EasyChair Preprint

№ 3340

Joint Self-Attention and Multi-Embeddings for Chinese Named Entity Recognition

Cijian Song, Yan Xiong, Wenchao Huang and Lu Ma

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 6, 2020

Joint Self-Attention and Multi-Embeddings for Chinese Named Entity Recognition

Cijian Song*, Yan Xiong*, Wenchao Huang*, Lu Ma†
*School of Computer Science and Technology
University of Science and Technology of China, Hefei, Anhui, PR China
†Beijing Institute of Remote Sensing, Beijing 122000, China
Email: song1995@mail.ustc.edu.cn, {yxiong, huangwc}@ustc.edu.cn, syml1234@163.com

***Abstract*—Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP), but it remains more challenging in Chinese due to the particularity and complexity of Chinese. Traditional Chinese Named Entity Recognition (Chinese NER) methods require cumbersome feature engineering and domain-specific knowledge to achieve high performance. In this paper, we propose a simple yet effective neural network framework for Chinese NER, named A-NER. A-NER is the first Bidirectional Gated Recurrent Unit - Conditional Random Field (BiGRU-CRF) model that combines self-attention mechanism with multi-embeddings technology. It can extract richer linguistic information of characters from different granularities (e.g., radical, character, word) and find the correlations between characters in the sequence. Moreover, A-NER does not rely on any external resources and hand-crafted features. The experimental results show that our model outperforms (or approaches) existing state-of-the-art methods on different domain datasets.**

I. Introduction

As a basic task in NLP, NER has drawn constant research attention for a few decades. Formally, NER aims to locate and classify named entities in text into predefined categories, such as person (PER), organization (ORG), location (LOC), geopolitical entity (GPE). NER is the first step in many NLP tasks, e.g., information extraction, knowledge graph. Today, NER for corpus texts from different sources will lead to more promising downstream applications.

Most related research treats NER as a sequence labeling task. Traditional methods for English NER mostly are linear statistical models, including Hidden Markov Model (HMM) and Conditional Random Field (CRF) [9], which require large amounts of knowledge (e.g., orthographic features, gazetteers). With the rapid development of deep learning, neural networks outperform popular statistical algorithms. Recurrent Neural Network (RNN), together with its variants such as Long Short-Term Memory (LSTM) [7] and Gated Recurrent Unit [3], have shown great success in modeling sequential data. Existing methods [10], [12] for English NER achieve state-of-the-art performance by using LSTM-CRF models and incorporating character information into word representations. Although the RNN-based models can handle long-distance dependencies, they tend to be biased towards the most recent inputs in the sequence. The work [15] applies self-attention to semantic role labeling task to draw structural information and global dependencies of long sentences.

Compared with English NER, Chinese NER is more challenging. Chinese has many more complicated properties than English, such as the lack of natural delimiters, complex composition forms and nesting definitions, unavailable conventional linguistic features, and so on. Besides, the same radical usually implies similar semantics and usage since Chinese is hieroglyphic. Some work based on LSTM-CRF [5], [19] or GRU-CRF [17] models attempts to address these challenges. But these methods either depend on carefully designed features and external resources or perform unsatisfactorily, making them difficult to adapt to new domains.

In this paper, to solve the above problems, we investigate a BiGRU-CRF model combining self-attention and multi-embeddings technologies, named A-NER. For an input sentence, in order to use background knowledge, we use pre-trained embeddings on a large corpus to initialize character and word embeddings, and radical embeddings are randomly initialized. The multi-embeddings layer captures the semantic information of characters from different granularities, ranging from radical-, character- to word-level. It consists of three parts: (i) a Convolutional Neural Network (CNN) that encodes radical composition information of each character into its radical-level embedding; (ii) to utilize the order of a sequence, we add positional encoding to character embedding to get character-level embedding; (iii) a Convolutional Gated Recurrent Unit (GRU-Conv) network that generates word-level embedding to learn higher-level features based on the context of words. Then, these embeddings are concatenated and fed to the self-attention layer to produce the final character representation. The self-attention mechanism can automatically focus on specific characters related to Chinese NER and find the correlations between characters. Finally, the BiGRU-CRF network takes the final character representations as input to perform sequence label prediction.

