EasyChair Preprint

№ 9055

Study on Chinese Named Entity Recognition Based on Dynamic Fusion and Adversarial Training

Fan Fei, Linnan Yang, Xingyu Wu, Shengken Lin, Huijie Dong and Changshan Yin

October 24, 2022

Study on Chinese Named Entity Recognition Based on Dynamic Fusion and Adversarial Training

Fei Fan[1], LinnanYang[1*], Xingyu Wu[1], Shengken Lin[1], Huijie Dong[1] and Changshan Yin[1]

[1] School of Big Data, Yunnan Agriculture University, Kunming, Yunnan, China
Feifan07@aliyun.com 1067692880@qq.com lny5400@163.com
islsk@stu.xhsysu.edu.cn 931737240@qq.com yincss@126.com

Abstract. This paper aims at the Chinese Named Entity Recognition task, uses NEZHA Chinese pre-trained language model as the word embedding layer, then adopts the BiLSTM network architecture to encode it, and finally connects the CRF layer to optimize the output sequence. In order to enhance the fusion of semantic features of the NEZHA model in the upper, middle, and lower layers, an attention mechanism has been adopted to integrate the NEZHA coding layers. At first, weight was given to each representation generated by its 12 Transformer coding layers. Secondly, the weight value was dynamically adjusted through supervised training, and then the generated layer representation was weighted average to get the final word embedded representation. Finally, some noise was introduced to the input data, which is used for adversarial training to improve the generalization and robustness of the model. The results show that the F1 Score of the proposed model on Chinese Clinical Named Entity Recognition Dataset and people's daily corpus are respectively 98.52% and 96.84%, which are 2.36% and 4.21% respectively higher than the benchmark model Bert BiLSTM CRF.

Keywords: Natural language processing, Chinese named entity recognition, NEZHA, Dynamic fusion, Adversarial training.

1 Introduction

Named Entity Recognition (NER) technology is one of the basic core tasks in natural language processing (NLP). It can identify entities in text and their corresponding types, such as human names, place names, institutional names, etc. It is an essential part of NLP downstream tasks such as Information Extraction, Q & A System, and Knowledge Graph.

Initially, rule and dictionary matching approaches were used in the research path of named entity recognition. For example, Liu et al. [1] recognized numeric and temporal expressions by designing and tuning optimal templates. Dictionary matching methods

extract all matching strings from the target sequence utilizing entities built into the dictionary. These approaches may be successful in specific domains, but both fail to solve the OOV (Out-Of-Vocabulary) problem [2]. In addition, both methods rely heavily on time-consuming manual features. Later on, statistical machine learning approaches became popular, and CRF models [3] became the most commonly used method for named entity recognition. Li et al. [4] and Malarkodi et al. [5] used CRF models to identify agricultural named entities, such as crops, pests, and pesticides, on their self-constructed annotated corpus and obtained encouraging experimental results by selecting different combinations of features. The statistical-based machine learning approach effectively improves the accuracy of Chinese named entity recognition, but it is still time-consuming and tedious because it relies on feature engineering [6]. In addition, the CRF model is often unsatisfactory for the extraction of a large number of word-length entities in the text, and there are many breaks in the continuous entities.

Recurrent neural networks (RNNs) are widely applied to natural language processing tasks due to they maintain a memory based on historical information. They can be able to align with the text. Among RNNs, the Bi-directional long-short term memory network (BiLSTM) [7] is one of the most widely used RNN structures. Huang [8] was the first to apply BiLSTM and CRF to the sequence annotation task. Since BiLSTM has the powerful ability to learn word context representations, it has been adopted as an encoder by most NER models. Yue [9] proposed the Lattice LSTM network that encodes input character sequences and potential words for all matching dictionaries, making full use of word and word order information.

