SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2020, Vol. 63 202102:1–202102:15
https://doi.org/10.1007/s11432-020-2982-y

Hierarchical LSTM with char-subword-word tree-structure representation for Chinese named entity recognition

Chen GONG, Zhenghua LI*, Qingrong XIA, Wenliang CHEN & Min ZHANG

School of Computer Science and Technology, Soochow University, Suzhou 215006, *China*

Received 22 April 2020/Accepted 14 May 2020/Published online 16 September 2020

Abstract Chinese named entity recognition (CNER) aims to identify entity names such as person names and organization names from Chinese raw text and thus can quickly extract the entity information that people are concerned about from large-scale texts. Recent studies attempt to improve performance by integrating lexicon words into char-based CNER models. These existing studies, however, usually focus on leveraging the context-free words in lexicon without considering the contextual information of words and subwords in the sentences. To address this issue, in addition to utilizing the lexicon words, we further propose to construct a hierarchical tree structure representation composed of characters, subwords and context-aware predicted words from segmentor to represent each sentence for CNER. Based on the tree-structure representation, we propose a hierarchical long short-term memory (HiLSTM) framework, which consists of hierarchical encoding layer, fusion layer and CRF layer, to capture linguistic knowledge at different levels. On the one hand, the interactions within each level help to obtain the contextual information. On the other hand, the propagations from the lower-levels to the upper-levels can provide additional semantic knowledge for CNER. Experimental results on three widely used CNER datasets show that our proposed HiLSTM model achieves significant improvement over several strong benchmark methods.

Keywords natural language processing, named entity recognition, representation learning, neural networks

Citation Gong C, Li Z H, Xia Q R, et al. Hierarchical LSTM with char-subword-word tree-structure representation for Chinese named entity recognition. Sci China Inf Sci, 2020, 63(10): 202102, https://doi.org/10.1007/s11432-020-2982-y

1 Introduction

As a fundamental task in natural language processing (NLP), the purpose of named entity recognition (NER) is to identify named entities from raw texts, such as person names, organization names, and location names. Named entities are indispensable for many down-stream NLP applications, such as information retrieval [1], relation extraction [2], and question answering [3]. For example, in the medical field, identifying entities such as disease names, symptom names, and medicine names from the electronic medical records allows doctors to quickly understand the health status and the treatment of patients, which is helpful for decision-making [4]; further extracting the relations between the entities can be used to study the similarities between different patients and to find the contraindications to medicine use [5]; NER is also an essential pre-processing step for online medical question answering [6].

Traditional machine learning methods for NER, such as hidden Markov models (HMMs) [7,8], support vector machines (SVM) [9], and conditional random fields (CRFs) [10], usually rely on hand-crafted

* Corresponding author (email: zhli13@suda.edu.cn)

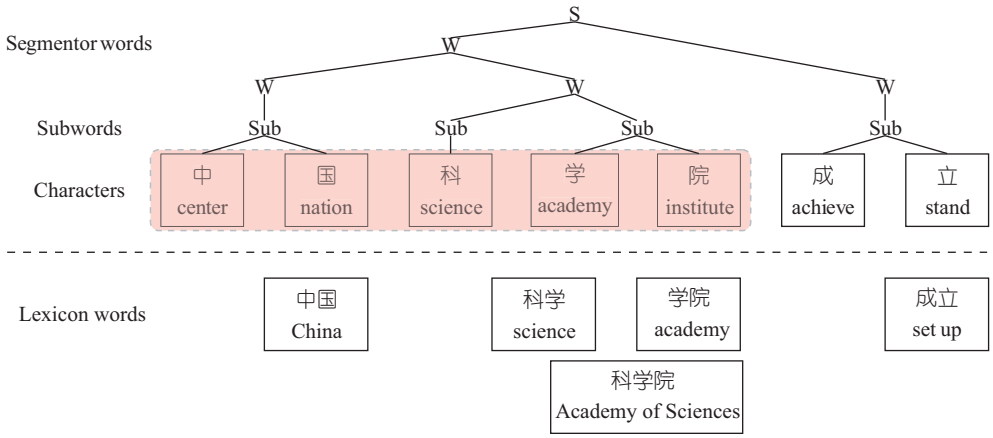

Figure 1 (Color online) The lexicon words (below) and char-subword-word tree-structure representation (upper) of an example.

discrete features, which is expensive to design. The selection of features has a great impact on the performance of entity recognition. With the development of deep learning, neural network based NER methods have received widespread attention from researchers. Currently, for English NER, the models with BiLSTM-CRF (bidirectional long short-term memory-conditional random fields) architecture have achieved the state-of-the-art results [11–13]. Inspired by the success of using BiLSTM-CRF in English NER, researchers propose two types of BiLSTM-CRF-based models for Chinese NER (CNER), i.e., character-based (char-based) models and word-based models.

