

Neural Chinese Named Entity Recognition via CNN-LSTM-CRF and Joint Training with Word Segmentation

Fangzhao Wu
Microsoft Research Asia
Beijing, China
wufangzhao@gmail.com

Junxin Liu
Tsinghua University
Beijing, China
ljx16@mails.tsinghua.edu.cn

Chuhan Wu
Tsinghua University
Beijing, China
wuch15@mails.tsinghua.edu.cn

Yongfeng Huang
Tsinghua University
Beijing, China
yfhuang@tsinghua.edu.cn

Xing Xie
Microsoft Research Asia
Beijing, China
xingx@microsoft.com

ABSTRACT

Chinese named entity recognition (CNER) is an important task in Chinese natural language processing field. However, CNER is very challenging since Chinese entity names are highly context-dependent. In addition, Chinese texts lack delimiters to separate words, making it difficult to identify the boundary of entities. Besides, the training data for CNER in many domains is usually insufficient, and annotating enough training data for CNER is very expensive and time-consuming. In this paper, we propose a neural approach for CNER. First, we introduce a CNN-LSTM-CRF neural architecture to capture both local and long-distance contexts for CNER. Second, we propose a unified framework to jointly train CNER and word segmentation models in order to enhance the ability of CNER model in identifying entity boundaries. Third, we introduce an automatic method to generate pseudo labeled samples from existing labeled data which can enrich the training data. Experiments on two benchmark datasets show that our approach can effectively improve the performance of Chinese named entity recognition, especially when training data is insufficient.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Information extraction**; *Neural networks*.

KEYWORDS

Named Entity Recognition; Word Segmentation; Neural Network

ACM Reference Format:

Fangzhao Wu, Junxin Liu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2019. Neural Chinese Named Entity Recognition via CNN-LSTM-CRF and Joint Training with Word Segmentation. In *Proceedings of the 2019 World Wide Web Conference (WWW '19)*, May 13–17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3308558.3313743>

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

© 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-6674-8/19/05.

<https://doi.org/10.1145/3308558.3313743>

1 INTRODUCTION

The goal of named entity recognition (NER) is to identify entity names from texts and classify their types into different categories such as person, location and so on [9, 26]. For example, given a sentence “Bill Gates is the founder of Microsoft”, a NER tool can recognize that “Bill Gates” is a named entity of person and “Microsoft” is a named entity of organization. NER is an important task in natural language processing field [25, 28], and an essential step for many downstream applications such as entity linking [11], relation extraction [20] and question answering [8].

NER has been studied for many years and various methods have been proposed [16, 25]. Most of these methods are designed for NER of English texts. For example, Settles [29] proposed to use conditional random fields (CRF) for English medical text NER and incorporated both orthographic features and semantic features. Huang et al. [14] proposed a LSTM-CRF neural model for English NER. They combined spelling features (e.g., whether a word starts with a capital letter), context features (unigrams and bigrams), and word embeddings to build features for words. Chiu and Nichols [3] proposed to learn word- and character-level features using LSTM and CNN networks for English NER. In their method, the word features include word embeddings, character embeddings learned from characters using CNN network, and capitalization features.

Compared with NER of English texts, Chinese NER is more difficult [6, 19]. First, Chinese texts lack strong indications of entity names existing in English texts such as capitalization [27]. Second, Chinese entity names are highly context-dependent. Almost every Chinese character and word can be used in Chinese entity names. The same word can be used as entity names or non-entity words in different contexts. For example, in the sentence “文献是朝阳区区长”, “文献” is a named entity of person. However, in most cases the word “文献” is used as a non-entity word in Chinese texts with the meaning of “literature”. In addition, the same word can be used as names of different kind entities. For instance, the word “阿里” can be the name of person entity (e.g., “拳王阿里是个传奇”), location entity (“西藏阿里美不胜收”), and organization entity (“杭州阿里吸引了很多人才”). Third, different from English texts, there is no delimiter such as whitespace to separate words in Chinese texts, making it more difficult to identify entity boundaries [24]. For example, a Chinese NLP tool may segment the sentence “习近平与特朗普通电话” into “习近/平常/与/特朗/普通/电话” and

predicts that “特朗” is a person name, while in fact “习近平” and “特朗普” are correct named entities of person. Besides, annotating sufficient data for training CNER models is very expensive and time-consuming, and the training data in many domains is limited. Thus, CNER is quite challenging.