Recently, the pre-trained language model BERT [10] has been widely used on various NLP tasks, and some researchers have proposed to use the pre-trained language model BERT as a word embedding layer, and the acquired word representations have richer semantic and syntactic information because it has a more expressive bi-directional Transformer encoder [11]. Fabio [12] applied the BERT-CRF model to Portuguese NER to obtain the best F1 value on HAREM I; Jana [13] used the BERT pre-trained language model for entity recognition to achieve quite desirable results on CoNLL-2002 Dutch, Spanish and CoNLL-2003 English.

In recent years, the combination of deep learning and adversarial training has become popular in the field of natural language processing, and it has become an alternative path for text research as a regularization method. Miyato et al. [14] used deep learning techniques and proposed for the first time to add perturbations to the word vector layer for a semi-supervised text classification task. Bekoulis et al. [15] applied adversarial training to a joint model of entity recognition and relation extraction in a joint model, achieving excellent results across languages and multiple datasets. Zhou et al. [16], on the other hand, added perturbation to the word embedding layer to improve the generalization ability of named entity recognition models with low resources.

The remainder of the paper is organized as follows. Section 2 describes the basic network model for named entity recognition used in this article. Section 3 shows the principle of dynamic fusion and adversarial training algorithm. Section 4 discusses related research. Finally, Section 5 draws conclusions.

2 Methodology

2.1 Model Architecture

The primary model designed in this paper is the NEZHA (NEural contextualiZed representation for Chinese lAnguage understanding) [17] pre-trained language model of dynamic fusion, followed by the BiLSTM network architecture, and finally added to the CRF network layer. The model structure is shown in Fig. 1.

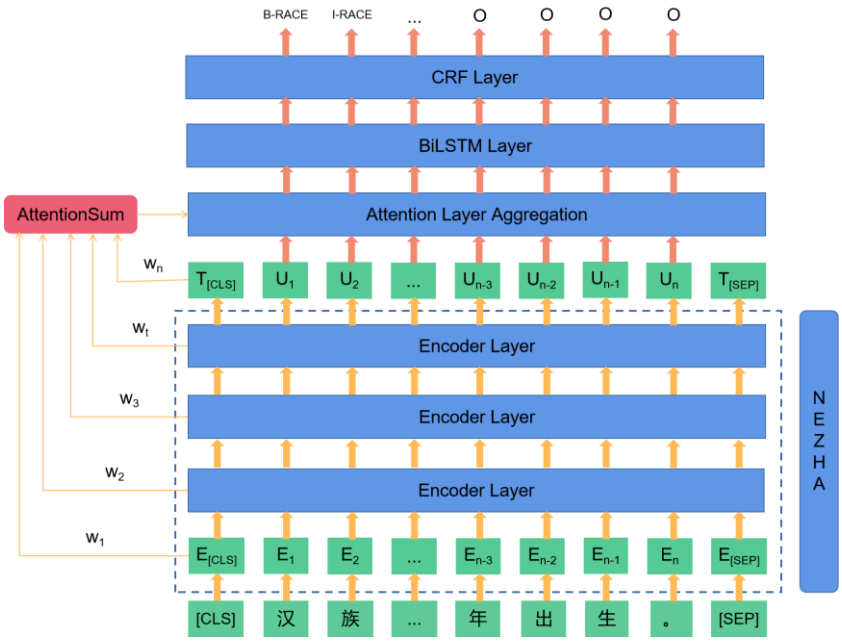

Fig. 1. Overall Model architecture

2.2 Nezha pre-trained language model

Huawei Noah's Ark Lab released the NEZHA pre-trained language model in 2019, based on the BERT model and trained on Chinese text. The main innovation is a functional relative position encoding technique, which encodes the relative position of the self-attentive layer by a predefined function without any trainable parameters.

In Transformer, each attention head processes a sequence of tokens $x=(x_1,x_2,...,x_n)$, where $x_i \in R^{d_x}$, and outputs a sequence $z=(z_1,z_2,...,z_n)$, where $z \in R^{d_z}$, and each attention head has three parameter matrices W^K, W^Q, $W^V \in R^{d_x \times d_z}$, to be learned, and the output z_i is computed as follows.

