Chinese Named Entity Recognition via Adaptive Multi-pass Memory Network with Hierarchical Tagging Mechanism

Pengfei Cao[1,2], **Yubo Chen**[1,2], **Kang Liu**[1,2] and **Jun Zhao**[1,2]
[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences,
Beijing, 100049, China
{pengfei.cao, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Named entity recognition (NER) aims to identify text spans that mention named entities and classify them into pre-defined categories. For Chinese NER task, most of the existing methods are character-based sequence labeling models and achieve great success. However, these methods usually ignore lexical knowledge, which leads to false prediction of entity boundaries. Moreover, these methods have difficulties in capturing tag dependencies. In this paper, we propose an **A**daptive **M**ulti-pass **M**emory **N**etwork with **H**ierarchical **T**agging Mechanism (**AMMNHT**) to address all above problems. Specifically, to reduce the errors of predicting entity boundaries, we propose an adaptive multi-pass memory network to exploit lexical knowledge. In addition, we propose a hierarchical tagging layer to learn tag dependencies. Experimental results on three widely used Chinese NER datasets demonstrate that our proposed model outperforms other state-of-the-art methods.

1 Introduction

The task of named entity recognition (NER) is to recognize the named entities from a plain text and classify them into pre-defined types. NER is a fundamental and preliminary task in natural language processing (NLP) area and is beneficial for many downstream NLP tasks such as relation extraction (Bunescu and Mooney, 2005), event extraction (Chen et al., 2015) and question answering (Yahya et al., 2013). In recent years, numerous methods have been carefully studied for NER task, including Conditional Random Fields (CRFs) (Lafferty et al., 2001) and Support Vector Machines (SVMs) (Isozaki and Kazawa, 2002). Currently, with the development of deep learning methods, neural networks have been introduced for the NER task. In particular, sequence labeling neural network models have achieved state-of-the-art performance (Lample et al., 2016; Zhang and Yang, 2018).

Though sequence labeling neural network methods have achieved great success for Chinese NER task, some challenging issues still have not been well addressed. One significant drawback is that previous methods usually **fail to correctly predict entity boundaries**. To conduct a quantitative analysis, we perform a BiLSTM+CRF model proposed by Huang et al. (2015), which is the most representative Chinese NER sequence labeling system, on WeiboNER dataset (Peng and Dredze, 2015; He and Sun, 2016), OntoNotes 4 dataset (Weischedel et al., 2011) and MSRA dataset (Levow, 2006). The F1 scores are 55.84%, 63.17% and 89.13%, respectively. We do a further analysis and find that the errors of predicting entity boundaries are particularly serious. The average proportion of predicting entity boundaries errors is 82% on these three datasets. For example, the character-based BiLSTM+CRF model fails to predict the entity boundaries of “北海道 (Hokkaido)” in Figure 1. To reduce the errors of predicting entity boundaries, some works (Peng and Dredze, 2016; Cao et al., 2018) try to jointly perform Chinese NER with Chinese word segmentation (CWS) for using word boundaries information. However, the joint model requires additional annotated training data for CWS task.

Fortunately, existing lexicons can provide information on word boundaries and we refer to the information as lexical knowledge. In addition, the cost of obtaining lexicon is low and almost all fields have their lexicons, such as biomedical, social science fields and so on. Recently, Zhang and Yang (2018) propose a lattice LSTM model capable of leveraging lexicon for Chinese NER. Though effective, the lattice LSTM

Proceedings of the 19th China National Conference on Computational Linguistics, pages 928-939, Hainan, China, October 31 - Novermber 1, 2020.

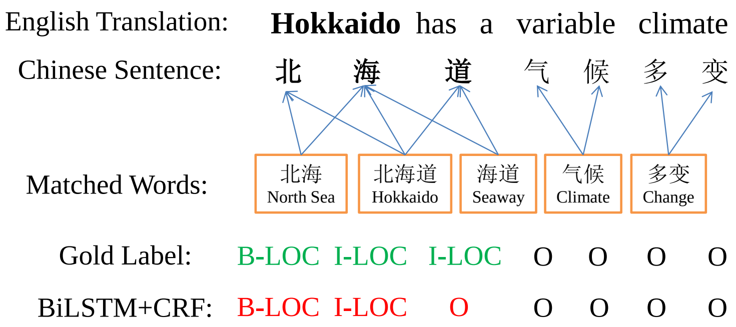


Figure 1: An example of Chinese NER with wrong entity boundaries using the BiLSTM+CRF model. It also shows the matched words for each character.

