basic constituent unit of a Chinese character and contains both simplified and traditional forms, which is closely related to the meaning of the character. Structural features consist of the decomposition of a Chinese character and have the meta-information of the characters. The unique position contains the absolute position sequence of each character in writing order. For a monoradical character, we just use itself as its radical-level features. Fig. 2 shows the basic information and meaning of the Chinese character "烫 (hot)"[1]. The radical-level features of "烫 (hot)" include: (1) Radical feature: A simplified radical consisting of four strokes, which contains semantic information about the character; (2) Structural features: "烫 (hot)" is composed of three monoradical: "氵 (water)", "扬 (raise)" and "火 (fire)", which cover the meta-information of the character; (3) Unique position: The absolute position sequence of the unique writing order of the character "烫 (hot)", i.e. "汤 (soup)" and "火 (fire)".

[1] from the online Xinhua dictionary at http://tool.httpcn.com/Zi/.

Fig. 2. Basic information about character "烫 (hot)".

The fine-gained semantic information of "烫 (hot)" is extrated by exploring the radical-level features. We did not continue to excavate the Wubi features of characters because radical features are the smallest semantic units of Chinese characters while Wubi features usually do not have obvious semantic information [11].

B. Character Embedding

More and more works choose pre-trained model BERT [13] instead of word2vec to server as the character encoder. BERT fuses token embeddings, position embeddings and segment embeddings as input to obtain a better dynamic vector representation with a deep network and huge number of parameters. However, BERT only masks a single character in the Chinese sequence, which obviously loses part of the word-level semantic information. BERT-wwm-ext [14] is built on BERT by using a larger corpus to mask all the consecutive Chinese characters that composed the words, so that the embeddings of characters have the semantic information of the words.

C. SoftLexicon

For the input sentence $s = \{c_1, c_2, \ldots, c_n\}$, an external gazetteer L is used to match the latent words corresponding to each character c_i. Then, a specific SoftLexicon [8] set is constructed for each character c_i: each word is assigned to the "BMES" word sets according to the position of the character c_i in the corresponding latent word. Then the word-level representation of the character can be obtained by integrating words information in the SoftLexicon set.

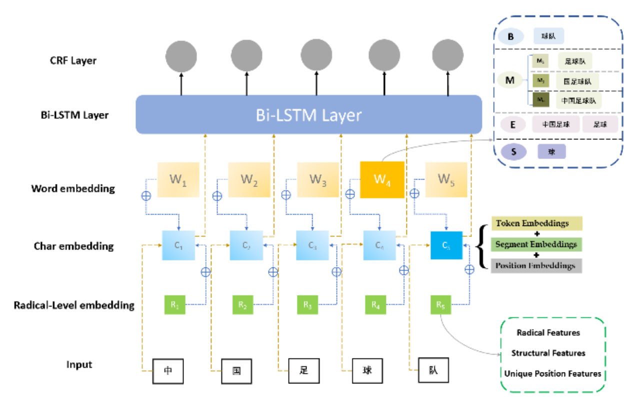

Fig. 3. The whole architecture of SLRL-NER. The top right part represents the exp-SoftLexicon set and ⊕ indicates the concatenation operation.

III. Approach

In this work, we propose the SLRL-NER model, which merges three types of information with different granularity: word, character and radical-level features to make full use of the semantic information in the input sequence and achieve excellent experimental results. The architecture of our model is shown in Fig. 3.

A. Radical-Level Representations Layer

The model input is a sentence $s = \{c_1, c_2, \ldots, c_n\}$, and each character $c_i = \{r_1, r_2, \ldots, r_m\} \in V_r$, where r_j denotes the radical-level features of the character. We use the radical-level lookup table V_r [11] which contains 4719 common Chinese characters. Each radical-level features r_j is represented by a dense vector $\boldsymbol{y}_j^r$:

$$\boldsymbol{y}_j^r = e^r(r_j)\,, \tag{1}$$

where e_r denotes the radical-level features embedding lookup table. Then we can get the radical-level embedding matrix $\boldsymbol{O} = \{\boldsymbol{y}_1^r, \boldsymbol{y}_2^r, \ldots, \boldsymbol{y}_m^r\}$ of the character c_i. To enable parallelization, the shape of radical-level embedding matrix $\boldsymbol{O}$ is set to be the form of $50 \times k$ where k is a hyper-parameter. For characters with more than k radical-level features, we perform a squeeze operation to take the top-k radical-level features, and for characters with less than k features, we proceed to random initialization to fill the feature matrix $\boldsymbol{O}$ to the fixed dimension. After obtaining the feature matrix $\boldsymbol{O} = \{\boldsymbol{y}_1^r, \boldsymbol{y}_2^r, \ldots, \boldsymbol{y}_k^r\}$, the unsupervised CNN is employed to extract the radical-level features of the characters, as shown in Fig. 4.

