ISBN 978-981-14-1684-2
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
(WCSE 2019 SUMMER)
Hong Kong, 15-17 June, 2019, pp. 337-344
doi: 10.18178/wcse.2019.06.051

Semi-supervised Chinese Named Entity Recognition with ELMo

Su Zhang, Wenxin Hu, and Jun Zheng[+]

East China Normal University, China

Abstract. Named entity recognition is a subtask of information extraction. In general, the task of named entity recognition is to identify three main categories, including entity, time and numeric class. In Chinese named entity recognition, ambiguity and out-of-vocabulary often occurs as tricky problems, but traditional character-based and word-based model do not fix it. In this paper, we propose a semi-supervised approach by taking pre-trained embeddings from language models (ELMo) as additional embedding of word embedding. Our method could catch deep contextualized word representation, which is capable to represent lexical ambiguity in different contexts and complexity of vocabulary usage, such as grammar and semantics, by this way we are capable to identify more precise content and label the word with limited labeling data. Experiments on MSRA show that our model outperforms both word-based and character-based LSTM baselines, achieving the best results.

Keywords: Named entity recognition, Neural networks, semi-supervised, language model, character-based, Embedding of language model.

1. Introduction

Named Entity Recognition (NER) is a fundamental task which automatically detects special named entities in corpus such as people, organizations, location names, time, event for natural language processing applications, such as information extraction, question answering, knowledge graph construction and so on.

Pre-trained word embedding is significant in many neural language model because of their simplicity and efficacy. Previous studies have shown that they capture useful semantic and syntactic information[1][2] and including them in NLP systems has been shown to be extremely helpful for a variety of downstream tasks.[3]

However, traditional word type embeddings mean each token is assigned a representation which is a function of the entire input sentence. It is essential to represent not only the meaning of a word, but also the word in context. Generally, words with similar contexts should have similar semantics. Accordingly, current neural models typically include a bidirectional recurrent neural network that encodes token sequences into a context sensitive representation before making token specific predictions[3][4][5]. Elmo (Embeddings from Language Models) learns a linear combination of the vectors stacked above each input word for each end task, which markedly improves performance over just using the top LSTM layer[7].

The pre-trained word embedding is used for semi-supervised NER, which trains embedding on a large, unlabeled corpus to learn complicated representations of context by using neural language model. Previous work has explored semi-supervised based methods for using a neural language model (LM) embedding to calculate probability of future words in English corpus and shows that the context sensitive representation captured in the LM embeddings is useful in the supervised sequence tagging setting[8]. Elmo learns internal state of deep bidirectional language model to represent semantic and syntactic information of words in context[7]. But previous works did not combine the structure of the neural network and the language model,

[+] Corresponding author. Tel.: +13524209333 Fax.: + 021-62576192.
E-mail address: jzheng@cc.ecnu.edu.cn

so it could not effectively concatenate the word embedding and embedding from language model to generate deep word representation.

In this paper, we propose a semi-supervised method for Chinese named entity recognition. The following is the work we do: 1.We enhance the representation of word embedding. Embedding from language model (ELMo) is taken as additional embedding to expand the representation of word embeddings. Specifically, in order to enhance the representation of word embedding we concatenate ELMo with ordinary embedding or hidden output information of BiLSTM-CRF model. 2.We use the Adam optimizer to train neural networks. We use the Adam optimization algorithm to minimize the mean square error (MSE) loss function on the training data[18]. The Adam optimization algorithm subtly combines the advantages of the AdaGrad optimization algorithm with the RMSProp optimization algorithm. The update step size can be calculated by taking into account the mean of the uncentralized variance and gradient of the gradient.

Our contributions is we could catch deep contextualized word representation, which is capable to represent lexical ambiguity in different contexts and complexity of vocabulary usage, such as grammar and semantics, by this way we are capable to identify more precise content and label the word with limited labeling data. Our experiments demonstrate that our model works well and in our model we get a 91.89% F1-score, which is higher compared to the character-based baseline.