cannot exploit all matched words. When the candidate labeled character is within a matched word (i.e. the character is not the first or the last character of the matched word), the lattice model cannot explicitly and directly exploit the matched word. For example, for the candidate labeled character “海 (Sea)”, it can match “北海 (North Sea)”, “海道 (Seaway)” and “北海道 (Hokkaido)” in lexicon according to its context. When exploiting the matched words for character “海 (Sea)”, the lattice model only considers “北海 (North Sea)” and “海道 (Seaway)”, ignoring “北海道 (Hokkaido)” which can help determine that the character “海 (Sea)” is the middle of an entity rather than beginning or ending. Moreover, the lattice model only processes the matched words once, when learning the lexical knowledge for a character. However, it needs more reasoning passes on the matched words to better learn lexical knowledge in complex sentences intuitively. Take the sentence “南京市长江大桥 (Nanjing Yangtze River Bridge)” for example, it is more complicated than the sentence in Figure 1 because it is prone to be misunderstood as “南京市长/江大桥 (The mayor of Nanjing is Jiang Daqiao)”. Thus, it needs more reasoning passes to learn the lexical knowledge for recognizing the entity “长江大桥 (Yangtze River Bridge)” than the entity “北海道 (Hokkaido)” in Figure 1. However, if the reasoning passes are too many, the performance will decrease in word sense disambiguation task (Luo et al., 2018). We argue that the problem also exists in Chinese NER task. Hence, how to exploit all matched words and perform flexible multi-pass reasoning according to the complexity of sentences should be well investigated.

Another issue is that most of the existing methods **cannot efficiently capture tag dependencies**. In sequence labeling neural network models, CRF is usually used as a decoding layer. Although the CRF decoder has achieved improvements, the transition matrix in CRF layer only learns the neighboring tag dependencies, which are typically first order dependencies (Zhang et al., 2018). Thus, CRF cannot well handle long-distance tag dependency problems. For example, in the sentence “耐克拥有比李宁更大的市场 (Nike has a larger market than Li Ning)”, the tag of “李宁 (Li Ning)” is dependent on the tag of “耐克 (Nike)”, as they should be the same entity type. Since “李宁 (Li Ning)” can be a person or an organization, it is more difficult to predict the tag of “李宁 (Li Ning)” than “耐克 (Nike)”. However, it is easy to tag “耐克 (Nike)” as an organization. If we capture the dependencies between “李宁 (Li Ning)” and “耐克 (Nike)”, we will have ample evidence to tag “李宁 (Li Ning)” as an organization. To address the issue, Zhang et al. (2018) exploit the LSTM as decoder instead of CRF. However, the unidirectional LSTM decoder only leverages the past labels and ignores the future labels. In another sentence “李宁努力地同耐克竞争 (Li Ning strives to compete with Nike)”, when predicting the tag of “李宁 (Li Ning)”, the future tag of “耐克 (Nike)” can help us to determine the tag of “李宁 (Li Ning)”. Thus, how to capture bidirectional (past and future) tag dependencies in the whole sentence is another challenging problem.

In this paper, we propose an **A**daptive **M**ulti-pass **M**emory **N**etwork with **H**ierarchical **T**agging Mechanism (**AMMNHT**) to address the aforementioned problems. To exploit all matched words and perform multi-pass reasoning across matched words for a character, memory network (Sukhbaatar et al., 2015) can be utilized for Chinese NER. However, conventional memory network follows pre-defined passes to perform multi-pass reasoning and cannot perform adaptive and proper deliberation passes according to the change of input sentence. We utilize reinforcement learning (Sutton et al., 1998) to adaptively determine the deliberation passes of memory network according to the complexity of sentences. Although we do not have explicit supervision for the reasoning passes of the memory network, we can obtain long-term feedback (or *reward*) from the final prediction, which inspires us to utilize reinforcement learning techniques. To capture bidirectional tag dependencies in the whole sentence, we propose a hierarchical tagging mechanism for Chinese NER task.

In summary, the contributions of this paper are listed as follows:

- We propose a novel framework to integrate lexical knowledge from the lexicon for Chinese NER task, which can explicitly exploit all matched words and adaptively choose suitable reasoning passes for each sentence. To our best knowledge, this is the first work to automatically determine the reasoning passes of memory network via reinforcement learning techniques.

- We propose a hierarchical tagging mechanism for Chinese NER to capture bidirectional tag dependencies in the whole sentence. To our knowledge, this is the first work to devise the hierarchical tagging mechanism for Chinese NER task.

- Experiments on three widely used Chinese NER datasets show that our proposed model outperforms previous state-of-the-art methods.