This paper makes the following contributions:

- We present a novel BiGRU-CRF framework for Chinese NER based on self-attention and multi-embeddings, which can learn richer semantic features about characters and capture structural information of sequences.
- A-NER is a simple, effective, and end-to-end model that can be easily applied to other tasks derived from NER.
- A-NER does not need feature engineering and external resources, which achieves remarkable performance.

The rest of this paper is organized as follows. Section II introduces the related work. Section III describes our model for Chinese NER. Section IV presents the experimental setup and results. Section V concludes this paper.

II. Related Work

After the concept of NER was proposed at Message Understanding Conference-6, people have done a lot of research. In recent years, several neural network architectures have been proposed for English NER. Collobert et al. [4] designed a CNN-CRF model that requires little feature engineering, reaching competitive results to the best statistical models. To extract word-level information, Huang et al. [8] first applied an LSTM-CRF network by using carefully designed spelling features. Lample et al. [10] and Ma and Hovy [12] presented two models similar to the work [8]. The former added an LSTM to automatically model the character-level spelling information of words, while the latter used a CNN.

Some work solves the challenges of Chinese NER. The statistical models [2], [18], [20] leveraged rich hand-crafted features. Peng and Dredze [13] explored a joint LSTM-CRF model that trains the positional character embedding and word embedding. E and Xiang [6] exploited character-word mixed embeddings. Inspired by the work [10], Dong et al. [5] used an LSTM to introduce radical-level spelling features. Self-attention was applied by Cao et al. [1] to adversarial transfer learning to improve model performance. Peng and Dredze [14] combined Chinese Word Segmentation with NER. A lattice-structured LSTM was proposed by Zhang and Yang [19], which leverages external lexicon data. Recently, Xu et al. [17] utilized multiple embeddings on GRU-CRF for NER without considering the radical composition features and interactions of characters in the sequence.

Different from existing work, A-NER is the first BiGRU-CRF model for Chinese NER by utilizing self-attention mechanism and multi-embeddings technology. It can effectively extract linguistic information of characters in the sequence and find their correlations, thereby improving the performance of the NER task.

III. Model Architecture

In this section, we describe the model in detail. The main architecture of A-NER is shown in Fig. 1.

Overall, it can be divided into four parts: multi-embeddings layer, self-attention layer, BiGRU encoding layer, and CRF decoding layer.

A. Multi-Embeddings Layer

The multi-embeddings layer is responsible for more abundantly extracting semantic features of each character in the sequence from different granularities. For the i-th character, we concatenate radical-, character- and word-level embeddings to form the final character representation $\mathbf{x}_i$. It is constructed as $\mathbf{x}_i = [\mathbf{x_r}; \mathbf{x_c}; \mathbf{x_w}]$, where $\mathbf{x_r} \in \mathbb{R}^{d_r}$, $\mathbf{x_c} \in \mathbb{R}^{d_c}$, and $\mathbf{x_w} \in \mathbb{R}^{d_w}$ are radical-, character-, and word-level embeddings, respectively.

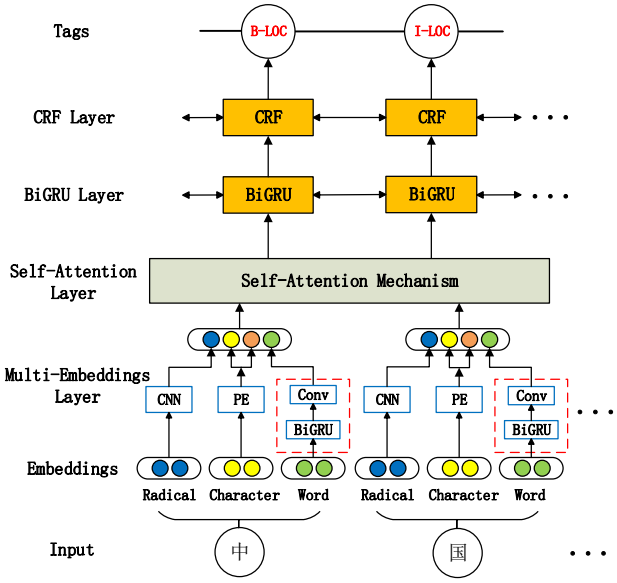

Fig. 1. Overall model architecture of A-NER. For an input sentence, we first use the multi-embeddings layer to obtain radical-, character- and word-level embeddings. These embeddings are concatenated to form the final representations of characters. Then, the final representations are fed to the self-attention layer to capture the correlations between characters. The output of the self-attention layer is used as the input of the BiGRU layer to learn contextual features. Finally, the CRF layer predicts the tag sequence.