The remainder of this paper is organized as follows. We first summarize the related work in Section 2. In Section 3, We describe the details of our proposed model. In Sections 4, we introduce the experimental settings and results. Finally, the conclusions and future work are presented in Section 5.

2. Related work

In this section we will summarize various previous solutions for NER.

Neural network for NER. Hammerton attempted to solve the problem using a unidirectional LSTM[9], which was among the first neural models for NER. Collobert used a CNN-CRF structure, obtaining competitive results to the best statistical models[10]. Dos Santos used character CNN to augment a CNN-CRF model[11]. Most recent work leverages an LSTM-CRF architecture. Huang used hand-crafted spelling features[12]; Ma and Hovy and Chiu and Nichols used a character CNN to represent spelling characteristics[13]; Lample used a character LSTM instead[6]. Character-based sequence labeling has been the appropriate approach for Chinese NER[14]. Clark used neural networks to do Chinese word segmentation and part-of-speech tagging[15]. The author automatically acquired task-related features through deep networks, thereby avoiding the use of specific tasks or hand-designed feature engineering. LSTM instead of neural networks. Hidden layer improved the long-term dependency problem that traditional neural networks cannot solve. Dong used Chinese hieroglyphics as the rule of character embedding[16].The decoding part used greedily decoding tag from left to right and (CJ Brame)labeling data that is difficult to identify categories by active learning[17]. Previous works tried to train various kinds of neural networks but the model need big train data to obtain information of word and could not make good use of context representation.

Semi-supervised approach for NER. Several Semi-supervised neural architectures had previously been proposed for English NER. The Context2vec learned context embedding representation method which computed by using an encoder of the unsupervised language model[12].Our baseline Semi-supervised system takes a similar structure to this line of work. Peters using a large unsupervised corpus and trained a bidirectional language model to extract the sequence representation which concatenate the embedding of the Bi-RNN into the CRF layer[8].Each word embedding as context representation which is derived from computing the internal state of a two-layer bidirectional language model (LM). We use biLM to obtain word representation from large corpus, by which we better leverage word information for Chinese NER. Previous works only learned lots of content to generate embedding from language model but did not combine embedding of language model into named-entity recognition model effectively.

3. Model

We follow the model of previous semi-supervised NER work which works well on English data set, and we consider the method concatenate embedding of input and additional embedding is reasonable[8]. Our main network structure consists traditional LSTM-CRF model and language model, which is illustrated in Figure 1. The model uses a Bi-directional Long Short-Term Memory(BiLSTM) to encode characters. Then we extract word embedding and concatenate hidden output information with ELMo(embedding form language model) for every token in a given input sequenceand. Finally, conditional random field as a decoder to get the posterior probability of corresponding named entity.

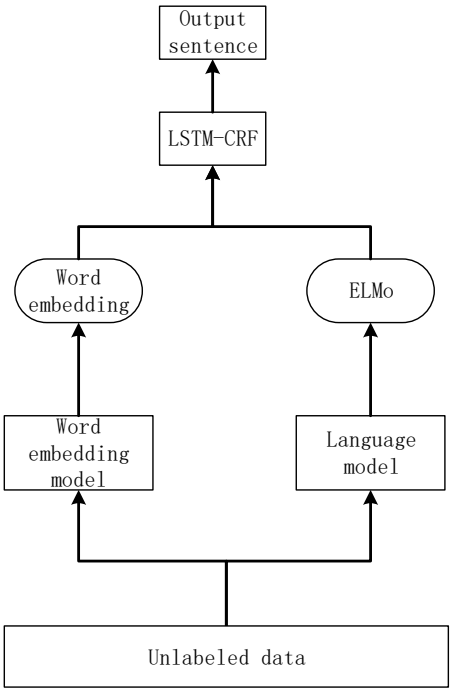

Fig. 1: The main network structure, the ELMo(embedding from language model) is used to enhance the input token representation in a traditional NER model.

3.1. LSTM

Long Short-Term Memory(LSTM) takes into account the order of words in sentence, and Bi-directional Long Short-Term Memory is capable to encode the information from the back to the front, which can better capture the double-sided semantic dependence.