Dong et al. [14] first employed char-based BiLSTM-CRF neural architecture for CNER by predicting the label of each character in the sentence and achieved good performance, which is free from hand-crafted features and does not require word segmentation (WS). However, their char-based method ignores the useful semantic and boundary information in Chinese words. As the basic semantic units in Chinese language utterances, words are informative for CNER model supervision. The word boundary information is also of great help for CNER due to the strong correlation between word and named entity boundaries. One simple and intuitive way to incorporate word information into CNER is the word-based method [15]. For the reason that there are no obvious delimiters between words in Chinese writing system, word-based methods need to perform WS first, before applying word sequence labeling to identify named entities. This two-stage pipeline, however, can cause error propagation problems, because incorrect WS will probably lead to CNER errors. Therefore, in order to make use of the word information without suffering from the error propagation problem, many recent studies focus on enhancing CNER by integrating word lexicon into the char-based models [15–18]. Zhang and Yang [15] and Liu et al. [17] successfully integrated word lexicon knowledge into char-based BiLSTM-CRF models. The basic idea is to first extract all the matched words according to a lexicon. For clarity, we refer to such matched words as lexicon words. For example, as shown in Figure 1 (bottom), given the sentence "中国科学院成立 (the Chinese Academy of Sciences was established)", lexicon words "中国 (China)", "科学 (science)", "学院 (academy)", "科学院 (Academy of Sciences)" and "成立 (set up)" are extracted by matching the sentence with a large automatically-obtained lexicon. After obtaining the lexicon words, Zhang and Yang [15] and Liu et al. [17] incorporated these words into char-based CNER models by leveraging the structure of lattice LSTM and word-character LSTM (WC-LSTM), respectively.

Although the lexicon-enhanced char-based CNER models have achieved good performance, the lexicon words are context-free thus the context-aware word information for disambiguation is ignored. To overcome this limitation, in addition to utilizing the lexicon words, we further use the word sequence predicted by the word segmentor as extra context-aware word information. We denote these predicted words as segmentor words. Segmentor words are produced by the WS model which is trained on large-scale manually annotated datasets. They take advantage of the context-aware information in the manually annotated data thus are helpful for disambiguation. In view of the diverse segmentation granularities for

Chinese WS caused by different linguistic perspectives, we adopt our previously proposed multi-grained word segmentor [19] to generate words of all different granularities by exploiting large-scale heterogeneous datasets with different segmentation criteria. Taking the sentence "中国科学院成立 (the Chinese Academy of Sciences was established)" in Figure 1 as an example, the multi-grained word segmentor splits the sentence into 4 words of different granularities: "中国 (China)", "科学院 (Academy of Sciences)", "中国科学院 (Chinese Academy of Sciences)", and "成立 (set up)". Among them, the word sequence "中国 (China)/科学院 (Academy of Sciences)/成立 (set up)" has the finest granularity, while the word sequence "中国科学院 (Chinese Academy of Sciences)/成立 (set up)" has the coarsest granularity. We define the words in these two word sequences as fine-grained words and coarse-grained words respectively. These multi-grained segmentor words (denoted as "W" in Figure 1) can be useful due to their potential complementarity: fine-grained words reduce data sparseness, whereas coarse-grained words reserve more semantics and tend to have the same boundaries as entities. For instance, the boundaries of the organization named entity "中国科学院 (Chinese Academy of Sciences)" and the coarse-grained word "中国科学院 (Chinese Academy of Sciences)" are the same. In this study, we represent these segmentor words as extra features for char-based CNER models, which can avoid the error propagation problem in the word-based method.

Moreover, we also split the fine-grained segmentor words into subwords (denoted as "Sub" in Figure 1) and encode the subwords in the char-based CNER model to alleviate the unknown word problem by assuming a word's meaning can be reconstructed from its parts [20] and offer additional semantic knowledge to CNER. For example, the meaning of "科学院 (Academy of Sciences)" can be obtained by composing the meaning of its subwords "科 (science)" and "学院 (academy)".