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j w^v) \tag{1}$$

The attention score α_{ij} between the hidden states at position i and position j is calculated using the softmax function.

$$\alpha_{ij} = \frac{exp\ e_{ij}}{\sum_k exp\ e_{jk}} \tag{2}$$

where e_{ij} is the scaled dot product between the linear transformations of the input elements.

$$e_{ij} = \frac{(x_i W^Q)(x_j W^k)^T}{\sqrt{d_z}} \tag{3}$$

The traditional Transformer or BERT model, which uses an absolute position information encoding technique, is insensitive to the word order requirement for multi-headed attention between words. The position encoding embedding and the word embedding are summed linearly together as the input to the model. Parametric relative positional encoding was proposed later, noting that the score calculation depends on the parametric embedding of the relative distance between two positions. Specifically, the output z_i of equation (1) and the computational procedure of e_{ij} of equation (3) are modified as follows.

$$z_i = \sum_{j=1}^{n} \alpha_{ij} (x_j W^V + \alpha_{ij}^V) \tag{4}$$

$$e_{ij} = \frac{(x_i W^Q)(x_j W^k + \alpha_{ij}^K)^T}{\sqrt{d_z}} \tag{5}$$

In the above two equations, α_{ij}^V, $\alpha_{ij}^K \in \mathbb{R}^{d_z}$ are two vectors with relative positions encoded for i and j, which are shared over all attentional heads. Transformer-XL and XLNet use different formulations to implement relative position encoding.

In contrast, the NEZHA model uses a functional relative position encoding, where the output and attention scores are computed depending on the sine function of the relative position, specifically, α_{ij}^V and α_{ij}^K in the model are derived from the sine function and the mode is fixed during the model training, As shown in the following equation, α_{ij} is used to represent α_{ij}^V, α_{ij}^K uniformly, and α_{ij} under dimension 2k and dimension 2k+1 is considered respectively.

$$\alpha_{ij}[2k] = \sin\left(\frac{j-i}{10000^{\frac{2\cdot k}{d_z}}}\right) \tag{6}$$

$$\alpha_{ij}[2k+1] = cos\left(\frac{j-i}{10000^{\frac{2\cdot k}{d_z}}}\right) \tag{7}$$

Each dimension of the location encoding corresponds to a sinusoidal function with different wavelengths for different dimensions. In the above equation, d_z is equal to the hidden dimension of each attention head in the NEZHA model (i.e., the hidden dimension divided by the number of attention heads). The wavelength is a geometric progression from 2π to $10000 \cdot 2\pi$. The fixed sine function was chosen because it allows the model to extrapolate to longer sequence lengths than those encountered in training. Techniques that have proven effective in BERT pre-trained models are also used, namely the full word mask technique, mixed-precision training, and LAMB optimizer,

and the model performance is improved, and the training speed is significantly increased.

$$\alpha_{ij}[2k+1] = cos\left(\frac{j-i}{10000^{\frac{2\cdot k}{d_z}}}\right) \tag{7}$$

2.3 BiLSTM Network Layer

Traditional recurrent neural networks are usually characterized by a temporal structure and play an important role in tasks such as natural language processing and sequence prediction. LSTM is a recurrent neural network with memory units proposed by Schmidhuber in 1997, which is an improved version of the traditional recurrent neural network model.

LSTM is a typical operation of the popular RNN model that captures random sequence features and processes sequence data. The LSTM network mainly consists of three gating units and one memory unit, which will output two states to the next unit, the unit's state and the hidden state, respectively. The LSTM can capture remote dependencies through three gating mechanisms, namely, input gates, forgetting gates, and output gates. The input gate determines the percentage of the state saved from the current state to the cell state, the forgetting gate determines what information needs to be retained in the previous step, and the output gate is used to output the percentage of the state information from inside the memory cell at the current moment to determine the value of the next hidden state. The internal structure of an LSTM cell is shown in Figure 2.