In order to handle these challenges, in this paper we propose a neural approach for Chinese named entity recognition. First, in our approach we propose a CNN-LSTM-CRF architecture to fully capture both local and long-distance contexts for CNER. Our model contains three layers, i.e., a convolutional neural network (CNN) [18] layer to learn contextual character representations from local contexts, a bidirectional long-short term memory (Bi-LSTM) [13] layer to learn contextual character representations from long-distance contexts, and a CRF layer to jointly decode character labels. In addition, we propose a unified framework to jointly train CNER and Chinese word segmentation (CWS) models in order to improve the ability of CNER model in predicting entity boundaries. In our framework, the CWS model shares the same character embeddings and CNN network with the CNER model, and has an independent CRF layer for label decoding. Besides, we propose an automatic method to generate pseudo labeled samples based on existing labeled data by randomly replacing entity names in labeled sentences with other entity names of the same concept. These pseudo labeled samples are usually correct in grammar and smooth in semantics, and can enrich the training data and improve the generalization ability of CNER model. Extensive experiments are conducted on two benchmark datasets. The experimental results show that our approach can significantly improve the performance of Chinese named entity recognition and outperform many baseline methods.

2 RELATED WORK

Chinese named entity recognition is usually modeled as a character-level sequence labeling problem [10, 22], since Chinese sentence is a string of characters and there is no explicit delimiter like whitespace to separate characters into words. Statistical sequence modeling methods such as CRF [15] are widely used in Chinese NER [24, 27]. A core step in these methods is building feature representation for each character in sentence. Traditionally, these character features are manually designed [1, 32]. For example, character unigrams, bigrams and clusters are used as character features in [1]. Zhang et al. [33] proposed to build character features using surrounding characters and whether target character is in external dictionaries. Yu et al. [32] proposed to combine character-level features, word-level features, POS tagging features and dictionary features for CNER. These handcrafted features require a large amount of domain knowledge to design and many of them rely on external resources such as gazetteers, which may not exist in many domains. Moreover, handcrafted features such as character n-grams usually cannot capture global context information. Different from above methods, our approach can learn character feature representations from data without feature engineering, and can incorporate both local and long-distance context information.

In recent years, neural network methods have been applied to English NER [14, 16]. Most of these methods are based on LSTM-CRF architecture. For example, Lample et al. [16] used Bi-LSTM to learn hidden representations of words and used CRF to do label

decoding. These methods for English NER are usually based on word-level input, and cannot be directly applied to Chinese NER which is a character-level sequence labeling task. CNN is exploited in [3] and [25] to learn word representations from characters for English NER. These word representations are combined with word embeddings and/or other features to build word features. However, there is no explicit word boundaries in Chinese texts. Thus, these CNN incorporated methods for English NER are not suitable for Chinese NER. Different from these methods, in our approach CNN is used to learn representations of characters rather than words.

Recently, neural network methods are also applied to Chinese NER [6, 7, 12]. These methods are based on LSTM-CRF framework, where LSTM is used to learn hidden representations of characters and CRF is used for joint labeling decoding. Many of these methods also incorporate handcrafted features, such as radical features [7] and position features [12], to build character representations. Different from these methods, our approach can train CNER model in an end-to-end manner and does not need any manual feature engineering. In addition, our approach is based on the CNN-LSTM-CRF architecture to fully capture both local and long-distance contexts.

Jointly training NER model with related tasks has the potential to improve the performance of NER [23, 28]. For example, Luo et al. [23] proposed to jointly train NER and entity linking models, and achieved state-of-the-art performance on English NER. Since Chinese texts lack word delimiters and identifying entity boundaries is very challenging in CNER, we propose to jointly train CNER model with word segmentation to improve the ability of CNER model in identifying entity boundaries. Similar with our approach, Peng et al. [28] proposed to jointly train CNER and CWS models in domain adaptation scenario. In their method, CNER and CWS share the same character representation model and have different label decoding models. In our approach we regard word segmentation as an auxiliary task and use it to learn word boundary aware character representations for CNER. Experiments show that our approach is more effective. In addition, we propose to automatically generate pseudo labeled samples from labeled data, which to our best knowledge has not been explored in existing CNER methods.

3 OUR APPROACH

In this section we introduce our approach for CNER in detail. First, we present the CNN-LSTM-CRF neural architecture for CNER. Next, we introduce a unified framework to jointly train CNER and word segmentation models. Finally, we introduce an automatic method to generate pseudo labeled samples from existing labeled data.

3.1 CNN-LSTM-CRF Architecture for CNER

Our proposed CNN-LSTM-CRF neural architecture for Chinese named entity recognition is illustrated in Fig. 1. Next we introduce each layer from bottom to top in detail.