In this paper, we propose to construct a char-subword-word representation with tree structure by making full use of characters, subwords, and segmentor words. The upper part of Figure 1 shows an example of the tree-structure representation, where "Sub", "W", and "S" represent subwords, segmentor words, and sentence, respectively. Based on the representation, we further propose a hierarchical LSTM (HiLSTM) framework to capture and learn from the information in the char-subword-word representation. Extensive experiments on three widely used datasets are conducted to determine the effectiveness of the proposed HiLSTM model. The main contributions of our paper can be summarized as follows:

• We propose to construct a char-subword-word hierarchical tree structure composed of characters, subwords and segmentor words to represent each sentence for CNER, in order to take full advantage of the information from different linguistic levels. To our knowledge, we are the first to use subwords and multi-grained segmentor words in neural network char-based CNER.

• We propose an effective HiLSTM model to capture and characterize the char-subword-word representation for CNER. On the one hand, within each level of the proposed HiLSTM model, contextual information is gained by propagating the information of each sentence in both forward and backward directions and is helpful for disambiguation. On the other hand, the propagations from the lower character level to the upper subword and word levels can deliver the semantic knowledge of lower levels to upper levels.

• We conduct experiments on three widely used CNER datasets and give detailed analyses to verify the effectiveness of our proposed HiLSTM model. Experimental results show that the proposed char-subword-word tree-structure representation can consistently improve the CNER models and our HiLSTM model achieves the state-of-the-art results on OntoNotes, Weibo NER, and MSRA datasets.

The remaining parts of the paper are organized as follows. Section 2 summarizes the related work. Section 3 describes the task of NER and then introduces our proposed char-subword-word tree-structure representation and the hierarchical neural network architecture. Section 4 presents the experimental results and gives detailed analyses to verify the effectiveness of our proposed HiLSTM model. Finally, we conclude our work in Section 6.

2 Related work

Typically, NER is regarded as a sequence labeling problem. Given an input sentence composed of a sequence of n tokens, the goal of NER is to train a classification model to classify each token in the sequence into a specific label to indicate whether the token is included in an entity. Early works in NER mostly focus on using traditional machine learning algorithms by first manually designing discrete features to represent context and then employing different machine learning models to recognize named entities based on the hand-crafted features. Saito et al. [7] designed three language features (i.e., character type and word length, orthography and spacing, and word candidate generation) and introduced a multi-language NER model based on HMMs. Solorio et al. [9] selected part-of-speech (POS) tags and morphological information (such as capitalization) as features, and then used these features to train an SVM classifier for NER. Mccallum et al. [10] proposed an automated feature induction method for CRFs in NER, leading to improved performance and decreasing count of features. However, the new features generated during feature induction are still based on hand-crafted observation tests which cost a lot. Although the above traditional machine learning methods have made progress in NER, they heavily rely on expensive human efforts for feature selection and are also limited in contextual representations.

Compared with traditional machine learning methods, deep learning methods have much stronger capability in automatically capturing contextual representations and solving non-linear problems. Therefore, recent NER models have shifted to neural network architectures. Collobert et al. [21] first successfully used deep learning methods to handle NER based on convolutional neural networks (CNN). However, CNN is not sensitive to position information. For example, although the contexts which contain the same words but with different word orders usually differ in semantics, they may have similar CNN representations. The loss of the useful position information limits the representation ability of CNN to some extent. In contrast, the BiLSTM-CRF architecture is effective in capturing position information in sequence data and characterizing long-distance contextual representations. Therefore, BiLSTM-CRF-based models have been widely used and achieved the state-of-the-art performance in English NER [11–13].

Different from most Indo-European languages such as English, Chinese has no delimiters between words, but leveraging word information for CNER can be helpful due to the rich semantic knowledge and boundary information provided by words [22]. Therefore, the ambiguity of Chinese word boundaries brings challenge to CNER. One intuitive way of incorporating word information into CNER is the word-based approach. This approach performs Chinese word segmentation (CWS) first and builds CNER models based on the automatic CWS outputs by predicting the label of each word in the CWS outputs [15]. A limitation of word-based approach is that the segmentation error in automatic CWS outputs can be further propagated into CNER, leading to degraded CNER performance.