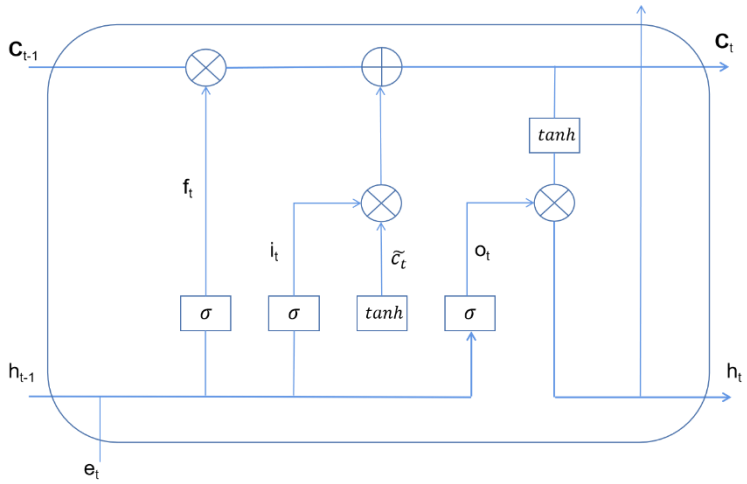

Fig. 2. Overall Model architecture

Since the one-way LSTM network can only capture the historical information of the acquired sequence, the category information of each word in the sequence is closely related to the context. Inspired by this, Graves et al. improved and proposed a bidirectional LSTM network (BiLSTM), which takes the model's ability to extract contextual

information a step further, with applications in more domains such as speech recognition, lexical annotation, and named entity recognition.

First, we use the left and right contexts to identify named entities and then apply BiLSTM to mine the hidden expressions of characters from the global context, as shown in the following equation.

$$\left[\underset{h_1}{\rightarrow},\underset{h_2}{\rightarrow},\ldots,\underset{h_N}{\rightarrow}\right]=\underset{\text{LSTM}}{\longrightarrow}\left(\left[e_1,e_2,\ldots,e_N\right]\right) \tag{8}$$

$$\left[\underset{h_1}{\leftarrow},\underset{h_2}{\leftarrow},\ldots,\underset{h_N}{\leftarrow},\right]=\underset{\text{LSTM}}{\longleftarrow}\left(\left[e_1,e_2,\ldots,e_N\right]\right) \tag{9}$$

Where e_i denotes the i-th character embedded after the word embedding layer, h_i denotes the output of the forward and reverse LSTM, and the hidden representation of the i-th character is concatenated by h_i.

$$h_i=\left[\underset{h_i}{\rightarrow},\underset{h_i}{\leftarrow}\right] \tag{10}$$

Finally, the output of the BiLSTM layer is defined as $h=[h_1,h_2,...,h_N]$, where $h_i \in R^{2s}$,S denotes the dimensionality of the LSTM hidden states.

2.4 CRF Network Layer

While models of the coding layer can identify entity boundaries and do not consider whether the relationship between entity sequences is correct, CRF models can obtain global optimal label sequences by considering dependencies between neighboring labels and are therefore often applied to tasks such as speech labeling and named entity recognition. CRF is a sequence labeling algorithm based on the EM and HMM models. By considering the global information on the tag sequence, the tag bias problem can be solved, and the tags are better predicted.

The rationale of CRF is to calculate the conditional probability distribution of the output random variable with a given random variable as the input, usually decoded using the Viterbi algorithm. The CRF model for named entity recognition used word sequences in input sentences as observed sequences and the labeling procedure is inferred from the most likely label sequences based on known word sequences.

3 Related Work

3.1 Dynamic Fusion

The paper published by Jawahar [18] in ACL 2019 states that low-level networks of BERT learn phrase-level information representations, middle networks of BERT learn rich linguistic features, while BERT's high-level network learns rich semantic information features. For entity recognition in the general domain, the model focuses on the top-level semantic features while ignoring the underlying features urgently needed by the entity recognition task. In addition, taking information directly from a high level can easily lead to over-fitting. Therefore, this study employs an attention-based multilayer dynamic weight fusion approach to the BERT model.