The first layer is character embedding, which aims to convert a sentence from a sequence of characters into a sequence of dense vectors. In this layer, an embedding matrix $E \in \mathcal{R}^{D \times V}$ is used to map each character into a dense vector, where D is embedding dimension and V is vocabulary size. Denote an input sentence as $s = [w_1, w_2, \dots, w_N]$, where N is sentence length and $w_i \in \mathcal{R}^V$ is the

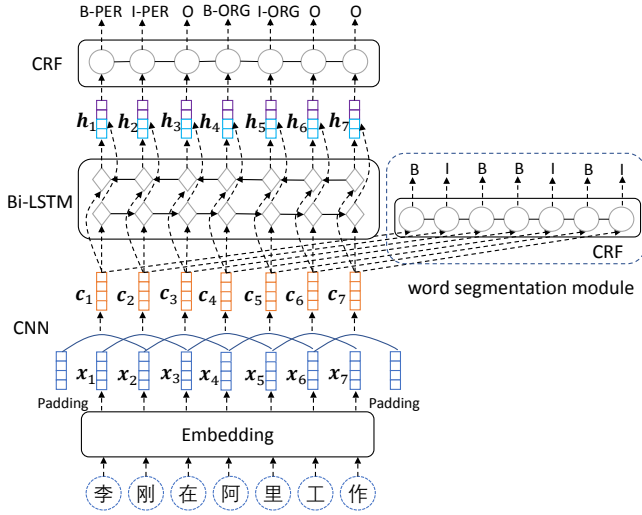


Figure 1: The framework of our approach.

one-hot representation of the i_{th} character. The output of this layer is a sequence of vectors $[x_1, x_2, \dots, x_N]$, where $x_i = Ew_i \in \mathcal{R}^D$.

The second layer is a CNN network. CNN has been widely used in computer vision field to extract local information of images [17]. The local context of sentences is also very important for Chinese named entity recognition. For example, the word “阿里” is a location entity in “西藏阿里”, while is a organization entity in “杭州阿里”. Motivated by these observations, we propose to use CNN to capture the local context information for CNER. Denote $w \in \mathcal{R}^{KD}$ as a filter in CNN where K is the window size, then the contextual representation of the i_{th} character learned by this filter is:

$$c_i = f(w^T \times x_{[i-\frac{K-1}{2}]:[i+\frac{K-1}{2}]}), \quad (1)$$

where $x_{[i-\frac{K-1}{2}]:[i+\frac{K-1}{2}]}$ represents the concatenation of the embeddings of characters from $[i - \frac{K-1}{2}]$ to $[i + \frac{K-1}{2}]$, and f is the activation function. Here we select ReLU [17] as the activation function. In this layer, we use multiple filters with different window sizes (ranging from 2 to 5) to learn contextual character representations. Denote the filter number as M . The contextual representation of the i_{th} character (denoted as c_i) is the concatenation of the outputs of all filters at this position. The output of the CNN layer is $c = [c_1, c_2, \dots, c_N]$, where $c_i \in \mathcal{R}^M$.

The third layer is a Bi-LSTM network. LSTM is a special type of recurrent neural networks (RNN) which can capture long-distance sequence information and is powerful in modeling sequential data. LSTM can generate hidden states for tokens in a sequence using all previous contexts, which is beneficial for Chinese named entity recognition. For example, in sentence “阿里是中国非常著名的IT企业” (Alibaba is a famous Chinese IT company), although “IT企业” has a long distance with “阿里”, it is very important for inferring that “阿里” is an organization entity. In addition, both left and right contexts can be useful for recognizing named entities. Thus, we use bidirectional LSTM (Bi-LSTM) to learn hidden representations of characters from global contexts. Denote \overrightarrow{lstm} as

the LSTM scanning text from left to right, and \overleftarrow{lstm} as the LSTM scanning text from right to left, then the hidden representations learned by these LSTMs can be represented as $[\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N] = \overrightarrow{lstm}([c_1, c_2, \dots, c_N])$ and $[\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_N] = \overleftarrow{lstm}([c_1, c_2, \dots, c_N])$. The hidden representation of the i_{th} character is a concatenation of \vec{h}_i and \overleftarrow{h}_i , i.e., $h_i = [\vec{h}_i, \overleftarrow{h}_i]$. The output of the Bi-LSTM layer is $h = [h_1, h_2, \dots, h_N]$, where $h_i \in \mathcal{R}^{2S}$ and S is the dimension of hidden states in LSTM.

The fourth layer in our model is conditional random field (CRF). In NER, neighboring labels usually have strong dependencies [25]. For example, *I-PER* label usually follows *B-PER* and *I-PER*, but it cannot follow *B-ORG* or *I-ORG*. Thus, it is beneficial to jointly decode the labels of characters in a sequence rather than decode them independently [4]. In this paper, we use first-order linear chain CRF [15] to jointly decode the labels of characters.