2 Related Work

In recent years, the NER task has attracted much research attention. Many methods have been proposed to perform the task. Early studies on NER often exploit CRFs (Lafferty et al., 2001) and SVMs (Isozaki and Kazawa, 2002). These methods rely heavily on feature engineering. However, the designed features may be not appropriate for the task, which can lead to error propagation problem. Currently, neural network methods have been introduced into NER task and achieved state-of-the-art performance (Lample et al., 2016). Huang et al. (2015) use the bidirectional long short term memory (BiLSTM) for feature extraction and the CRF for decoding. The model is trained via the end-to-end paradigm. After that, the BiLSTM+CRF model is usually exploited as the baseline model for NER task. Ma and Hovy (2016) use a character convolutional neural network (CNN) to represent spelling characteristic. Then the charcter representation vector is concatenated with word embedding as the input of the LSTM. Peters et al. (2017) leverage a character language model to enhance the input of the model.

For Chinese NER, character-based methods have been the dominant approaches (Lu et al., 2016; Dong et al., 2016). These methods only focus on character sequence information, ignoring word boundaries information, which can cause errors of predicting entity boundaries. Thus, how to better exploit lexical knowledge has received much research attention. Word segmentation information is used as extra features for Chinese NER task (Peng and Dredze, 2015; He and Sun, 2016). Peng and Dredze (2016) and Cao et al. (2018) propose a joint model for Chinese NER, which is jointly trained with CWS task. Zhang and Yang (2018) investigate a lattice LSTM to encode a sequence of input characters as well as words that match a lexicon. However, the lattice model cannot exploit all matched words and only processes the matched words once. Recently, graph-based models have been proposed for Chinese NER (Gui et al., 2019; Sui et al., 2019). Based on the lattice structure, Sui et al. (2019) propose a graph neural network to encode word information.

Tag dependencies is also a challenging problem, but few attention has been paid to tackling the problem. Zhang et al. (2018) leverages LSTM as decoder for sequence labeling task. However, the unidirectional LSTM decoder only exploits the past predicted tags information, ignoring the future un-predicted tags. Hence, we propose a hierarchical tagging mechanism to capture bidirectional tag dependencies in the whole sentence. To our best knowledge, we are the first to introduce the hierarchical tagging mechanism to Chinese NER task. Moreover, to better capture the dependencies between tags, we try different hierarchical tagging mechanism.

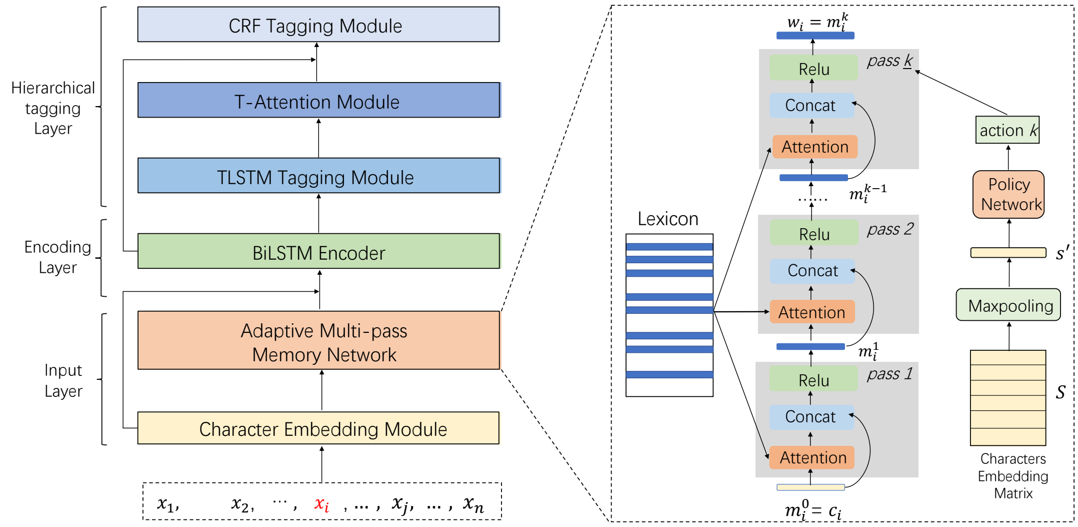


Figure 2: The architecture of our proposed adaptive multi-pass memory network with hierarchical tagging mechanism. The right part is the adaptive multi-pass memory network (AMMN). For each character, the lexical knowledge (i.e., w_i in the figure) is obtained via the AMMN. We concatenate the character embeddings and lexical knowledge as the input of the encoding layer. In this figure, we use the character x_i as an example to illustrate the process.

3 Method

The architecture of our proposed model is shown in Figure 2. The proposed model consists of three components: input layer, BiLSTM encoding layer and hierarchical tagging layer. In the following sections, we will describe the details of our proposed model.

3.1 Input Layer

The inputs of our proposed model are character embeddings and lexical knowledge, which are obtained via character embedding module and adaptive multi-pass memory network, respectively.