To address the error propagation issue, several works manage to integrate word information to char-based model. Zhao et al. [23], Peng et al. [24], and He et al. [25] treated word segmentation as soft features for CNER. The basic idea is to concatenate the word segmentation label (i.e., B (begin), I (inside), E (end), and S (single)) embedding with the character embedding, so as to augment the representation of the model input with word boundary information, but the semantic of words are ignored in their soft feature method. Instead of only considering explicit word boundary features, Peng and Dredze [26] and Cao et al. [27] proposed an adversarial multi-task learning (MTL) framework to enhance CNER with implicit task-shared features in WS and CNER tasks by training WS and CNER jointly. However, the MTL model requires additional manually annotated WS data. Recently, many studies pay attention to improving CNER performance using lexicon words and do not need additional manually labeled data. Zhang and Yang [15] successfully incorporated matched lexicon word representations into the hidden state of char-based model using a lattice LSTM structure, but face the issue of inefficiency due to the complex model architecture. To improve efficiency, Liu et al. [17] proposed a WC-LSTM to exploit lexicon knowledge by concatenating lexicon word representations with the character embeddings. Gui et al. [16] and Sui et al. [18] introduced graph neural network (GNN) to capture lexicon information. Although the above lexicon-enhanced CNER models have achieved good performance, their common limitation is that they only use context-free lexicon words to obtain word information, ignoring the context-aware word

information which is helpful for disambiguation.

In this study, in addition to leveraging lexicon words, we further use the segmentor words produced by the multi-grained segmentor [19] to extract context-aware word information. Moreover, we also propose to encode subwords in neural char-based CNER models to gain more semantic knowledge and alleviate the unknown word problem inspired by the success of utilizing subwords in machine translation [28]. We construct a char-subword-word tree structure composed of characters, subwords, and segmentor words to represent each sentence for CNER and propose a hierarchical network architecture based on the representation. To our knowledge, we are the first to design a tree-structure representation for mixed characters, subwords and segmentor words, and the first to use a hierarchical network architecture for CNER.

3 Our proposed HiLSTM model

In this section, we introduce the proposed HiLSTM model enhanced with char-subword-word tree-structure representation. First, we explain the problem of NER. Then, we describe the construction of the char-subword-word tree structure which is used to represent each sentence for CNER. After that, we introduce the framework of our HiLSTM model and the details of each module, including a hierarchical encoding layer, a fusion layer and a CRF layer.

3.1 Problem description

NER can be formulated as a sequence labeling problem, where entity boundary and entity category are jointly predicted.

Formally, given a dataset $\boldsymbol{D} = \{(\mathbf{sent}_d, \boldsymbol{y}_d)|1 \leqslant d \leqslant N\}$, where $\mathbf{sent}_d$ represents the d-th sentence in the dataset, $\boldsymbol{y}_d$ is the named entity label sequence of $\mathbf{sent}_d$. The named entity label is composed of the entity boundary label and the entity category label (which are concatenated by a "-" symbol). For the entity boundary label, we adopt the BMESO tagging scheme, among which B, M, E respectively represent that the concerned character situates at the beginning, middle, and end position of an entity, S represents a single-character entity, and O means the concerned character is not included in the entity. The entity category labels such as "PER (person)", "LOC (location)", and "ORG (organization)" refer to the type of each entity. Taking the sentence {"中 (center)", "国 (nation)", "科 (science)", "学 (academy)", "院 (institute)", "成 (achieve)", "立 (stand)"} as an example, given "中国科学院 (Chinese Academy of Sciences)" is an organization named entity, the label sequence of this sentence is {B-ORG, M-ORG, M-ORG, M-ORG, M-ORG, O, O}. The goal of NER is to train a classification model that can classify each token in an unlabeled sentence into a named entity label.