LSTMs [18] are improved vision of RNNs designed to cope with these gradient vanishing problems by incorporate a memory-cell. Primarily, a LSTM unit is composed of input gate, output gate and forget gate which control the proportions from previous state to forget, and the proportion of the input to pass on the memory cell.

Our following implementation is:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \quad (1)$$

$$f_t = \sigma\left(W_f h_{t-1} + U_f x_t + b_f\right) \quad (2)$$

$$\widetilde{c_t} = \tanh(W_c h_{t-1} + U_c x_t + b_c) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \quad (4)$$

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \quad (5)$$

$$h_t = o_t \tanh \odot (c_t) \quad (6)$$

where σ is the element-wise sigmoid function and $\odot$ is the element-wise product. x_t is the input vector (e.g. word embedding) at time t, h_t is the hidden state (also called output) vector storing all the useful information at time t, and b are biases.

For many sequence labeling tasks, we get the context vector of a characters using a bidirectional LSTM. For a given sentence $(x_1, x_2, \ldots, x_n)$ containing n characters and each character represented as a d-

dimensional vector, a LSTM computes hidden state $\overrightarrow{h_t}$ takes information from past and $\overleftarrow{h_t}$ takes information from future. We can get another LSTM which achieves the right context information. We refer to the former as the forward LSTM and the latter as the backward LSTM. Then the two hidden states are concatenated to obtain the final representations, $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$.

In addition, we use the Adam optimizer to train the network, setting the first-order momentum attenuation coefficient (ρ_1) to 0.9, the first-order momentum attenuation coefficient (ρ_2)to 0.999. Adam calculates independent adaptive learning rates for different parameters by calculating the first-order moment estimation and second-order moment estimation of the gradient.

3.2. CRF layer

Conditional random field(CRF) could employ the ordering of the LSTM output labels, so we use CRF to encode the output layer of BiLSTM.

A standard CRF layer is used on top of the hidden context vector h_t which are taken as features to make independent tagging decisions for each output y_t .But in Chinese NER, there are strong dependencies across output labels. For example, I-PER cannot follow B-ORG, which constraints the possible output tags after B-ORG. Therefore, we use CRF to simulate the output of the entire sentence together. For an input sentence,

$$X = (x_1, x_2, \dots, x_n) \tag{7}$$

we regard P as the matrix of scores outputted by BLSTM network. P is of size $n \times k$, where k is the number of distinct tags, and $P_{i,j}$ is the score of the j^{th} tag of the i^{th} character in a sentence. For a sequence of predictions,

$$y = (y_1, y_2, \dots, y_n) \tag{8}$$

we define its score as

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

where A is a matrix of transition scores, such $A_{i,j}$ represents the score of a transition from the tag i to tag j. The start and end tag to the set of possible tags and they are the tags of y_0 and y_n that separately means the start and the end symbol of a sentence. Thus, A is a square matrix of size $k + 2$. A softmax layer over all possible tag sequences yields the probability of the sequence y:

$$p(y|X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})}} \tag{10}$$

During training, we maximize the log-probability of the correct tag sequence:

$$\begin{aligned} log\big(p(y|X)\big) &= s(X, y) - log\left(\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})}\right) \\ &= s(X, y) - logadd\ s(X, \tilde{y}) \end{aligned} \tag{11}$$

where Y_X represents all possible tag sequences even unreasonable tag that do not obey the IOB format for a sentence X. From the formulation above, it is evident that we encourage our network to produce a valid sequence of output labels. While decoding, we predict the output sequence that obtains the maximum score given by:

$$\begin{aligned} y^* &= argmax\ s(X, \tilde{y}) \\ \tilde{y} &\in Y_X \end{aligned} \tag{12}$$

We are only modeling bigram constraints between outputs and computing decoding by dynamic programming.

3.3. Character-based model

Finally, our character-based model is constructed by feeding the output vectors of LSTM into a CRF layer.