Character Embedding Module Similar to other methods using neural networks, the first step of our proposed model is to map discrete language symbols to distributed representations. Formally, given a Chinese sentence $s = \{x_1, x_2, \ldots, x_n\}$, each character x_i is represented by looking up embedding vector from a pre-trained character embedding table:

$$c_i = E^c(x_i) \tag{1}$$

where E^c is a pre-trained character embedding table and $c_i \in \mathbb{R}^{d_c}$. We obtain the characters embedding matrix, denoted as $S = \{c_1, c_2, \ldots, c_n\}$.

Adaptive Multi-pass Memory Network The adaptive multi-pass memory network has three inputs: the candidate character embedding c_i as the initial query vector, the characters embedding matrix S and the matched words $\{w_{i1}, w_{i2}, \ldots, w_{iN_i}\}$ of the character x_i as the external memory, where N_i is the number of matched words. Since a candidate character may match multiple words in a lexicon and one-pass attention calculation may not accurately learn lexical knowledge, memory network is exploited to perform a deep reasoning process to highlight the correct lexical knowledge. After each pass, we need to update the query vector for the next pass. Therefore, the memory network contains two phases: ***attention calculation*** and ***update mechanism***.

Attention Calculation: During each pass, the query vector is the output of the former pass. We use attention to model the relationship between the query vector and the matched words. At pass k, the attention calculation can be formulated as follows:

$$\begin{aligned} e_{it}^{k} &= w_{it}^{T} m_i^{k-1} \\ \alpha_{it}^{k} &= \frac{\exp(e_{it}^{k})}{\sum_{j=1}^{N_i} \exp(e_{ij}^{k})} \end{aligned} \tag{2}$$

where m_i^{k-1} denotes the output of pass $k-1$. We treat the candidate character embedding c_i as m_i^0.

Update Mechanism: After calculating the attention, we can obtain the memory state at the current pass:

$$u_i^k = \sum\nolimits_{t=1}^{N_i} \alpha_{it}^{k} w_{it} \tag{3}$$

We update the query vector by taking the former pass output and memory state of current pass into consideration for the next pass:

$$m_i^k = \mathrm{Relu}(W_m[m_i^{k-1} : u_i^k] + b_m) \tag{4}$$

where [:] is the concatenation operation. $W_m \in \mathbb{R}^{d_w \times 2d_w}$ and $b_m \in \mathbb{R}^{d_w}$ are trainable parameters. We use the output of the last pass as the lexical knowledge of the character x_i, denoted as w_i.

Empirically, different reasoning passes may obtain different performances (Luo et al., 2018). We assume that less reasoning passes are enough to tackle simple sentences than complicated sentences. However, conventional memory network cannot perform adaptive and proper deliberation passes according to the complexity of the input sentence. Therefore, we utilize reinforcement learning to automatically control the reasoning passes of the memory network. We will introduce ***state***, ***action*** and ***reward*** as follows:

State: We use the sentence embedding s' as the state. After getting the characters embedding matrix S, we perform the max-pooling operation and treat the result as the sentence embedding:

$$s' = \mathrm{Maxpooling}(S) \tag{5}$$

Action: We regard the reasoning pass as the action $a \in \{1, 2, \ldots, N\}$, where N is the maximal pass. We sample the value of a by a policy network $\pi_\Theta(a|s')$, which can be formulated as follows:

$$\pi_\Theta(a|s') = \mathrm{Softmax}(W_p s' + b_p) \tag{6}$$

where $W_p \in \mathbb{R}^{N \times d_c}$ and $b_p \in \mathbb{R}^{N}$ are trainable parameters. $\Theta = \{W_p, b_p\}$.

Reward: We can obtain a terminal reward after finishing the final prediction. In this work, we use the F1 score of each sentence as the reward r.

Given T training instances, the objective function of policy network is defined as :

$$J_1 = \sum\nolimits_{i=1}^{T} \log\pi_\Theta(a_{(i)}|s'_{(i)}) r_{(i)} \tag{7}$$

where $a_{(i)}$, $s'_{(i)}$ and $r_{(i)}$ are the action, state and reward of the training instance i, respectively. We use the policy gradient method to learn the parameter set Θ.

3.2 BiLSTM Encoding Layer

After obtaining character embeddings and lexical knowledge, we concatenate them as the input of the encoding layer. Long short term memory (LSTM) is a variant of recurrent neural network (RNN), which is designed to address the gradient vanishing and exploding problems in RNN via introducing gate mechanism and memory cell. In order to incorporate information from both sides of sequence, we use BiLSTM to extract features. The hidden state of BiLSTM can be defined as follows:

$$h_i = [\overrightarrow{h}_i : \overleftarrow{h}_i] \tag{8}$$

where $\overrightarrow{h}_i \in \mathbb{R}^{d_h}$ and $\overleftarrow{h}_i \in \mathbb{R}^{d_h}$ are the hidden states at position i of the forward and backward LSTM, respectively.