Denote $y = [y_1, y_2, \dots, y_N]$ as the label sequence of sentence s , where $y_i \in \mathcal{R}^L$ is the one-hot representation of the i_{th} character’s label, and L is the number of labels. The input of the CRF layer is the hidden representations of characters generated by the Bi-LSTM layer, i.e., $h = [h_1, h_2, \dots, h_N]$, and the output of the CRF layer is the label sequence y . CRF is a probabilistic model. The conditional probability of label sequence y given input h is computed as follows:

$$p(y|h; \theta) = \frac{\prod_{i=1}^N \psi(h_i, y_i, y_{i-1})}{\sum_{y' \in \mathcal{Y}(s)} \prod_{i=1}^N \psi(h_i, y'_i, y'_{i-1})}, \quad (2)$$

where $\mathcal{Y}(s)$ is the set of all possible label sequences of sentence s , θ is the parameter set, and $\psi(h_i, y_i, y_{i-1})$ is the potential function. In our approach, the potential function is:

$$\psi(h_i, y_i, y_{i-1}) = \exp(y_i^T W^T h_i + y_{i-1}^T T y_i), \quad (3)$$

where $W \in \mathcal{R}^{2S \times L}$ and $T \in \mathcal{R}^{L \times L}$ are parameters of the CRF layer, and $\theta = \{W, T\}$ in Eq. (2).

The loss function of our approach is the negative log-likelihood over all training samples, which can be formulated as follows:

$$\mathcal{L}_{NER} = - \sum_{s \in S} \log(p(y_s | h_s; \theta)), \quad (4)$$

where S is the set of sentences in training data, h_s and y_s are the hidden representations and label sequence of sentence s .

3.2 Joint Training with Word Segmentation

The NER task can be regarded as a combination of two sub-tasks: extracting entity names from texts (i.e., identifying the boundaries of entity names) and classifying their types. Identifying the boundaries of Chinese entity names is quite challenging, since there is no explicit word delimiters in Chinese texts. For example, many existing Chinese NLP tools such as LTP predict that “特朗” is an entity name in the sentence “习近平常与特朗普通电话”, while the correct entity name is “特朗普”.

In Chinese natural language processing field, the goal of Chinese word segmentation (CWS) is segmenting Chinese texts into words (in other words, predicting the boundaries of words in texts). Thus, CWS is highly related to CNER and has the potential to help CNER predict entity boundaries more accurately. Motivated

by this observation, we propose to enhance the ability of CNER model in identifying entity boundaries by jointly training CNER and CWS models. Similar with CNER, CWS is also usually modeled as a character-level sequence labeling problem [2, 10, 21], and CRF is widely used in CWS methods. Our unified framework for jointly training CNER with CWS is shown in Fig. 1. In our framework the CNER and CWS models share the same character embeddings and CNN network. In this way, the useful information in word segmentation can be encoded to learn word boundary aware contextual character representations, which is useful for predicting entity boundaries. The loss function of the CWS module is:

$$\mathcal{L}_{CWS} = - \sum_{s \in S} \log(p(\mathbf{y}_s^{seg} | \mathbf{c}_s; \theta^{seg})), \quad (5)$$

where \mathbf{y}_s^{seg} is the label sequence of sentence s for word segmentation, and θ^{seg} is the parameter set of CWS model. \mathbf{c}_s is the hidden character representations of sentence s output by the CNN network.

The final objective function of our approach is the combination of the CNER loss and the CWS loss as follows:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{NER} + \lambda\mathcal{L}_{CWS}, \quad (6)$$

where $\lambda \in [0, 1)$ is a coefficient to control the relative importance of the CWS loss in the overall loss.

3.3 Pseudo Labeled Data Generation

The labeled data for Chinese named entity recognition is usually scarce in many domains. Manually annotating sufficient samples for CNER is time-consuming and expensive. Thus, an automatic method to generate labeled samples for CNER will be very useful. In this paper we propose to automatically generate pseudo labeled samples based on existing labeled data. Our method is motivated by the observation that if an entity name in a sentence is replaced by another entity name with the same concept, then the new sentence is usually correct in grammar and semantics. For example, a Chinese sentence may be “李刚在阿里工作”, where “李刚” is a person entity and “阿里” is a company entity. If “王小超” is another person name and “谷歌” is another company name, then we can obtain a new sentence by replacing “李刚” with “王小超” and “阿里” with “谷歌”, i.e., “王小超在谷歌工作”. If we have the NER labels of the original sentence, then we can easily infer the NER labels of the pseudo sentence. For instance, if the NER labels of the original sentence in aforementioned example are “B-PER/I-PER/O/B-ORG/I-ORG/O/O”, then the NER labels of the pseudo sentence are “B-PER/I-PER/I-PER/O/B-ORG/I-ORG/O/O”. In addition, if we have CWS labels of the original sentence, then we can also automatically obtain CWS labels of the pseudo sentence, since entity names are usually regarded as independent words. For instance, in aforementioned example, if the CWS labels of the original sentence are “B/I/B/B/I/B/B/I”, then the CWS labels of the pseudo sentence are “B/I/I/B/B/I/B/B/I”.