Since the structural and unique position features of each character c_i mainly occur in pairs, the CNN apply filters $\boldsymbol{H} \in \mathbb{R}^{50\times 2}$ with a window size of 2. After x successive convolutions of $\boldsymbol{O}$, the radical-level features are extracted using maximum pooling and then represented as a 50-dimensional vector $\boldsymbol{y}^r$.

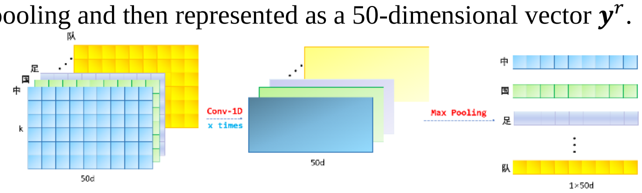

Fig. 4. Extracting radical-level features of characters in sequences using Convolutional Neural Networks.

B. Character Representation Layer

The character c_i in sentence is represented using a 768-dimensional dense vector $\boldsymbol{x}_i^c$:

$$\boldsymbol{x}_i^c = e^c(c_i). \tag{2}$$

Here, e^c denotes the character embedding lookup table, which is derived from BERT-wwm-ext [14]. BERT [13] masked the characters in the sequence without fully preserving the semantic information of the word. In BERT-wwm-ext, besides a character that constitutes a word, other parts that belong to the same word are masked, so that the pre-trained model is endowed with the semantic information of the word. As shown in Fig. 3, the final embedding representation of each character consists of token embeddings, position embeddings, and segment embeddings. After training with the full word masking strategy, the character embedding representation $\boldsymbol{x}^c$ is obtained.

C. Word Representation Layer

We build an exp-SoftLexicon set for each character in the sentence. Firstly, latent words are filtered by lexicon and classified into the six word sets "$BM_1M_2M_oES$" according to the position of the characters in the candidate words. The six word sets are constructed as follows:

$$\begin{aligned}
&B(c_i) = \{w_{i,k} \mid \forall w_{i,k} \in L, 1 \le i < k \le n\} \\
&M_1(c_i) = \{w_{j,k} \mid \forall w_{j,k} \in L, 2 \le j+1 = i < k \le n\} \\
&M_2(c_i) = \{w_{j,k} \mid \forall w_{j,k} \in L, 3 \le j+2 = i < k \le n\} \\
&M_o(c_i) = \{w_{j,k} \mid \forall w_{j,k} \in L, 3 \le j+2 < i < k \le n\} \\
&E(c_i) = \{w_{j,i} \mid \forall w_{j,i} \in L, 1 \le j < i \le n\} \\
&S(c_i) = \{c_i \mid \exists c_i \in L\}
\end{aligned} \tag{3}$$

Here, L denotes the lexicon we use in this work and $w_{i,j}$ denotes the matched word starting from the i-th character and ending at the j-th character. Additionally, if a word set is empty, a special word "NONE" is added to the empty word set. An example is shown in Fig. 3. By expanding the position of "Middle", our model distinguishes the candidate words in different positions in the "Middle" group more accurately, which reduces the loss of relative position of the SoftLexicon method and improves NER performance. After obtaining the expanded "$BM_1M_2M_oES$" word sets for each character, each word set is then compressed into a vector with fixed dimension. We obtain the representation of the word set W by a weighted strategy, then we concatenate all the six word sets representation to get the exp-SoftLexicon embedding z^w:

$$v^w(W) = \frac{6}{Z} \sum_{w \in W} z(w) e^w(w)\,, \tag{4}$$

$$Z = \sum_{W \in B \cup M_1 \cup M_2 \cup M_o \cup E \cup S} \; \sum_{w \in W} z(w)\,, \tag{5}$$

$$z^w = [v^w(B); v^w(M_1); v^w(M_2); v^w(M_o); v^w(E); v^w(S)], \tag{6}$$

where v^w denotes the weighting function, $e^w(w)$ denotes the embedding vector of word w, $z(w)$ denotes the number of occurrences of lexical word w in the statistical data, W denotes one of the "$BM_1M_2M_oES$" word sets corresponding to character c_i, Z is the sum of occurrence of all matched words in the six words sets, and z^w denotes the exp-SoftLexicon embedding vector corresponding to the character. In this work, the statistical dataset is made up of training and validation data. In addition, the frequency of w does not increase if w is covered by a subsequence of another matching lexical word. This avoids the problem that the frequency of a shorter word is always less than the frequency of the longer word containing it.