Different BERT pre-trained language models also contain layer coding layers; generally, there are 12, 24, and 48 layers, recording the number of layers as L, as attention mechanism-based layer fusion, where both α and γ are trainable parameters. This is shown in Equations (11) and (12).

$$\mathrm{h} = \gamma \sum_{i=1}^{N} w_i \, h_i \tag{11}$$

$$w_i = \frac{\exp\,(\alpha_i)}{\sum \exp\,(\alpha_j)} \tag{12}$$

Where h is the output of the middle layer of the BERT model; w is the weight of each layer.

This paper weighs the representations generated by the 12-layer coding layer in the NEZHA pre-trained language model, then determines the weights dynamically through training. Each layer's weighted average gets the final feature representation and is then sent to the subsequent downstream network layer to obtain the prediction results.

3.2 Adversarial Training

The father of GAN Lan Goodfellow first proposed the concept of adversarial training [19] in 15 years of ICLR, simply adding a perturbation to the original input sample and being trained with it after obtaining the adversarial sample. That is, the problem can be abstracted into such a model:

$$\min_{\theta} -logP(y|x + r_{adv}; \theta) \tag{13}$$

Among these, y is the gold label, and θ is the model parameter. So how is the disturbance calculated? Goodfellow suggests that neural networks are vulnerable to linear perturbations due to their linear characteristics. So, he proposed the Fast Gradient Sign Method (FGSM) to calculate the perturbations of the input samples. Perturbation can be defined as:

$$r_{adv} = \epsilon \cdot sgn\big(\nabla_x L(\theta, x, y)\big) \tag{14}$$

Where sgn is the symbolic function and L is the loss function. In summary, the two roles of adversarial training are improving the robustness of the model in response to malicious adversarial samples, providing a regularized supervised learning algorithm, reducing overfitting, and improving generalization ability.

At this point, the theoretical part of adversarial training is compared to intuitive, Madry's previous work in ICLR 2018[20], and redefined the problem as a problem of finding a saddle point, the well-known Min-Max formula:

$$\min_{\theta} E_{(x,y)\in\mathcal{D}} \left[\max_{r_{adv}\epsilon s} L(\theta, x + r_{adv}, y) \right] \tag{15}$$

The formula is divided into two parts, maximization of internal loss function and minimization of external empirical risk. The internal max is designed to find the perturbation of the worst-case, that is, the attack, where L is the loss function and S is the range space of the perturbation. The external min is designed to find the most robust model parameter, namely the defense, based on the attack mode, where the input samples are distributed. According to Madry, this formula simply and clearly defines two questions

of the "spear and shield" of adversarial samples: how to construct strong enough adversarial samples? Moreover, how to make the model boring? The rest is the question of how to solve it.

We mentioned above that Goodfellow proposed FGSM in ICLR 2015 subsequently, and in ICLR 2017 [13], Goodfellow made a simple little modification to the portion of the computational perturbations in FGSM. Assuming that the embedding vectors of the input text sequence is [v1, v1, ..., vT], The perturbation for the x's embedding is:

$$r_{adv} = \epsilon \cdot g / \left|\left|g\right|\right|_2 \tag{16}$$

$$\mathrm{g} = \nabla_x L(\theta, x, y) \tag{17}$$

Where, G on behalf of the gradient, $\left|\left|g\right|\right|_2$ for L2 norm of the gradient, with the L2 norm made a scale, from the formula, L2 normalized more keep the direction of the gradient, and Max normalization is not necessarily in the same direction that the original gradient. Of course, they all have a common premise. That is, the loss function L must be linear or at least locally linear so as to ensure that the direction of gradient lifting is the optimal direction. Assuming that the number of rounds of mini-Batch is M and the number of the epoch is T, it can be seen from the code that the time complexity of FGSM and FGM is O(T*M).

In this study, two adversarial training algorithms, FGM and FGSM, found that the FGM effect was more applicable, so this method was used for adversarial training.

4 Experiment

4.1 Datasets

This paper selects CNER datasets from this paper [19], which are filtered and manually annotated according to the resume summary data of senior managers of listed companies. The datasets contain 1027 resume summaries with entity notation classified into eight categories, including human name, nationality, origin, race, major, degree, institution, and title.