In our method, given a set of labeled samples for CNER, we first extract all the entity names (denoted as EN) from them. Then we randomly select a labeled sentence, and replace each entity in this sentence with an entity randomly sampled from EN which has the same concept. In this way a pseudo sentence is generated and its NER labels (as well as CWS labels) can be automatically inferred.

This step is repeated multiple times until a predefined number of pseudo labeled samples are generated.

4 EXPERIMENTS

4.1 Datasets and Experimental Settings

Two benchmark datasets for Chinese named entity recognition are used in our experiments. The first one is the MSRA corpus released by the third SIGHAN Chinese language processing bakeoff¹ (denoted as *Bakeoff-3*). It contains 46,364 sentences for training and 4,365 sentences for test. The second dataset is the MSRA corpus released by the fourth SIGHAN Chinese language processing bakeoff² (denoted as *Bakeoff-4*). This dataset contains 23,181 training sentences and 4,636 test sentences. Both datasets contain three types of named entities, i.e., *person* (denoted as *PER*), *location* (denoted as *LOC*) and *organization* (denoted as *ORG*). The statistics of these datasets are summarized in Table 1.

Table 1: The statistics of datasets. #Sen and #Ent represent the numbers of sentences and entities.

Dataset	Training		Test		
	#Sen	#Ent	#Sen	#Ent	OOV Rate
<i>Bakeoff-3</i>	46,364	74,528	4,365	6,190	28.16%
<i>Bakeoff-4</i>	23,181	37,811	4,636	7,707	21.42%

Following [31], we use *BIO* tagging scheme for CNER, where *B*, *I* and *O* respectively represent the beginning, inside and outside of entity name. The labels *B* and *I* are further combined with entity types, such as “B-PER” and “I-LOC”. As for CWS, we use the *BI* tagging scheme. In the *Bakeoff-3* dataset the annotation of word segmentation in training set is provided, while in the *Bakeoff-4* dataset it is not provided. Thus, we used LTP³ tool to segment the sentences in the training set of the *Bakeoff-4* dataset. Note that the word segmentation information is not involved in the test data.

In our experiments, the embeddings of characters are pretrained on the Sogou News Corpus⁴ using the word2vec tool⁵. The dimension of character embedding is 200. These embeddings are tuned during model training. Since dropout method [30] is an effective way to mitigate overfitting, we apply it to the outputs of embedding, CNN and LSTM layers. Our approach is implemented using Tensorflow library, and RMSProp [5] is used as the optimization algorithm to learn model parameters. Hyperparameters are selected according to cross-validation on training data. In detail, the number of filters in CNN network is 400, the hidden state size of LSTM is 200, the dropout rate is 0.2, and the λ in Eq. (6) is 0.4. In each experiment, we randomly sample 10% of the training data for validation and the remaining for model training. Each experiment is repeated 10 times, and the average results are reported.

¹<http://sighan.cs.uchicago.edu/bakeoff2006/download.html>

²<https://www.aclweb.org/mirror/ijcnlp08/sighan6/chinesebakeoff.htm>

³<https://www.ltp-cloud.com/intro/en/>

⁴<http://www.sogou.com/labs/resource/ca.php>

⁵<https://code.google.com/archive/p/word2vec/>

Table 2: The performance of different methods on the *Bakeoff-3* dataset. P , R and F represent precision, recall and Fscore respectively. R_{oov} stands for the recall of out-of-vocabulary entities.