*1) **Radical-Level Embedding**:* In Chinese linguistics, each character is semantically meaningful, thanks to its pictographic root from ancient Chinese. Intuitively, each Chinese character can be decomposed into smaller and primitive radicals, which contain inherent characteristics and linguistic information about the character itself. For example, the characters "江" and "汗" share the same radical "氵" that is a variant of Chinese character "水", indicating that they both have the meanings related to water. Meanwhile, other radicals "工" and "干" show their different spelling characteristics and semantic information. Therefore, the radical sequences of characters are useful to reflect their features in vector space.

We design a CNN to extract local context features of radical sequences of characters so that the radical-level embeddings are sensitive to the spelling of characters. The radical compositions of Chinese characters are obtained from the online Xinhua Dictionary[1]. Fig. 2 illustrates the process of obtaining the radical-level embedding $\mathbf{x_r}$ of a character. For the radical sequence $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_l)$ of each Chinese character, where l is the number of radicals, we perform convolution and max-pooling operations:

$$\mathbf{x_r} = \mathrm{Maxp}\big(\mathrm{Conv}(\mathbf{R})\big) \tag{1}$$

where $\mathbf{x_r} \in \mathbb{R}^{d_r}$. Using it not only allows us to get radical composition information about characters but also generalizes the model performance when some characters rarely or never appear.

[1]http://tool.httpcn.com/Zi/.

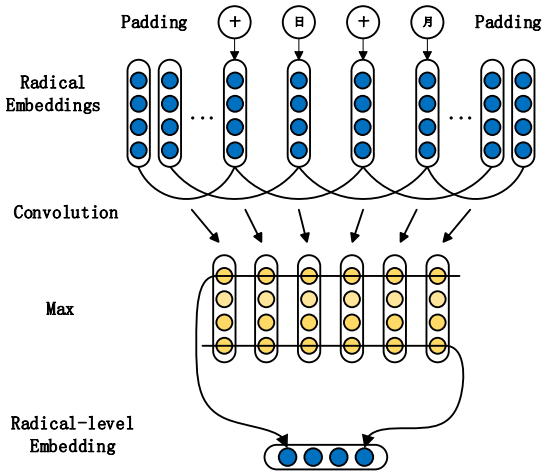

Fig. 2. The convolutional neural network for extracting radical-level context features of each Chinese character. Take the character "朝" as an example. It can be decomposed into four radicals: "十", "日", "十" and "月". Then, these initial radical embeddings are fed to CNN to obtain radical-level embedding of the character "朝".

*2) **Character-Level Embedding**:* Characters are the basic units in Chinese that can express clear meanings. Every sentence or word is composed of characters, so it is necessary to exploit the rich semantics inherent in characters. Moreover, the order of characters in a sentence is critical, while random characters are meaningless. To take advantage of the order of the sequence, we add "positional encodings" [16] to the character embeddings to inject some information about the relative or absolute position of the characters in the sequence.

For an input character sequence $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n)$, where $\mathbf{c}_i \in \mathbb{R}^{d_p}$ is the raw character embedding, n is the sequence length. We use sine and cosine functions of different frequencies to directly construct the positional encoding with the same dimension d_p:

$$PE_{2k}(pos) = \sin\left(pos/10000^{2k/d_p}\right) \tag{2}$$

$$PE_{2k+1}(pos) = \cos\left(pos/10000^{2k/d_p}\right) \tag{3}$$

where pos is the position and k is the dimension. That is, each dimension of the positional encoding corresponds to a sinusoid.