D. Bi-LSTM Layer

After obtaining three different granularity features of the input sequence, the next step is to preserve their individual information as completely as possible and integrate them into the character representation. We finally choose to concatenate the representation vectors of the three, and the final representation of each character is obtained by:

$$\boldsymbol{x}^c \leftarrow [(\boldsymbol{x}^c; \boldsymbol{y}^r; \boldsymbol{z}^w)]. \tag{7}$$

Then, the final representations of characters are fed into Bi-LSTM. The definition of LSTM is as follows:

$$\begin{bmatrix} \boldsymbol{i}_t \\ \boldsymbol{f}_t \\ \boldsymbol{o}_t \\ \tilde{\boldsymbol{c}}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \left(W \begin{bmatrix} \boldsymbol{x}_t^c \\ \boldsymbol{h}_{t-1} \end{bmatrix} + b \right),$$
$$\boldsymbol{c}_t = \tilde{\boldsymbol{c}}_t \odot \boldsymbol{i}_t + \boldsymbol{c}_{t-1} \odot \boldsymbol{f}_t, \tag{8}$$
$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t).$$

where σ is the element-wise sigmoid function and $\odot$ denotes the product of elements, W and b are trainable parameters. Memory unit c can be considered as long-term memory and hidden state $\boldsymbol{h}$ as short-term memory. The backward LSTM shares the same definition as the forward LSTM, but models sequences in the opposite order. The hidden states of the i-th step of forward and backward LSTM are concatenated together to form the context-dependent representation of $\boldsymbol{c}_i$.

E. CRF Decoding Layer

Finally, to parse the dependencies between the continuous labels, we use a standard Conditional Random Field (CRF) [15] layer to perform sequence tagging. We use the following equation to calculate the score of the labels sequence: $y = \{y_1, y_2, \ldots, y_n\}$:

$$Score(X, Y) = \sum_{i=1}^{n} P_{i,y_i} + \sum_{i=0}^{n} T_{y,y_{i+1}}, \tag{9}$$

where P is the output of the Bi-LSTM, which represents the emission score of the tag y_i in the sentence, and the transition matrix T denotes the transition probability from tag y_i to tag y_{i+1}.

In this work, we use the sentence-level negative log-likelihood loss function to train the model, and $L2$ regularization with the parameter λ is used to alleviate overfitting:

$$\mathcal{L} = Score(X, Y) - \log \sum_{y^i \in Y} e^s(x, y_i) + \frac{\lambda}{2} \|\theta\|^2. \tag{10}$$

IV. EXPERIMENT

A. Experimental Setup

Datasets. We evaluate the proposed model on three standard Chinese NER datasets, including **OntoNotes 4.0** [16], **Weibo** [17-18], and **Resume** [4]. **OntoNotes 4.0** is drawn from the news domain and contains four types of named entity. **Weibo** is built based on Chinese social media Sina Weibo, which contains PER, ORG, GEP, and LOC for both named entity and nominal mention. **Resume** is composed of resumes collected from Sina Finance. It is annotated with 8 types of named entities. We use the same dataset split and lexicon as the Simplified Lattice. The lexicon consists of 5.7k single-character words, 291.5k two-character words, 278.1k three-character words, and 129.1k other words. As for the pre-trained word embeddings, we also use the same one as the Simplified Lattice, which are pre-trained on Chinese Giga-Word using Word2vec [19].

Implementation Detail. In this work, the model is trained using stochastic gradient descent. The initial learning rate for Weibo is 0.005 and the other datasets are 0.0015. We apply dropout [20] to embedding layer with rate of 0.5 in order to avoid overfitting. Additionally, the hidden size of Bi-LSTM is set to 200 for small datasets **Weibo** and **Resume**, and 400 for larger dataset **OntoNotes 4.0**.