3.3 Hierarchical Tagging Layer

In the hierarchical tagging layer, we exploit the LSTM as the first tagging module named as TLSTM and the CRF as the second tagging module.

The First Tagging Module: TLSTM When detecting the tag of character x_i, the inputs of the first tagging module are: h_i from the BiLSTM encoding layer, former hidden state $\hat{h}_{i-1}$, and former predicted tag vector $\widehat{T}_{i-1}$. Formally, the TLSTM can be written precisely as follows:

$$
\begin{aligned}
\begin{bmatrix} i_i \\ o_i \\ f_i \\ \tilde{c}_i \end{bmatrix} &= \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W_d^T \begin{bmatrix} h_i \\ \hat{h}_{i-1} \\ \widehat{T}_{i-1} \end{bmatrix} + b_d \right) \\
\hat{c}_i &= \hat{c}_{i-1} \odot f_i + \tilde{c}_i \odot i_i \\
\hat{h}_i &= o_i \odot \tanh(\hat{c}_i) \\
\widehat{T}_i &= W_{td}\hat{h}_i + b_{td}
\end{aligned} \tag{9}
$$

where i, f, o are the input gate, forget gate and output gate, respectively. $\widehat{T}$ is the predicted tagging vector.

Tagging Attention Module: T-Attention Tagging attention aims to dynamically leverage the hidden states and preliminary predictions of the TLSTM. $\widehat{H} = \{\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_n\}$ and $T_{raw} = \{\widehat{T}_1, \widehat{T}_2, \ldots, \widehat{T}_n\}$ denote the hidden states and preliminary predictions of the TLSTM, respectively. The attention is expressed as follows:

$$
\begin{aligned}
\hat{h}_{di} &= [\hat{h}_i : \widehat{T}_i] \\
m_i &= u_d^T \tanh(W_{da}\hat{h}_{di} + b_{da}) \\
\alpha_i &= \frac{\exp(m_i)}{\sum_{j=1}^{n} \exp(m_j)} \\
r_i &= \tanh(\sum\nolimits_{j=1}^{n} \alpha_j \hat{h}_{dj})
\end{aligned} \tag{10}
$$

where $u_d \in \mathbb{R}^{d_{da}}$ is the context vector, which is randomly initialized and learned during the training process (Yang et al., 2016b). r_i denotes the representation of the hidden states and preliminary predictions of the TLSTM.

The Second Tagging Module: CRF $H = \{h_1, h_2, \ldots, h_n\}$ and $R = \{r_1, r_2, \ldots, r_n\}$ denote the outputs of BiLSTM encoding layer and tagging attention module, respectively, which are concatenated as the input of the CRF module, denoted as $H_c = \{h_{c1}, h_{c2}, \ldots, h_{cn}\}$.

Given a sentence $s = \{x_1, x_2, \ldots, x_n\}$ with a final predicted tag sequence $y = \{y_1, y_2, \ldots, y_n\}$, the CRF tagging process is formalized as follows:

$$
\begin{aligned}
o_i &= W_o h_{ci} + b_o \\
s(s, y) &= \sum\nolimits_{i=1}^{n} (o_{i,y_i} + T_{y_{i-1},y_i}) \\
y^* &= \arg\max_{y \in Y_s} s(s, y)
\end{aligned} \tag{11}
$$

where o_{i,y_i} is the score of the y_i-th tag of the character x_i. T denotes the transition matrix which defines the scores of two successive labels. Y_s represents all candidate tag sequences for given sentence s. We use the Viterbi algorithm to get the final best-scoring tag sequence y^*.

3.4 Training

The probability of the ground-truth tag sequence $\bar{y}$ can be computed by:

$$
p(\bar{y}|s) = \frac{\exp(s(s, \bar{y}))}{\sum_{\tilde{y} \in Y_s} \exp(s(s, \tilde{y}))} \tag{12}
$$

Dataset	# Train sentence	# Dev sentence	# Test sentence
MSRA	41.4k	4.6k	4.0k
OntoNotes 4	22.7k	3.9k	2.7k
WeiboNER	1.4k	0.27k	0.27k

Table 1: Statistics of the datasets.

Given a set of manually labeled training data $\{s^{(i)}, \bar{y}^{(i)}\}|_{i=1}^{T}$, the objective function of the tagging layer can be defined as follows:

$$J_2 = \sum_{i=1}^{T} \mathrm{log}p(\bar{y}^{(i)}|s^{(i)}) \tag{13}$$

The objective function of the whole model is listed as follows:

$$J = \lambda J_1 + J_2 \tag{14}$$

As the adaptive multi-pass memory network and hierarchical tagging layer are correlated mutually, we train them jointly. We pre-train the model before the joint training process starts using the objective function J_2. Then, we jointly train the model using the objective function J.