	5%				25%				100%			
	P	R	F	R_{oov}	P	R	F	R_{oov}	P	R	F	R_{oov}
<i>LSTM+Softmax</i>	56.14	49.83	52.68	33.03	73.02	66.94	69.84	49.00	82.76	78.52	80.59	60.57
<i>LSTM+CRF</i>	61.00	54.32	57.37	37.09	77.75	70.42	73.90	53.05	87.30	82.93	85.06	68.06
<i>MDA</i>	65.12	57.90	61.27	42.74	79.71	73.16	76.29	58.34	87.78	82.43	85.02	67.15
<i>LSTM+CRF+Pseudo</i>	61.21	57.25	59.07	40.22	79.57	75.80	77.63	60.45	88.13	85.74	86.92	73.19
<i>CNN+LSTM+Softmax</i>	62.72	56.96	59.62	39.75	77.83	74.54	76.14	57.15	87.60	83.50	85.50	67.41
<i>CLC</i>	65.58	61.60	63.45	43.04	81.22	76.20	78.59	59.23	90.39	86.43	88.37	71.54
<i>CLC+Joint</i>	70.70	62.93	66.56	45.78	81.20	78.32	79.72	63.13	90.29	86.98	88.60	73.61
<i>CLC+Pseudo</i>	69.86	64.13	66.84	49.17	82.69	79.93	81.27	65.38	89.59	88.46	89.02	76.99
<i>CLC+Joint+Pseudo</i>	70.31	66.51	68.33	49.30	83.57	81.62	82.57	66.86	89.48	89.36	89.42	77.93

Table 3: The performance of different methods on the *Bakeoff-4* dataset.

	5%				25%				100%			
	P	R	F	R_{oov}	P	R	F	R_{oov}	P	R	F	R_{oov}
<i>LSTM+Softmax</i>	51.78	42.88	46.74	30.24	69.32	67.16	68.18	49.62	83.39	80.31	81.81	63.08
<i>LSTM+CRF</i>	54.18	47.53	50.48	30.64	74.90	71.64	73.21	56.07	87.10	85.07	86.07	69.96
<i>MDA</i>	56.66	49.75	52.90	34.93	76.29	73.91	75.05	59.09	87.56	85.01	86.26	70.77
<i>LSTM+CRF+Pseudo</i>	57.37	49.54	53.09	34.53	76.31	75.79	76.02	59.85	86.94	86.90	86.92	72.07
<i>CNN+LSTM+Softmax</i>	59.09	51.12	54.74	35.81	76.42	73.74	74.99	58.13	88.48	86.34	87.40	69.47
<i>CLC</i>	58.42	53.56	55.83	39.41	78.84	78.50	78.59	61.82	89.86	88.74	89.30	73.97
<i>CLC+Joint</i>	61.88	56.23	58.76	41.49	82.87	78.63	80.65	63.43	90.23	88.80	89.51	74.86
<i>CLC+Pseudo</i>	61.82	59.69	60.67	43.40	82.33	80.41	81.35	64.00	90.44	89.80	90.12	75.00
<i>CLC+Joint+Pseudo</i>	64.21	61.58	62.78	47.58	83.16	81.93	82.52	66.88	90.09	90.28	90.18	77.13

4.2 Performance Evaluation

First we compare the performance of our approach with different neural network based CNER methods, including: 1) *LSTM+CRF*, using Bi-LSTM to learn character representations and CRF to decode labels [28]; 2) *LSTM+Softmax*, using Bi-LSTM to learn character representations and softmax for label decoding; 3) *MDA*, multi-task domain adaptation method proposed by Peng et al. [28], which jointly trains CNER and CWS models via multi-task learning; 4) *LSTM+CRF+Pseudo*, *LSTM+CRF* trained on both original and pseudo labeled samples; 5) *CLC*, our CNN-LSTM-CRF architecture for CNER; 6) *CNN+LSTM+Softmax*, replacing the CRF layer in *CLC* with a softmax layer; 7) *CLC+Joint*, our unified framework for jointly training CNER and CWS models; 8) *CLC+Pseudo*, *CLC* trained on both original and pseudo labeled samples; 9) *CLC+Joint+Pseudo*, *CLC* with both joint training and pseudo labeled data.

We conducted experiments on different ratios (i.e., 5%, 25% and 100%) of training data to test the performance of different methods with insufficient and sufficient labeled samples. For those methods which involve pseudo labeled samples, the number of pseudo samples is the same with the real labeled samples. The experimental results on the *Bakeoff-3* and *Bakeoff-4* datasets are shown in Tables 2 and 3 respectively. We have several findings from the results.

First, our CNN-LSTM-CRF architecture (*CLC*) performs better than the popular *LSTM-CRF* architecture [28] on Chinese NER. For

example, on the *Bakeoff-3* dataset, the improvement of *CLC* over *LSTM-CRF* in terms of Fscore is 6.08% on 5% training data and 3.31% on 100% training data. These results validate that our CNN-LSTM-CRF architecture is more suitable for Chinese NER than LSTM-CRF. This is because our model can capture both local and long-distance contexts of sentences to learn contextual character representations, which is useful for Chinese NER.