3.2 Construction of char-subword-word tree-structure representation

To make full use of the information in multiple linguistic levels (i.e., character-level, subword-level, and word-level) for CNER, we represent the characters, subwords and words of each sentence in a unified manner by constructing a char-subword-word tree structure. Formally, we use $\mathbf{sent} = \{c_i|1 \leqslant i \leqslant n\}$ to represent a sentence composed of n characters. We denote the subsequence of **sent** which begins with character index b and ends with character index e as $c_{b,e}$. To fully leverage the linguistic information in different levels, we assign each character c_i with two types of representations, namely $\text{seg}(c_i)$ and $\text{sub}(c_i)$, referring to corresponding segmentor words and subwords for c_i respectively. To obtain the segmentor words $\text{seg}(c_i)$ for c_i, we first segment the sentence **sent** into words of different granularities (denoted as $\boldsymbol{w}$) using a multi-grained segmentor [19], and then pick all the words ends with character c_i from $\boldsymbol{w}$ to form $\text{seg}(c_i)$: $\text{seg}(c_i) = \{c_{b,i}|b \leqslant i, c_{b,i} \in \boldsymbol{w}\}$. To get the corresponding subword $\text{sub}(c_i)$ for c_i, we further split the fine-grained word (the word which has the least number of characters) in $\text{seg}(c_i)$ into subwords [20], and $\text{sub}(c_i)$ is the subword ends with c_i. Finally, we construct a char-subword-word tree structure to represent sentence, as illustrated in the upper part of Figure 1. We represent each character in the tree as a triple $(c_i, \text{sub}(c_i), \text{seg}(c_i))$. For example, the representation for the character c_5 = "院 (institute)" in

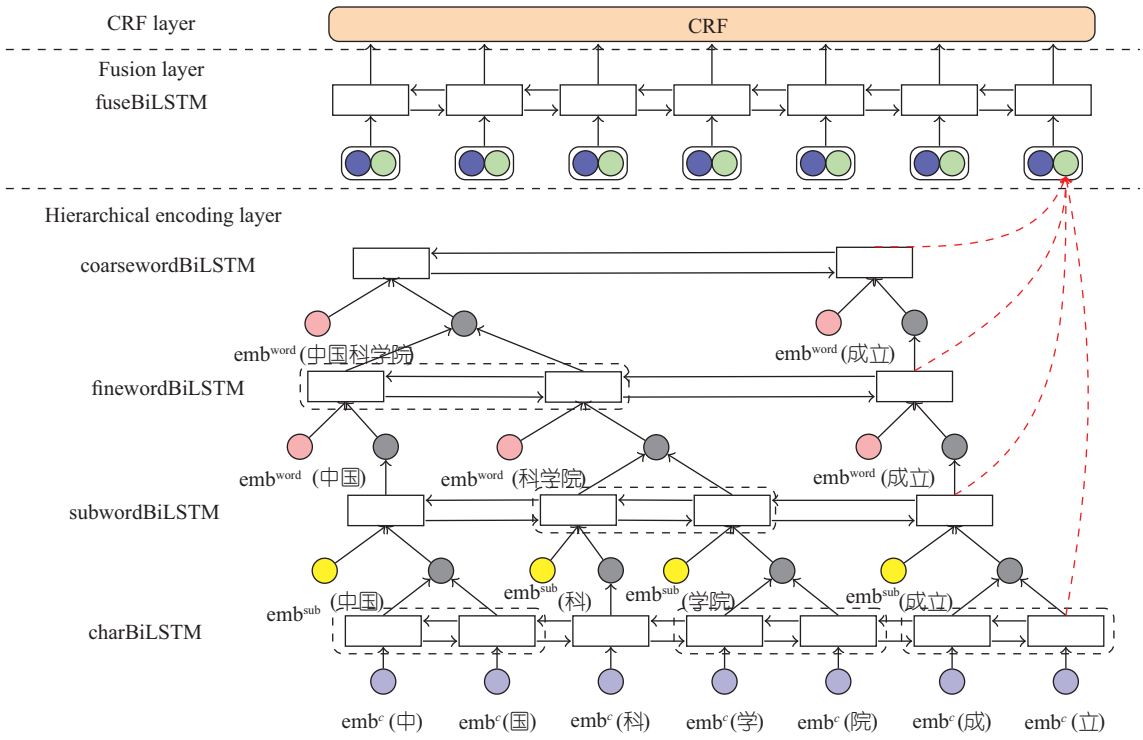

Figure 2 (Color online) The framework of our proposed HiLSTM model.

Figure 1 is (c_5, $c_{4,5}$, $\{c_{1,5}, c_{3,5}\}$) = ("院 (institute)", "学院 (academy)", {"中国科学院 (Chinese Academy of Sciences)", "科学院 (Academy of Sciences)"}).