The representation of character sequence $c_1, c_2, \dots, c_n$is computed to word embedding as inputs. The inputs of our model x_j is represented using

$$x_j^c = e^c(c_j) \tag{13}$$

e^c denotes a character embedding lookup table.

A bidirectional LSTM (same structurally as Eq. 11) is applied to $x_1, x_2, ..., x_m$ to obtain $\overrightarrow{h_1^c}, \overrightarrow{h_2^c}, \ldots, \overrightarrow{h_m^c}$ and $\overleftarrow{h_1^c}, \overleftarrow{h_2^c}, ..., \overleftarrow{h_m^c}$ in the left-to-right and right-to-left directions, respectively, with two distinct sets of parameters. The hidden vector representation of each character is:

$$h_j^c = \left[\overrightarrow{h_j^c}; \overleftarrow{h_j^c}\right] \tag{14}$$

Finally, the hidden vector representation h_j^c which is the output of BiLSTM are fed to the CRF layer to jointly decode the best label sequence.

A standard CRF model is used on $h_1^c, h_2^c, \ldots h_m^c$, for sequence labelling. Figure 2 illustrates the architecture of our network in detail.

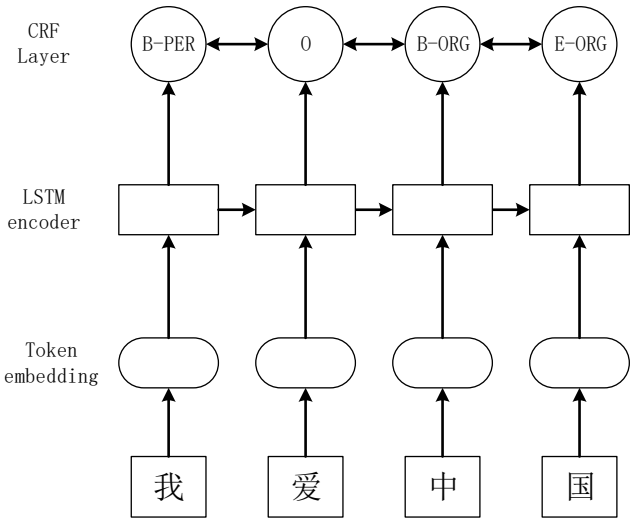

Figure 2: The basic architecture of our Character-based model.

3.4. Combining LSTM-CRF model with ELMo

In our LSTM-CRF model, we take the embeddings from language model(ELMo) as additional inputs to the sequence tagging model. Specifically, we concatenate the LM embeddings h^{LM} with word embedding x_k as input embedding $[x_k; h^{LM}]$ to expand word representation. In addition, we concatenate the LM embeddings h^{LM} with the hidden vector from one of the bidirectional LSTM layers in the sequence model. In our experiments, we found that adding h^{LM} to the hidden vector of the first layer and take $[h_k; h_k^{LM}]$ as the input of second layer performed the best. Model is able to disambiguate the same word into different representation based on its context. In addition, representation of out-of-vocabulary tokens can be represented in our language model by which we are capable to label the word with limited labeling data. Figure 3 illustrates the architecture of our model.

ELMo is a combination of multi-level representations of the biLM. For a certain word t_k, an L layer bidirectional language model biLM can be represented by $2L + 1$ vectors:

$$\begin{aligned} R_k &= \left\{X^{LM}, \overrightarrow{h_k}^{LM_j}, \overleftarrow{h_k}^{LM_j} \middle| j = 1,2, \ldots, L\right\} \\ &= \left\{h_k^{LM_j} \middle| j = 1,2, \ldots, L\right\} \end{aligned} \tag{18}$$

ELMo integrates the output of the multi-layered biLM into a vector $E(R_k; \theta_e)$. The simplest case is that ELMo only uses the top output:

$$E(R_k) = h_k^{LM,L} \tag{19}$$

This case is similar to the TagLM and CoVe models. However, we found that the best ELMo model is to add the weight of all biLM layers add the weight of normalized softmax :

$$E(R_k; \omega, \gamma) = \gamma \sum_{j=0}^{L} s_j h_k^{LM,j} \tag{20}$$

γ denotes a scaling factor, which represents layer nomalization for each layer of biLM before computing weight.