4 Experiments

4.1 Datasets

We evaluate our proposed model on three widely used datasets, including MSRA (Levow, 2006), OntoNotes 4 (Weischedel et al., 2011) and WeiboNER (Peng and Dredze, 2015; He and Sun, 2016). The MSRA dataset contains three entity types (person, location and organization). The OntoNotes 4 dataset annotates 18 named entity types. In this work, we use the four most common named entity types (person, location, organization and geo-political), as same as previous studies (Che et al., 2013; Zhang and Yang, 2018). The WeiboNER dataset is annotated with four entity types (person, location, organization and geo-political), including named entities and nominal mentions.

For MSRA dataset, we use the same data split as Dong et al. (2016). Since MSRA dataset does not have development set, we sample 10% data of training set as development set. For OntoNotes 4 dataset, we take the same data split as Che et al. (2013) and Zhang and Yang (2018). For WeiboNER dataset, we use the same training, development and testing splits as Peng and Dredze (2015) and He and Sun (2016). The details of the datasets are shown in Table 1.

4.2 Evaluation Metrics and Experimental Settings

For evaluation metrics, we use the Micro averaged Precision (P), Recall (R) and F1 score as metrics in our experiments, as the same as previous works (Che et al., 2013; Zhang and Yang, 2018), which are calculated per-span.

Hyper-parameters tuning is made through adjustments according to the performance on the development sets. The dimension of character embedding d_c is 100. The size of word embedding d_w is 50. The hidden size of LSTM d_h is set to 300. The dropout rate is 0.3. The λ is set to 0.1. Adam (Kingma and Ba, 2014) is used for optimization, with an initial learning rate of 0.001. The character embeddings used in this work are pre-trained on Chinese Wikipedia corpus by using word2vec toolkit (Mikolov et al., 2013). We use the same lexicon as Zhang and Yang (2018).

4.3 Compared with State-of-the-art Methods

4.3.1 Evaluation on MSRA

We compare our proposed model with previous methods on MSRA dataset. The results are listed in Table 2[1]. Zhang et al. (2006) leverage rich handcrafted features for Chinese NER. The model gives very competitive performance. Dong et al. (2016) incorporate radical features into neural LSTM+CRF model,

[1] * in Table 2, 3 and 4 denotes that a model exploits additional labeled data.

Models	P(%)	R(%)	F1(%)
Chen et al. (2006)	91.22	81.71	86.20
Zhou et al. (2006)	88.94	84.20	86.51
Zhang et al. (2006)*	92.20	90.18	91.18
Zhou et al. (2013)	91.86	88.75	90.28
Dong et al. (2016)	91.28	90.62	90.95
Zhang and Yang (2018)	93.57	92.79	93.18
Cao et al. (2018)	91.73	89.58	90.64
AMMNHT	**93.62**	**92.96**	**93.29**

Table 2: Experimental results on MSRA dataset.

Models	P(%)	R(%)	F1(%)
Che et al. (2013)*	77.71	72.51	75.02
Wang et al. (2013)*	76.43	72.32	74.32
Yang et al. (2016a)	65.59	71.84	68.57
Yang et al. (2016a)*	72.98	80.15	76.40
Zhang and Yang (2018)	76.35	71.56	73.88
AMMNHT	**76.51**	**71.70**	**74.03**

Table 3: Experimental results on OntoNotes 4 dataset. The first and second blocks list word-based methods and character-based method, respectively.

achieving the F1 score of 90.95%. Cao et al. (2018) achieve competitive performance via adversarial transfer learning method. We can observe that our proposed model gets significant improvements over previous state-of-the-art methods. For example, compared with the latest model (Cao et al., 2018) which uses additional CWS training data, our proposed method improves the F1 score from 90.64% to 93.29%. Moreover, compared with Zhang and Yang (2018), our model also greatly improves the performance. We also perform a t-test ($p < 0.01$), which indicates that our method outperforms all of the compared methods.

4.3.2 Evaluation on OntoNotes

We evaluate our proposed model on OntoNotes 4 dataset. Table 3 lists the results of our proposed model and previous state-of-the-art methods. In the first two blocks, we give the performance of word-based and character-based methods for Chinese NER, respectively. Based on the gold segmentation, Che et al. (2013) propose an integer linear program based inference algorithm with bilingual constraints for NER. The model gives a 75.02% F1 score. With gold word segmentation, the word-based models achieve better performance than the character-based model. This demonstrates that word boundaries information is useful for Chinese NER task. Compared with the character-based method (Zhang and Yang, 2018), our model improves the F1 score from 73.88% to 74.03%. Compared with the word-based method (Wang et al., 2013), our model also achieves better performance. The great improvements over previous state-of-the-art methods demonstrate the effectiveness of our proposed model.