We choose the sinusoidal version because it may allow the model to extrapolate to longer sequences. Compared with the learned positional embeddings, it does not introduce additional parameters and produces almost the same results. Then, we concatenate the positional encoding to the raw character embedding as a character-level embedding: $\mathbf{x_c} = [\mathbf{c}_i; \mathbf{PE}_i]$, where $\mathbf{x_c} \in \mathbb{R}^{d_c}$, $d_c = 2d_p$.

*3) **Word-Level Embedding**:* Word is a higher-level representation of Chinese characters, embodying the linguistic characteristics and logical rules of a sentence. However, a word may contain multiple characters. To align each word and character, we copy the word by the number of characters that make up it. For example, in the word "中国", "中" and "国" are aligned with the same raw word embedding of "中国", which reflects their common context and usage scenarios.

To capture the linguistic information contained in the word sequences and reduce the word segmentation errors, we use a GRU-Conv network. It is composed of a bidirectional GRU layer and a convolution layer. First, the raw word embedding $\mathbf{w}_i$ is fed to the BiGRU layer:

$$\mathbf{M} = \mathrm{BiGRU}(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n) \tag{4}$$

The BiGRU can learn long-distance dependencies between words. Then, the convolution layer takes the output $\mathbf{M}$ as input to extract local context features:

$$\mathbf{N} = \mathrm{Conv}(\mathbf{M}) \tag{5}$$

Finally, a word-level embedding is represented as $\mathbf{x_w} = \mathbf{N}_i$, where $\mathbf{x_w} \in \mathbb{R}^{d_w}$.

B. Self-Attention Layer

Traditional embedding representation methods don't consider the correlations between characters, resulting in the information in the input sequence is not fully utilized. To solve this problem, we use a self-attention mechanism [16] to extract lexical features and semantic information deeply. This mechanism can automatically focus on specific characters that play an important role in Chinese NER and find the relationships within a sequence while ignoring useless information.

As the model processes each character (e.g., each position in the sequence), self-attention allows it to focus on other positions in the input sequence for clues that can help lead to a better encoding for this character. In other words, it combines the "understanding" of other relevant characters into the one we're currently processing.

For the output $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ of the multi-embeddings layer, the vector-matrix $\mathbf{X}$ is mapped to queries $\mathbf{Q}$, keys $\mathbf{K}$ and values $\mathbf{V}$ matrices by individually multiplying the trained weight matrices $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$. Then the $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are fed to the scaled dot-product attention function to generate the output matrix $\mathbf{X}'$ as:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\Big)\mathbf{V} \tag{6}$$

where d_k is the dimension of $\mathbf{K}$.

The first step is to calculate the relevance scores between characters by computing the dot products of the query with all keys, determining how much focus to other characters in the sentence when we encode a character. And it divides the scores by $\sqrt{d_k}$, which results in more stable gradients. Then a softmax is applied to get the weights on the values, meaning how much each character will be expressed at this position. Finally, we sum up the weighted value vectors to generate the self-attention output.

C. BiGRU Layer

After obtaining the relevance of characters by the self-attention layer, we use a BiGRU layer to learn contextual features and long-distance dependencies in the sequence. For

an input sequence $\mathbf{X}' = (\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_n)$, it returns a context sequence $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$.

The BiGRU layer consists of a forward GRU network and a backward GRU network. They are two distinct networks with different parameters. The forward GRU computes a representation $\overrightarrow{\mathbf{h}_t}$ of the left context of the sequence at every character t. Similarly, the backward GRU computes the right context representation $\overleftarrow{\mathbf{h}_t}$ from the opposite direction. The context representation of the character is formed by concatenating its left and right context representations, $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$.

D. CRF Layer

Although the hidden context vector $\mathbf{h}_t$ can be directly made independent tagging decisions for each output y_t, there are strong dependencies across output labels because of grammar rules (e.g., I-ORG cannot follow B-PER). For NER, it is beneficial to consider the correlations between labels in neighborhoods and jointly decode the best chain of labels. CRF [9] is a probability graph model that follows the Markov property, which focuses on the sentence level rather than decoding the tag separately. It has been shown that CRF can produce higher tagging accuracy in general. Therefore, we model them jointly using a CRF.