4. Experiments

4.1. Data set

We test our model on MSRA data set of the third SIGHAN Bakeoff Chinese named entity recognition task. This dataset contains three types of named entities: locations, persons, organizations. Standard precision (P), recall (R) and F1-score (F1) are used as evaluation metrics. Chinese word segmentation is not available in test set. We just replace every digit with a zero and unify the styles of punctuations appeared in MSRA and pretrained embeddings.

Our hyperparameter settings are modified based on previous work in the literature and grid search adjustments for the MSRA data set[20]. In particular, the embedding sizes are set to 300 and the hidden size of LSTM models to 300. Dropout is applied to both word and character embeddings with a rate of 0.5. Adaptive Moment Estimation (Adam) is used for optimization, with an initial learning rate of 0.001 and a decay rate of 0.05.

4.2. Results

Most recent work leverages an LSTM-CRF architecture. Lample et al.[6] use a character LSTM instead. Our baseline word-based system takes a similar structure to this line of work. Compared with the previous LSTM + CRF model, we combine the language model with the original model to get a more reasonable structure. Following previous English semi-supervised method[8] we trained on Chinese data set after tuning hyperparameters. Different from results reported by[8] in English, 80 dimensions is not enough to represent Chinese character. We use 300 dimensions in the following experiments.

Table 1: Results with different methods. We train our models for 100 epochs in this experiment.

Model	PER	LOC	ORG	P	R	F
Zhou[20]	90.09	85.45	83.10	88.94	84.20	86.51
Chen[21]	82.57	90.53	81.96	91.22	81.71	86.20
Zhou[22]	90.69	91.90	86.19	91.86	88.75	90.28
Dong[16]	91.77	92.10	87.30	91.28	90.62	90.95
Lu[24]	-	-	-	-	-	87.94
Wang[25]	-	-	-	91.39	91.09	91.24
BILSTM-CRF	89.23	92.50	85.45	89.93	86.93	88.41
Our model	91.52	**93.79**	**88.73**	**93.27**	**90.55**	**91.89**

Table1 shows our results compared with other models for Chinese named entity recognition. PER, LOC, and ORG indicate the person name, place name, and institution name. P, R, and F respectively indicate the accuracy rate, recall rate, and F value. We tried the model using elmo as word embeddings and the LSTM-CRF model combining with ELMo, Experiments show that the effect of our integrated model works. Zhou[19] got first place using word-based CRF model on MSRA data set with F1 86.51%. Chen [21] achieved F1 86.20% using character-based CRF model. Zhou[22] adopted a more granular labelling schemes for example changing PER tags. Dong[16] incorporates Chinese radical-level information to character-based BLSTM-CRF achieved F1 90.95%. Lu[21] present a position-sensitive skip-gram model to learn multi-prototype Chinese character embeddings. Wang[22] propose a novel architecture for NER problems based on Gated Convolutional Neural Networks and achieved F1 91.24%. Our neural network architecture does not need any extra information even less data than supervised model. In addition, we find that our model has a significant improvement over other models in such data as location.

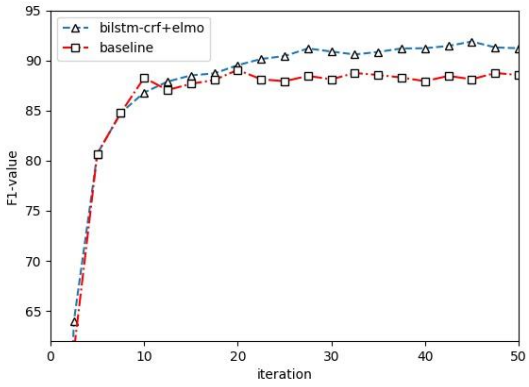

Fig. 3: F1 against training iteration number.

Figure 3 shows the F1- score of baseline and our models against the number of training iterations. As can be seen from the figure, additional representation of embedding is useful for improving character-based NER, improving the baseline result from 88.41% to 91.89%.