4.3.3 Evaluation on WeiboNER

We compare our proposed model with the latest models on WeiboNER dataset. The experimental results are shown in Table 4, where NE, NM and Overall denote F1 scores for named entities, nominal entities and both, respectively. Peng and Dredze (2016) propose a model that jointly performs Chinese NER and CWS task, which achieves better results than Peng and Dredze (2015) for named entity, nominal mention and overall. Recently, Zhang and Yang (2018) propose a lattice LSTM model to exploit word sequence information. The model gives a 58.79% F1 score on overall performance. It can be observed that our proposed model achieves great improvements compared with previous methods. For example, compared

Models	NE	NM	Overall
Peng and Dredze (2015)	51.96	61.05	56.05
Peng and Dredze (2016)*	**55.28**	62.97	58.99
He and Sun (2016)	50.60	59.32	54.82
He and Sun (2017)*	54.50	62.17	58.23
Zhang and Yang (2018)	53.04	62.25	58.79
Cao et al. (2018)	54.34	57.35	58.70
AMMNHT	54.09	62.43	**59.04**

Table 4: F1 scores (%) on WeiboNER dataset.

Models	MSRA	OntoNotes	WeiboNER
BiLSTM+CRF	89.13	63.17	55.84
BiLSTM+CRF+AMMN	92.40	73.11	58.65
BiLSTM+HT	90.53	64.14	56.55
AMMNHT	**93.29**	**74.03**	**59.04**

Table 5: F1 score (%) of AMMNHT and its simplified models on MSRA, OntoNotes 4 and WeiboNER datasets, respectively.

with the lattice LSTM model, our proposed model improves the F1 score from 53.04% to 54.09% for named entity. It proves the effectiveness of our proposed model.

4.4 Ablation Experiment

To investigate the effectiveness of adaptive multi-pass memory network and hierarchical tagging mechanism, we conduct the ablation studies. The baseline and simplified models of the proposed model are detailed as follows: (1) **BiLSTM+CRF**: The model is exploited as the strong baseline in our experiment. (2) **BiLSTM+CRF+AMMN**: The model integrates lexical knowledge from a lexicon via adaptive multi-pass memory network. (3) **BiLSTM+HT**: The model exploits the BiLSTM to extract features and uses the hierarchical tagging layer to predict labels.

From the results listed in Table 5, we have several important observations as follows:

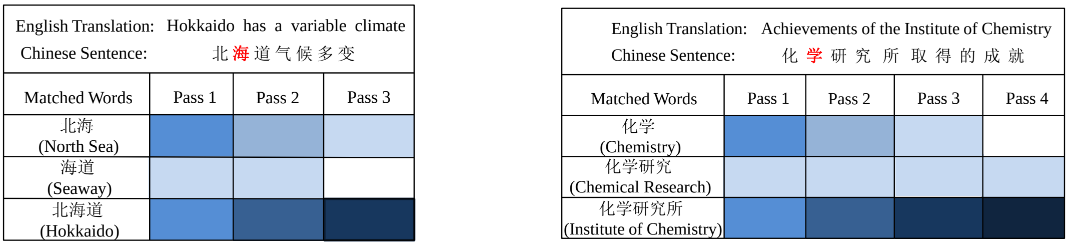
- **Effectiveness of Adaptive Multi-pass Memory Network.** We observe that the BiLSTM+CRF+AMMN model outperforms the BiLSTM+CRF on these three datasets. For example, compared with the baseline, it improves the F1 score from 89.13% to 92.40% on MSRA dataset. Compared the AMMNHT with BiLSTM+HT, we can find similar phenomenon. The great improvements demonstrate the effectiveness of the adaptive multi-pass memory network.

- **Effectiveness of Hierarchical Tagging Mechanism.** Compared with the BiLSTM+CRF, the BiLSTM+HT model improves the performance, achieving 1.40% improvements of F1 score on MSRA dataset. Moreover, the AMMNHT also outperforms the BiLSTM+CRF+AMMN. The great improvements indicate the hierarchical tagging mechanism is very effective for Chinese NER task.

- **Effectiveness of Adaptive Multi-pass Memory Network and Hierarchical Tagging Mechanism.** We observe that the proposed model AMMNHT achieves better performance than its simplified models on the three datasets. For example, compared with BiLSTM+CRF, the AMMNHT model improves the F1 score from 89.13% to 93.29% on MSRA dataset. It indicates that simultaneously exploiting the adaptive multi-pass memory network and hierarchical tagging mechanism is also very effective.

4.5 Adaptive Multiple Passes Analysis

To better illustrate the influence of multiple passes and adaptive multi-pass memory network, we give the results of fixed multiple passes and adaptive multi-pass memory network in Table 6. The results



(a) Attention visualization of AMMN when learning lexical knowledge for the candidate character "海 (sea)".

(b) Attention visualization of AMMN when learning lexical knowledge for the candidate character '学 (subject)".