Second, joint label decoding via CRF is better than independent label decoding for Chinese NER. For instance, *CLC* performs much better than *CNN+LSTM+Softmax*, and *LSTM+CRF* can consistently outperform *LSTM+Softmax*. This is because there are strong dependencies between neighboring NER labels, and CRF can effectively capture these dependencies. In addition, *CNN+LSTM+Softmax* consistently outperforms *LSTM+Softmax*, although they both use independent label decoding. It further validates better character representations can be learned by combining CNN and LSTM to capture both local and long-distance contexts.

Third, by jointly training CNER and CWS models using our unified framework, our approach (i.e., *CLC+Joint*) can achieve better performance, especially when training data is insufficient. This is because there is inherent relatedness between CNER and CWS, since CNER model needs to identify entity boundaries and the goal of CWS is to identify word boundaries. Experimental results show that our unified framework is effective in exploiting the inherent

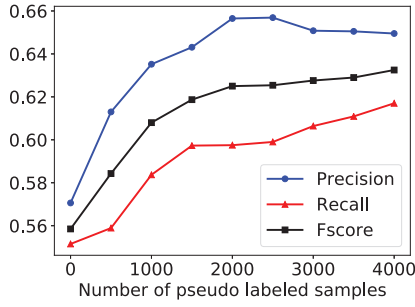


Figure 2: The influence of pseudo labeled data size.

relatedness between CNER and CWS, and can improve the ability of CNER model in identifying entity boundaries via jointly training CNER and CWS models. *MDA* method [28] also combines CNER and CWS. Our *CLC+Joint* approach can consistently outperform *MDA*. There are two major differences between our approach and *MDA*. First, our approach is based on the CNN-LSTM-CRF architecture while *MDA* is based on LSTM-CRF. Second, in *MDA* method CNER and CWS are regarded as equally important and share the same character representations, while in our approach CWS is regarded as an auxiliary task and used to learn word boundary aware character representations from local contexts. These results indicate our approach is more suitable to exploit the useful information in word segmentation for CNER.

Fourth, the automatically generated pseudo labeled samples can effectively improve the performance of our approach and baseline methods. For example, *CLC+Pseudo* significantly outperforms *CLC*, and *LSTM+CRF+Pseudo* consistently outperforms *LSTM+CRF*, especially in terms of OOV recall. Besides, the performance improvement becomes bigger when training data is insufficient. These results validate that the pseudo labeled samples automatically generated from labeled data can provide useful information for training CNER model. They can improve the generalization ability of CNER model by encouraging it to capture the contextual patterns of entity names, which is important for recognizing OOV entities. Moreover, according to Tables 2 and 3, after incorporating both pseudo labeled samples and joint training with CWS, the performance of our approach (*CLC+Joint+Pseudo*) can be further improved.

4.3 Influence of Pseudo Labeled Samples

Next we explore the influence of the number of automatically generated pseudo labeled samples on the performance of our approach. We conducted experiments on both datasets and the results on the *Bakeoff-4* dataset with 5% training data are shown in Figure 2. The results on the *Bakeoff-3* dataset show similar patterns.

According to Figure 2, as the number of pseudo labeled samples increases, the recall and Fscore of our approach consistently improve, while the precision first increases and slightly declines afterwards. These results show that the pseudo labeled samples automatically generated from labeled data can provide useful information for training CNER model and can improve its ability

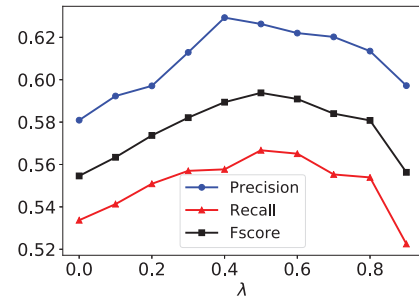


Figure 3: The influence of λ value.

in capturing contextual patterns of entity names, which is beneficial for identifying OOV entities. Thus, the recall of our approach continuously improves when more pseudo labeled samples are incorporated. However, when too many pseudo labeled samples are used, the CNER model may overfit these patterns, resulting in the slight decline of precision.

4.4 Influence of Hyperparameters

In this section we conduct experiments to explore the influence of hyperparameters on the performance of our approach. Due to space limit, here we only show the influence of the most important hyperparameter on our approach, i.e., λ in Eq. (6). The results are summarized in Figure 3. This experiment was conducted on the *Bakeoff-4* dataset with 5% training data. λ is used to control the relative importance of CWS loss in the overall loss of our joint training framework (Eq. (6)). According to Figure 3, the performance of our approach improves when λ increases from 0, and declines when λ becomes too large. This is because when λ is too small, the useful information in word segmentation is not fully exploited. Thus, the performance is not optimal. When λ becomes too large, the auxiliary CWS task is over-emphasized and the loss of NER is not fully respected. Thus, the performance is also not optimal. A moderate value of λ is most suitable for our approach.