B. Experimental Results

Table 1 display the experimental results on three datasets. The character-based model gets performance boosted with the addition of softword and bichar features, which demonstrates it is critical to cooperate lexicon and character features in Chinese NER task. Another notable observation is that models use BERT encoder persistently outperform those without BERT. It indicates pre-trained character embeddings with context awareness are significant to this work. After replacing the encoder of Simplified Lattice with BERT-wwm-ext, we observe no significant change in performance, and even a decrease on the Resume dataset. This indicates that Simplified Lattice does not exploit the full potential of BERT-wwm-ext. GLYNN [21] improved the performance of Chinese NER by using a CNN encoder to integrate glyph features from character images, indicating that Chinese NER can benefit from the pictogram features. The SLRL-NER model integrates lexicon, character, and radical-level features leads to 0.47 and 0.25 increments of F1 score over the state of-the-art model on **OntoNotes 4.0** and **Resume**, respectively.

TABLE I MAIN RESULTS ON THE THREE DATASETS

Models	**OntoNotes 4.0**	**Resume**	**Weibo**		
	F1	***F1***	***NE***	***NM***	***F1***
Char-based	64.30	93.48	46.11	55.29	52.77
+bichar+softword	71.89	94.41	50.55	60.11	56.75
Peng and Dredze	–	–	55.28	62.97	58.99
He and Sun	–	–	54.50	62.17	58.23
SLK-NER	80.20	95.80	–	–	64.00
GLYNN+BERT	–	95.66	–	–	69.00
Lattice-LSTM	73.88	94.46	53.04	62.25	58.79
Simplified Lattice	75.64	95.53	59.08	62.22	61.42
Simplified Lattice+BERT	82.81	96.11	70.94	67.02	70.50
Simplified Lattice+BERT-wwm-ext	82.85	95.98	71.07	67.64	70.77
SLRL-NER	**83.28**	**96.36**	**72.15**	**68.62**	**71.90**

In the results on **Weibo**, NE, NM and Overall denote F1 scores for named entities, nominal entities (excluding named entities) and both, respectively. Weibo is made up of short and informal texts created by users, so it is difficult to identify. Therefore, baseline models adopt multitask learning with character embedding feature [22], semi-supervised learning, and cross-domain learning [23]. However, the architecture of the above models is complicated and may bring noise to the task, so these models only achieve limited improvement. In contrast, we use exp-SoftLexicon module to fuse the information of multiple candidate words to efficiently reduce word boundary conflicts. Additionally, we fully consider the radical-level, character-level and word-level semantic information in the sentence to alleviate the data sparsity problem. The experimental results show that our method is superior on social media domain compared with the state-of-the-art methods, which has 1.40 F1 score improvement.

C. Robustness Research

We perform experiments to verify the robustness of SLRL-NER. The results are shown in Fig. 5. The NER task becomes more challenging as the length of the named entity increases. The F1 scores on all three datasets suffers certain decrease. Specifically, the performance of both Simplified Lattice and Lattice-LSTM showed a large degree of fluctuation and degradation. SLRL-NER presents only a small decrease under different testing environments. Compared with the Simplified Lattice and Lattice-LSTM, our method is more robust. All the experiments are conducted on a single GPU with Quadro RTX 8000.

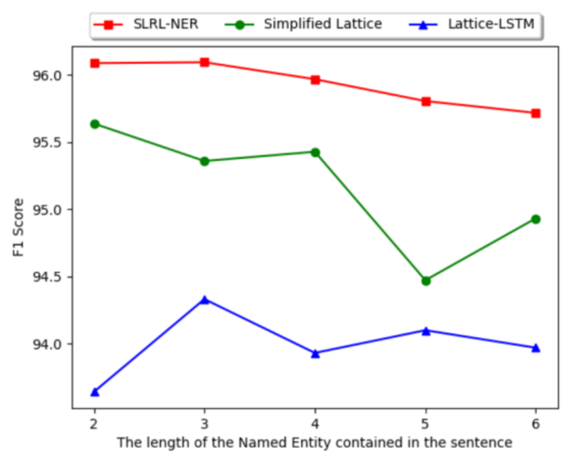

Fig. 5. Model performance against entity lengths. Batch size = 8 for SLRL-NER and Simplified Lattice models. Lattice-LSTM can only be trained with batch size = 1 due to its DAG structure.

D. Ablation Study

To investigate the contribution of each component of our method, we conducted ablation studies on all three datasets, the results are reported in Table 2. We find that: (1) Without radical-level features, F1 scores decrease 0.51, 1.36, and 0.46 on three datasets respectively. Radical-level features brings significant performance improvement. (2) After replacing the exp-SoftLexicon component with the standard SoftLexicon structure, the F1 score of SLRL-NER also decreases. It indicates the usefulness of finer grained relative position information in the input sequence.