Given a generic sentence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, the score of prediction sequence $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is defined as:

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \tag{7}$$

where $\mathbf{P}$ is the matrix of scores output by the BiGRU, and P_{i,y_i} represents the score of the tag y_i of the i^{th} character. $\mathbf{A}$ is a matrix of transition scores produced by the CRF, and $A_{y_i, y_{i+1}}$ represents the transition score from tag y_i to tag y_{i+1}. Then, a softmax over all possible tag sequences $\mathbf{Y_X}$ yields a probability for the sequence $\mathbf{y}$:

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X}, \mathbf{y})}}{\sum_{\mathbf{y}' \in \mathbf{Y_X}} e^{s(\mathbf{X}, \mathbf{y}')}} \tag{8}$$

During training, the log-probability of the correct tag sequence is maximized. While decoding, we predict the optimal label sequence with the maximum score given by:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}' \in \mathbf{Y_X}} s(\mathbf{X}, \mathbf{y}') \tag{9}$$

We only model bigram interactions between outputs and adopt the Viterbi algorithm during decoding.

IV. Experiments

In this section, we show the details of our experimental datasets, settings, and results.

A. Experimental Setup

Datasets. We evaluate our model on two datasets. For the social domain, we use a standard Weibo NER dataset [13] with a lot of irregular spoken usage, which includes both named and nominal mentions. For the news domain, we experiment on a formal MSRA News dataset [11], which only has named entities. The two corpora have been separately divided into training, development, and test sets, as shown in Table I.

Table I. Statistics of Corpus Datasets

Dataset	Type	Train	Dev	Test
Weibo	Sentence	1.4k	0.27k	0.27k
	Char	73.8k	14.5k	14.8k
	Entity	1.89k	0.39k	0.42k
MSRA	Sentence	46.4k	-	4.4k
	Char	2169.9k	-	172.6k
	Entity	74.8k	-	6.2k

Pretrained Embeddings. Following the related work [6], [13], [17], we use the Jieba toolkit for word segmentation. Character embeddings and word embeddings are initialized with pre-trained embeddings provided by Tencent AI Lab [2]. Moreover, we get radical compositions of Chinese characters from the online Xinhua Dictionary. These radical embeddings and out-of-vocabulary words are all randomly initialized with a uniform distribution. We will fine-tune these initial embeddings, modifying them during the gradient update by back-propagation.

Parameter Settings. Parameter optimization is performed with a standard "Adam" algorithm. All embeddings have the same dimension, where $d_r = d_p = d_w = 200$. We set the hidden state size of GRU to 150. For CNN, we use 200 filters with window size 3. To mitigate overfitting, we apply the dropout method and early stopping to regularize our model. We fix the dropout rate at 0.5 for all dropout layers (e.g., CNN, BiGRU).

Evaluation Metrics. In this paper, we use the IOB (Inside, Outside, Beginning) tagging scheme. Standard precision (P), recall (R), and F1-score (F1) are used as evaluation metrics. Early stopping is applied in each experiment based on the loss of the development set. We run five experiments on each dataset and count the average precision, recall, and F1-score.

B. Experimental Results

This section presents the experimental results and discusses their implications.

*1) **Weibo Dataset**:* The results on the Weibo dataset are shown in Table II. Obviously, our proposed method achieves state-of-the-state performance. Compared with the best baseline [17], A-NER improves 2.97% in precision, 2.13% in recall, and 2.93% in F1-score, respectively. The reasons are as follows: (i) the multi-embeddings layer makes full use of semantic information from radical-, character- and word-level; (ii) the self-attention mechanism can effectively capture the correlations between characters. Through the above ways, we obtain richer semantic features and structural information of characters in sentences.

In the second part of Table II, we give the results of our baseline (i.e., A-NER removes the self-attention layer) and

[2]https://ai.tencent.com/ailab/nlp/embedding.html.

Table II. Weibo NER Results

Models	P	R	F1
Peng and Dredze (2015) **[13]**	63.84	29.45	40.38
Peng and Dredze (2016) **[14]**	61.64	38.55	47.43
E and Xiang (2017) **[6]**	65.29	39.71	49.47
Cao et al. (2018) **[1]**	55.72	50.68	53.08
Zhang and Yang (2018) **[19]**	53.04	62.25	58.79
Xu et al. (2019) **[17]**	**75.17**	**64.39**	**68.93**
Baseline	76.03	64.87	70.01
A-NER	**78.14**	**66.52**	**71.86**

A-NER. We observe that the self-attention mechanism can effectively improve the model performance.