5. Conclusions

We proposed an advanced Semi-supervised approach for NER and empirically demonstrated that they achieve relatively good performance with much less data than models trained in the standard supervised approach. Unannotated sentences are used to strengthen our training on annotated data. Our model can effectively learn contextual content so that it can be easily transferred to other areas. Therefore, our model is not suitable for highly professional data sets, such as medical data or data sets that are not suitable for non-standard, such as social data. In the future, we hope to transfer our model to China's medical and social media.We have consideration for adding external information such as clinical dictionaries or emotions expressed by social media emoji in data preprocessing layer.

6. Acknowledgements

We thank all viewers who provided the thoughtful and constructive comments on this paper. The third author is the corresponding author. This research is funded by the Science and Technology Commission of Shanghai Municipality (No. 18511106202 and No.17511102000) and by Xiaoi Research. The computation is performed in ECNU Public Platform for Innovation(001).

7. References

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In NIPS.

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In EMNLP.

[3] Ronan Collobert, Jason Weston, Le´on Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. In JMLR.

[4] Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In ICLR.

[5] Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs- CRF. In ACL.

[6] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In NAACL-HLT.

[7] Peters, M. E. , Neumann, M. , Iyyer, M. , Gardner, M. , Clark, C. , & Lee, K. , et al. (2018). Deep contextualized word representations.

[8] Peters, M. E. , Ammar, W. , Bhagavatula, C. , & Power, R. . (2017). Semi-supervised sequence tagging with

bidirectional language models.

[9] James Hammerton. 2003. Named entity recognition with long short-term memory. In HLT-NAACL 2003-Volume 4. pages 172–175.

[10] Ronan Collobert, Jason Weston, Le´on Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. Journal of Machine Learning Research 12(Aug):2493–2537.

[11] Cıcero dos Santos, Victor Guimaraes, RJ Nitero´i, and Rio de Janeiro. 2015. Boosting named entity recognition with neural character embeddings. In Proceedings of NEWS 2015 The Fifth Named Entities Workshop. page 25.

[12] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for chinese word segmentation. In EMNLP. Lisbon, Portugal, pages 1197– 1206. http://aclweb.org/anthology/D15-1141.

[13] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via Bi-directional LSTM-CNNs- CRF. In ACL. volume 1, pages 1064–1074.

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

[15] Zhang, Y., Clark, S.: A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In: Proceedings of 2010 Conference on Empirical Methods in Natural Language Processing, pp. 843–852 (2010)

[16] Dong, C. , Zhang, J. , Zong, C. , Hattori, M. , & Di, H. . (2016). Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition. Natural Language Understanding and Intelligent Applications. Springer International Publishing.

[17] Shen, Y. , Yun, H. , Lipton, Z. C. , Kronrod, Y. , & Anandkumar, A. . (2017). Deep active learning for named entity recognition.

[18] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9, 1735–1780 (1997)

[19] Ga´bor Melis, Chris Dyer, and Phil Blunsom. 2017. On the state of the art of evaluation in neural language models. CoRR abs/1707.05589.

[20] Zhou, J., He, L., Dai, X., Chen, J.: Chinese named entity recognition with a multi- phase model. In: Proceedings of 5th SIGHAN Workshop on Chinese Language Processing, pp. 213–216 (2006)

[21] Chen, A., Peng, F., Shan, R., Sun, G.: Chinese named entity recognition with conditional probabilistic models. In: Proceedings of 5th SIGHAN Workshop on Chinese Language Processing, pp. 173–176 (2006)

[22] Zhou, J., Qu, W., Zhang, F.: Chinese named entity recognition via joint identifi- cation and categorization. Chin. J. Electron. 22, 225–230 (2013)

[23] Zhang Y , Yang J . Chinese NER Using Lattice LSTM[J]. 2018.

[24] Yanan Lu, Yue Zhang, and Dong-Hong Ji. 2016. Multi- prototype Chinese character embedding. In LREC.

[25] Wang, C. , Chen, W. , & Xu, B. . (2017). Named entity recognition with gated convolutional neural networks.