TABLE II ABLATION STUDY OF SLRL-NER

Models	OntoNotes	Weibo	Resume
SLRL-NER	**83.28**	**71.90**	**96.36**
-radical-level features	82.77	70.54	95.90
- exp-Middle Group	82.89	71.16	96.21

V. Conclusion

In this work, we propose SLRL-NER, a noval lattice model which incorporates radical-level, character-level, and word-level information for Chinese NER. In order to leverage the radical-level features of the characters, we design a text CNN module to extract the radical-level information. The exp-SoftLexicon module is used to precisely capture the relative position information of characters in the potential words, which efficiently mitigates the challenge caused by the increase of entity length. Experiments on three Chinese NER datasets from different domains demonstrate our approach is superior compared with the state-of-the-art methods.

Acknowledgment

This work was supported by the National Key R&D Program of China under Grant No. 2020YFB1710200.

References

[1] Dennis Diefenbach, Vanessa Lopez, Kamal Deep Singh, and Pierre Maret. Core techniques of question answering systems over knowledge bases: a survey. Knowl. Inf. Syst., 55(3):529–569, 2018.

[2] LiYang LiuQiao, LiuYao DuanHong, et al. Knowledge graph construction techniques. Journal of computer research and development, 53(3):582, 2016.

[3] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In ACL, pages 167–176. The Association for Computer Linguistics, 2015.

[4] Yue Zhang and Jie Yang. Chinese NER using lattice LSTM. In ACL, pages 1554–1564. Association for Computational Linguistics, 2018.

[5] Wei Liu, Tongge Xu, QingHua Xu, Jiayu Song, and Yueran Zu. An encoding strategy based word-character LSTM for chinese NER. In NAACL-HLT, pages 2379–2389. Association for Computational Linguistics, 2019.

[6] Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. FLAT: chinese NER using flat-lattice transformer. In ACL, pages 6836–6842. Association for Computational Linguistics, 2020.

[7] Yuyang Nie, Yuanhe Tian, Xiang Wan, Yan Song, and Bo Dai. Named entity recognition for social media texts with semantic augmentation. In EMNLP, pages 1383–1391. Association for Computational Linguistics, 2020.

[8] Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. Simplify the usage of lexicon in chinese NER. In ACL, pages 5951–5960. Association for Computational Linguistics, 2020.

[9] Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. Radical embedding: Delving deeper to chinese radicals. In ACL, pages 594–598. The Association for Computer Linguistics, 2015.

[10] Yanran Li, Wenjie Li, Fei Sun, and Sujian Li. Component-enhanced chinese character embeddings. In EMNLP, pages 829–834. The Association for Computational Linguistics, 2015.

[11] Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. Character-based LSTM-CRF with radical-level features for Chinese named entity recognition. In NLPCC/ICCPOL, volume 10102 of Lecture Notes in Computer Science, pages 239–250. Springer, 2016

[12] Mengge Xue, Bowen Yu, Tingwen Liu, Yue Zhang, Erli Meng, and Bin Wang. Porous lattice transformer encoder for chinese NER. In COLING,

pages 3831–3841. International Committee on Computational Linguistics, 2020.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT, pages 4171–4186. Association for Computational Linguistics, 2019.

[14] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. Pre-training with whole word masking for chinese BERT.CoRR, abs/1906.08101, 2019.

[15] John D. Lafferty, Andrew McCallum, and FernandoC. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML, pages 282–289. Morgan Kaufmann, 2001.

[16] Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. Ontonotes release 4.0. LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium, 2011.

[17] Nanyun Peng and Mark Dredze. Named entity recognition for chinese social media with jointly trained embeddings. In EMNLP, pages 548–554. The Association for Computational Linguistics, 2015.

[18] Hangfeng He and Xu Sun. F-score driven max margin neural network for named entity recognition in Chinese social media. In EACL (2), pages 713–718. Association for Computational Linguistics, 2017

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In ICLR (Workshop Poster), 2013.

[20] Nitish Srivastava, Geoffrey E. Hinton, AlexKrizhevsky, Ilya Sutskever, and Ruslan Salakhut-dinov. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res.,15(1):1929–1958, 2014.

[21] Chan Hee Song and Arijit Sehanobish. Using Chinese glyphs for named entity recognition (student abstract). In AAAI, pages 13921–13922. AAAI Press, 2020.

[22] Nanyun Peng and Mark Dredze. Improving named entity recognition for chinese social media with word segmentation representation learning. In ACL. The Association for Computer Linguistics, 2016.

[23] Hangfeng He and Xu Sun. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In AAAI, pages 3216–3222. AAAI Press, 2017.