*2) **MSRA Dataset**:* Table III shows the experimental results on the MSRA dataset. The existing state-of-the-art system [19] investigates a lattice-structured LSTM model to incorporate lexicon information into the neural network. Their model achieves the precision of 93.57%, recall of 92.79%, and F1-score of 93.18%. However, it uses external lexicon data, so the quality of the lexicon will affect the performance of Chinese NER due to some noise words.

Table III. MSRA NER Results

Models	P	R	F1
Chen et al. (2006) **[2]**	91.22	81.71	86.20
Zhang et al. (2006) **[18]**	92.20	90.18	91.18
Zhou et al. (2013) **[20]**	91.86	88.75	90.28
Dong et al. (2016) **[5]**	91.28	90.62	90.95
Zhang and Yang (2018) **[19]**	**93.57**	**92.79**	**93.18**
Xu et al. (2019) **[17]**	91.57	91.33	91.45
Baseline	92.34	91.87	92.10
A-NER	**93.28**	**92.16**	**92.71**

Our model performs slightly worse than the lattice-structured LSTM but outperforms all other methods. A-NER's precision, recall, and F1-score are 93.28%, 92.16%, and 92.71%, respectively. The reason is that A-NER may suffer from insufficient learning ability on such huge datasets since it is relatively simple. In general, A-NER is very effective and robust, which reaches a competitive result without any external resources and carefully designed features.

V. Conclusion

In this paper, we propose a BiGRU-CRF framework that combines self-attention with multi-embeddings to solve the challenges encountered by Chinese NER. For an input sequence, A-NER can extract semantic information about characters from multiple granularities and learn the correlations between characters. Therefore, our model obtains richer linguistic information than previous work and dose not require large amounts of task-specific knowledge. The proposed model is evaluated on different datasets and achieves state-of-the-art or competitive performance.

Acknowledgment

The research is supported by the National Key R&D Program of China 2018YFB0803400, 2018YFB2100300, National Natural Science Foundation of China under Grant No.61972369, No.61572453, No.61520106007, No.61572454, and the Fundamental Research Funds for the Central Universities, No. WK2150110009.

References

[1] Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. Adversarial transfer learning for chinese named entity recognition with self-attention mechanism. In *EMNLP*, pages 182–192, 2018.

[2] Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. Chinese named entity recognition with conditional random fields. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 118–121, 2006.

[3] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014.

[4] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12(Aug):2493–2537, 2011.

[5] 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 *Natural Language Understanding and Intelligent Applications*, pages 239–250. Springer, 2016.

[6] Shijia E and Yang Xiang. Chinese named entity recognition with character-word mixed embedding. In *CIKM*, pages 2055–2058. ACM, 2017.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[8] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[9] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[10] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL-HLT*, pages 260–270, 2016.

[11] Gina-Anne Levow. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, 2006.

[12] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, volume 1, pages 1064–1074, 2016.

[13] Nanyun Peng and Mark Dredze. Named entity recognition for chinese social media with jointly trained embeddings. In *EMNLP*, pages 548–554, 2015.

[14] Nanyun Peng and Mark Dredze. Improving named entity recognition for chinese social media with word segmentation representation learning. In *ACL*, volume 2, pages 149–155, 2016.

[15] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. Deep semantic role labeling with self-attention. In *AAAI*, pages 4929–4936, 2018.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[17] Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. Exploiting multiple embeddings for chinese named entity recognition. In *CIKM*, pages 2269–2272. ACM, 2019.

[18] Suxiang Zhang, Ying Qin, Wen-Juan Hou, and Xiaojie Wang. Word segmentation and named entity recognition for sighan bakeoff3. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 158–161, 2006.

[19] Yue Zhang and Jie Yang. Chinese ner using lattice lstm. In *ACL*, volume 1, pages 1554–1564, 2018.

[20] Junsheng Zhou, Weiguang Qu, and Fen Zhang. Chinese named entity recognition via joint identification and categorization. *Chinese Journal of Electronics*, 22(2):225–230, 2013.