This paper also performs statistics for text length, and considering the better capture of context information as well as the input sequence length limit of the BERT model, we tangent the text length by a 512-character length. To ensure the integrity of the sentence and reduce the absence of context semantics, the period as a tangent truncates the forward index of length 512, and the remaining sentences are added to the next subsequence, thus ensuring that a sentence is not cut into two parts.

4.2 Experimental setup

This experiment uses a UTC 2021 x86_64 GNU/Linux server with a Quadro RTX 5000 graphics card and CUDA version 11.0. The neural network model is built in the PyTorch framework with PyTorch 1.7.1, and its corresponding Python version is 3.8. the following table lists the hyper-parameter values of the experimental model, with the word vector dimension set to 100 and the hidden layer dimension of the network model set to 128. In this paper, we introduce the Warm-up mechanism to mitigate the model

over-fitting problem, with preheating learning rate set to 0.05, recession rate of 0.5, ADAM optimization algorithm, NEZHA fine-tuning learning rate of 1e-4, 5e-5, model training batch parameter of 16,and the number of iterations is set to 8. In addition, the accuracy P, recall R, and F1 values are used as the evaluation indicators of the experiment.

Table 1. Hyper-parameters.

Hyper-Parameters	Values
Warmup_proportion	0.05
decay_rate	0.5
train_epoch	8
learnig_rate	1e-4
embed_learning_rate	5e-5
sequence_length	512
lstm_hidden	128
batch_size	16

4.3 Results

The recognition effect of the model was tested on the annotated corpus without relying on artificial design features by constantly adjusting the model parameters. The training set in the corpus, the test set, divided by the validation set is 7:2:1 with no overlap between the three, so it is reasonable to take the output of the test set as an evaluation index of entity recognition effect.

To show the performance effect of the NEZHA-BiLSTM-CRF model proposed here in the Chinese named entity recognition task, we performed comparison experiments on other models, including BiLSTM-CRF, Bert-BiLSTM-CRF. We then compared the comprehensive effect of adding dynamic fusion and adversarial training. The specific comparison results are shown in the table below, and in the table, DF represents dynamic fusion, and AT represents adversarial training.

Table 4. Experimental results.

Model	CNER			People's Daily
	Precision (%)	Recall (%)	F1 (%)	F1 (%)

BiLSTM-CRF	95.74	95.72	95.70	87.94
Bert-BiLSTM-CRF	95.62	96.71	96.16	92.63
Nezha-BiLSTM-CRF	96.70	98.10	97.39	95.60
NEZHA-BiLSTM-CRF+DF	97.01	99.08	98.00	96.44
NEZHA-BiLSTM-CRF+DF+AT	97.60	99.50	98.52	96.84

From the results in the table, it can be seen that the F1 value of the proposed model and method in this paper improves by 2.82% in terms of comprehensive evaluation metrics when adding the comparison benchmark model BiLSTM-CRF, while based on the benchmark model BiLSTM-CRF, the large-scale pre-trained model BERT increased by only 0.46%, whereas the Chinese pre-trained language model NEZHA can improve by 1. 69%, which fully shows that the NEZHA pre-trained language model has more ability to extract features, is well-targeted for Chinese NER tasks, and can greatly improve the training speed of the model.

The addition of the NEZHA pre-trained language model can be improved by 0.61%, which shows the effectiveness of the dynamic fusion at all layers of NEZHA. Finally, the F1 value increases by 0.52% after adding countermeasure training, which shows that the countermeasure training algorithm can improve the generalization and robustness of the model.

In addition, we also selected the People's Daily corpus released by the Institute of Computational Language of Peking University for further validation of the experiment, which is one of the largest Chinese annotated corpora constructed in China, with information such as names of people, places, and organizations annotated in the corpus. The validation results also show the model's effectiveness and method proposed in this paper.