5 CONCLUSION

In this paper we propose a neural approach for Chinese named entity recognition. First, we propose a CNN-LSTM-CRF neural architecture for Chinese NER, where CNN and Bi-LSTM networks are used to learn character representations from both local and long-distance contexts, and CRF is used for joint label decoding. Second, we propose a unified framework to jointly train Chinese named entity recognition and word segmentation models to exploit the inherent relatedness between these two tasks and enhance the ability of Chinese NER model in identifying entity boundaries. In addition, we propose an automatic method to generate pseudo labeled samples from existing labeled data to enrich the training data. Extensive experiments on two benchmark datasets validate that our approach can effectively improve the performance of Chinese named entity recognition especially when training data is insufficient, and can consistently outperform many baseline methods.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China under Grant number 2018YFC1604002, and the National Natural Science Foundation of China under Grant numbers U1705261, U1536201, and U1536207.

REFERENCES

- [1] Aitao Chen, Fuchun Peng, Roy Shan, and Gordon Sun. 2006. Chinese named entity recognition with conditional probabilistic models. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*. 173–176.
- [2] Xinchu Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for chinese word segmentation. In *EMNLP*. 1197–1206.
- [3] Jason P. C. Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *TACL* 4 (2016), 357–370.
- [4] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.
- [5] Yann Dauphin, Harm de Vries, and Yoshua Bengio. 2015. Equilibrated adaptive learning rates for non-convex optimization. In *NIPS*. 1504–1512.
- [6] Chuanhai Dong, Huijia Wu, Jiajun Zhang, and Chengqing Zong. 2017. Multi-channel LSTM-CRF for Named Entity Recognition in Chinese Social Media. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. 197–208.
- [7] Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition. In *International Conference on Computer Processing of Oriental Languages*. 239–250.
- [8] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. In *ACL*. 260–269.
- [9] Cicero dos Santos and Victor Guimarães. 2015. Boosting Named Entity Recognition with Neural Character Embeddings. In *Proceedings of the Fifth Named Entity Workshop*. 25–33.
- [10] Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics* 31, 4 (2005), 531–574.
- [11] Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In *SIGIR*. 765–774.
- [12] Hangfeng He and Xu Sun. 2017. A Unified Model for Cross-Domain and Semi-Supervised Named Entity Recognition in Chinese Social Media. In *AAAI*. 3216–3222.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [14] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).
- [15] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*. 282–289.
- [16] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *NAACL*. 260–270.
- [17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.
- [18] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1, 4 (1989), 541–551.
- [19] Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*. 108–117.
- [20] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*. 2124–2133.
- [21] Junxin Liu, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2018. Neural Chinese Word Segmentation with Dictionary Knowledge. In *NLPCC*. 80–91.
- [22] Zhangxun Liu, Conghui Zhu, and Tiejun Zhao. 2010. Chinese named entity recognition with a sequence labeling approach: based on characters, or based on words? In *Advanced intelligent computing theories and applications with aspects of artificial intelligence*. Springer, 634–640.
- [23] Gang Luo, Xiaojian Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *EMNLP*. 879–888.
- [24] Wencan Luo and Fan Yang. 2016. An Empirical Study of Automatic Chinese Word Segmentation for Spoken Language Understanding and Named Entity Recognition. In *NAACL*. 238–248.
- [25] Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF. In *ACL*. 1064–1074.
- [26] Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura, and Tomoko Ohkuma. 2017. Character-based Bidirectional LSTM-CRF with words and characters for Japanese Named Entity Recognition. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*. 97–102.
- [27] Nanyun Peng and Mark Dredze. 2015. Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings. In *EMNLP*. 548–554.
- [28] Nanyun Peng and Mark Dredze. 2017. Multi-task Domain Adaptation for Sequence Tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. 91–100.
- [29] Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*. 104–107.
- [30] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15, 1 (2014), 1929–1958.
- [31] Chunqi Wang, Wei Chen, and Bo Xu. 2017. Named Entity Recognition with Gated Convolutional Neural Networks. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. 110–121.
- [32] Xiaofeng Yu, Wai Lam, Shing-Kit Chan, Yiu Kei Wu, and Bo Chen. 2008. Chinese NER Using CRFs and Logic for the Fourth SIGHAN Bakeoff. In *IJCNLP*. 102–105.
- [33] Yuejie Zhang, Zhiting Xu, and Tao Zhang. 2008. Fusion of multiple features for chinese named entity recognition based on CRF model. In *Asia Information Retrieval Symposium*. 95–106.