Figure 3: Two examples of attention weights in adaptive multi-pass memory network. The reasoning passes are 3 and 4, respectively. Darker colors mean that the attention weight is higher.

Pass	MSRA	OntoNotes	WeiboNER
1	92.64	72.87	58.52
2	92.96	73.50	58.83
3	93.14	73.77	58.74
4	93.12	73.85	58.34
5	93.03	73.46	58.13
Adaptive	**93.29**	**74.03**	**59.04**

Table 6: F1 score (%) of different passes from 1 to 5 and adaptive passes on the test sets. It shows suitable reasoning passes of memory network can boost the performance.

show that multiple passes operation performs better than one pass. The reason is that multiple passes reasoning can help to highlight the most appropriate matched words. The cases in Figure 3 show that the deep deliberation can recognize the correct lexical knowledge by enlarging the attention gap between correct matched words and incorrect ones. When the number of passes is too large, the performance stops increasing or even decreases due to over-fitting. In contrast to the fixed multiple passes memory network, the adaptive multi-pass memory network has 0.21% improvements of F1 score on the WeiboNER dataset. Furthermore, the two examples in Figure 3 show that adaptive multi-pass memory network can choose suitable reasoning passes according to the complexity of the input sentence, which also demonstrates the effectiveness of the adaptive multi-pass memory network.

5 Conclusion

In this paper, we propose an adaptive multi-pass memory network to incorporate lexical knowledge from a lexicon for Chinese NER task which can adaptively choose suitable reasoning passes according to the complexity of each sentence. Besides, we devise a hierarchical tagging layer to capture tag dependencies in the whole sentence. The adaptive memory network and hierarchical tagging mechanism can be easily applied to similar tasks involving multi-pass reasoning and decoding process, such as knowledge base question answering and machine translation. Experimental results on three widely used datasets demonstrate that our proposed model outperforms previous state-of-the-art methods.

Acknowledgments

This work is supported by the Natural Key R&D Program of China (No.2017YFB1002101), the National Natural Science Foundation of China (No.61533018，No.61922085 , No.61976211, No.61806201) and the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006). This work isalso supported by Beijing Academy of Artificial Intelligence (BAAI2019QN0301), the CCF-Tencent Open Research Fund and independent research project of National Laboratory of Pattern Recognition.

References

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of EMNLP*, pages 724–731.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for chinese named entity recognition with self-attention mechanism. In *Proceedings of EMNLP*.

Wanxiang Che, Mengqiu Wang, Christopher D Manning, and Ting Liu. 2013. Named entity recognition with bilingual constraints. In *Proceedings of NAACL-HLT*, pages 52–62.

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*, pages 213–216.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of ACL*, pages 167–176.

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 *Natural Language Understanding and Intelligent Applications*, pages 239–250.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuan-Jing Huang. 2019. A lexicon-based graph neural network for chinese ner. In *EMNLP-IJCNLP*.

Hangfeng He and Xu Sun. 2016. F-score driven max margin neural network for named entity recognition in chinese social media. *arXiv preprint arXiv:1611.04234*.

Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Proceedings of AAAI*.

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

Hideki Isozaki and Hideto Kazawa. 2002. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

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

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*, pages 108–117.

Yanan Lu, Yue Zhang, and Dong-Hong Ji. 2016. Multi-prototype chinese character embedding. In *Proceedings of LREC*.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. In *Proceedings of ACL*, pages 2473–2482.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of ACL*, pages 1064–1074.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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

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

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of ACL*, pages 1756–1765.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for chinese named entity recognition via collaborative graph network. In *EMNLP-IJCNLP*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Proceedings of NeurIPS*, pages 2440–2448.

Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. MIT press Cambridge.

Mengqiu Wang, Wanxiang Che, and Christopher D Manning. 2013. Effective bilingual constraints for semi-supervised learning of named entity recognizers. In *Proceedings of AAAI*.

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

Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, and Gerhard Weikum. 2013. Robust question answering over the web of linked data. In *Proceedings of CIKM*, pages 1107–1116.

Jie Yang, Zhiyang Teng, Meishan Zhang, and Yue Zhang. 2016a. Combining discrete and neural features for sequence labeling. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 140–154.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016b. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. In *Proceedings of ACL*, pages 1554–1564.

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

Yuan Zhang, Hongshen Chen, Yihong Zhao, Qun Liu, and Dawei Yin. 2018. Learning tag dependencies for sequence tagging. In *Proceedings of IJCAI*, pages 4581–4587.

Junsheng Zhou, Liang He, Xinyu Dai, and Jiajun Chen. 2006. Chinese named entity recognition with a multi-phase model. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*.

Junsheng Zhou, Weiguang Qu, and Fen Zhang. 2013. Chinese named entity recognition via joint identification and categorization. *Chinese journal of electronics*.