5 Conclusion

For the problems of text context-dependence and long entity type in the task of Chinese named entity recognition, this paper proposed a recognition method based on the Nezha pre-training model, which improves the performance of the model to a certain extent, and the hybrid precision training and lamb trainer also significantly improve the training speed. In this paper, we performed dynamic fusion for each layer of the NEZHA pre-trained model to obtain a better vector representation and verified the effectiveness of the semantic fusion mechanism. In addition, the input samples are trained in the way of adversarial training, which improves the robustness and generalization ability of the model to a certain extent, and improves the effect of Chinese named entity recognition.

The named entity recognition model proposed in this paper has achieved good recognition results in the open Chinese corpus. However, there are still some shortcomings that exist and need to be further improved and enhanced in future research work, which can be broadly focused on the following three aspects: first, potential word features will be incorporated into the model in this paper, combining BERT and Lattice LSTM to characterize the polysemy of words, while adding potential word features as a way to

address the lack of contextual information. Second, introduce more advanced pre-trained language models, such as XLNET, RoFormer, etc., in order to further explore the model performance for better recognition effects. Third, some model compression methods, such as knowledge distillation [21], pruning, etc., are tried to reduce the training time, computational power, and spatial complexity.

Acknowledgement. This work has been supported by the Major Project of Science and Technology of Yunnan Province under Grant No.202002AD080002.

References

1. Liu W, Yu B, Zhang C, et al. Chinese Named Entity Recognition Based on Rules and Conditional Random Field[C]//Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence. 2018: 268-272.
2. Wang X, Jiang X, Liu M, et al. Bacterial named entity recognition based on dictionary and conditional random field[C]//2017 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, 2017: 439-444.
3. Wallach H M. Conditional random fields: An introduction[J]. Technical Reports (CIS), 2004: 22.
4. Li X, Wei X H, Jia L, et al. Recognition of crops, diseases and pesticides named entities in Chinese based in conditional random fields[J]. Trans. Chin. Soc. Agric. Mach, 2017, 48: 178-185.
5. Malarkodi C S, Lex E, Devi S L. Named Entity Recognition for the Agricultural Domain[J]. Res.Comput. Sci., 2016, 117: 121-132.
6. Zhou G D, Su J. Named entity recognition using an HMM-based chunk tagger[C]//Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002: 473-480.
7. Hochreiter S., Schmidhuber J.: Long short-term memory. In: Neural computation 9.8, pp. 1735-1780 (1997)
8. Huang Z., Xu W., Yu K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv:1508.01991 (2015)
9. Zhang Y., Yang J.: Chinese NER using lattice LSTM. arXiv:1805.02023 (2018)
10. Devlin J., Chang M.W., Lee K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
11. Vaswani A., Shazeer N., Parmar N., et al.: Attention is all you need//Advances in neural information processing systems. 5998-6008 (2017)
12. Souza F., Nogueira R., Lotufo R.: Portuguese named entity recognition using BERT-CRF.arXiv:1909.10649 (2019)
13. Straková J., Straka M., Hajič J.: Neural architectures for nested NER through linearization.arXiv:1908.06926, (2019)
14. Wei J., Ren X., Li X., et al.: Nezha: Neural contextualized representation for chinese language understanding. arXiv:1909.00204 (2019)
15. Jawahar G., Sagot B., Seddah D.: What does BERT learn about the structure of language? In:Proceedings of the 57rd Annual Meeting of the ACL, pp. (2019)
16. Goodfellow I.J., Shlens J., Szegedy C.: Explaining and harnessing adversarial examples.arXiv:1412.6572 (2014)
17. Madry A., Makelov A., Schmidt L., et al.: Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083 (2017)

18. Miyato T., Dai A.M., Goodfellow I.: Adversarial training methods for semi-supervised text classification. arXiv:1605.07725 (2016)
19. Miyato T, Dai A M, Goodfellow I. Adversarial training methods for semi-supervised text classification[J]. arXiv preprint arXiv:1605.07725, 2016.
20. Chen X, Cardie C. Multinomial adversarial networks for multi-domain text classification[J]. arXiv preprint arXiv:1802.05694, 2018.
21. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015, 2(7).