3.3 Framework of HiLSTM

In order to capture and characterize the char-subword-word representation, we propose a hierarchical LSTM architecture for CNER thus helps to improve model performance by providing rich information from multiple linguistic levels. The framework of the proposed HiLSTM model is shown in Figure 2. It consists of three components, including a hierarchical encoding layer, a fusion layer and a CRF layer. First, the hierarchical encoding layer captures the rich contextual and semantic information in four levels, namely character-level, subword-level, fine-grained word-level, and coarse-grained word-level. Then, a fusion layer is used to fuse the four-level hierarchical LSTM hidden states with the char/subword/word embeddings. Finally, a standard CRF is employed on top of the fusion layer for training and decoding. We will describe each component in Subsections 3.4–3.6.

3.4 Hierarchical encoding layer

The hierarchical encoding layer contains four sub-layers, i.e., character-level, subword-level, fine-grained word-level, and coarse-grained word-level sub-layers. The propagations within each sub-layer help to gain contextual information, and the semantic knowledge of the lower sub-layers can be delivered to the upper sub-layers through the low-to-upper propagations.

Character-level encoding layer. For a given sentence **sent** = $\{c_i|1 \leqslant i \leqslant n\}$, each character c_i in the sentence is first mapped to a dense vector representation $\boldsymbol{x}_i^c$ by looking up the pre-trained character embedding matrix $\mathbf{emb}^c$:

$$\boldsymbol{x}_i^c = \mathbf{emb}^c(c_i). \tag{1}$$

Then we feed $\boldsymbol{x}_i^c$ into the character-level BiLSTM to capture the contextual information:

$$\boldsymbol{h}_i^c = \text{charBiLSTM}(\boldsymbol{x}_i^c). \tag{2}$$

Subword-level encoding layer. We denote the j-th subword in **sent** which begins with the b-th character and ends with the e-th character as $\text{sub}_j = c_{b,e}$. The input of the subword-level encoder contains both contextual representation and semantic representation. The contextual representation of each subword is built by combining the character-level BiLSTM outputs of its constituent characters. The semantic representation is obtained from a pre-trained subword embedding matrix $\mathbf{emb}^{\text{sub}}$. For each subword sub_j, we concatenate its contextual representation and semantic representation as the input of the subword-level BiLSTM:

$$\begin{aligned} \boldsymbol{x}_j^{\text{sub}} &= \text{avgpool}(\{\boldsymbol{h}_i^c | b \leqslant i \leqslant e\}) \oplus \mathbf{emb}^{\text{sub}}(\text{sub}_j), \\ \boldsymbol{h}_j^{\text{sub}} &= \text{subwordBiLSTM}(\boldsymbol{x}_j^{\text{sub}}), \end{aligned} \tag{3}$$

where avgpool means the average pooling operation.

Fine-grained word-level encoding layer. We define the word which has the minimum number of characters in $\text{seg}(c_i), c_i \in$ **sent** as fine-grained word. For the k-th fine-grained word which begins with the b-th subword and ends with the e-th subword, $\text{fine}_k = \text{sub}_{b,e}$, the input of the fine-grained word-level encoder is

$$\boldsymbol{x}_k^{\text{fine}} = \text{avgpool}(\{\boldsymbol{h}_j^{\text{sub}} | b \leqslant j \leqslant e\}) \oplus \mathbf{emb}^{\text{word}}(\text{fine}_k), \tag{4}$$

where $\mathbf{emb}^{\text{word}}$ is a pre-trained word embedding lookup table. We feed $\boldsymbol{x}_k^{\text{fine}}$ into the fine-grained word-level LSTM to form the final representation of each fine-grained word:

$$\boldsymbol{h}_k^{\text{fine}} = \text{finewordBiLSTM}(\boldsymbol{x}_k^{\text{fine}}). \tag{5}$$

Coarse-grained word-level encoding layer. We define the word which has the maximum number of characters in $\text{seg}(c_i), c_i \in$ **sent** as coarse-grained word. Similarly, for the p-th coarse-grained word $\text{coarse}_p = \text{fine}_{b,e}$, where coarse_p is the coarse-grained word starts with the b-th and ends with the e-th fine-grained word. The final output hidden state of the coarse-grained word-level encoder is

$$\begin{aligned} \boldsymbol{x}_p^{\text{coarse}} &= \text{avgpool}(\{\boldsymbol{h}_k^{\text{fine}} | b \leqslant k \leqslant e\}) \oplus \mathbf{emb}^{\text{word}}(\text{coarse}_p), \\ \boldsymbol{h}_p^{\text{coarse}} &= \text{coarsewordBiLSTM}(\boldsymbol{x}_p^{\text{coarse}}). \end{aligned} \tag{6}$$

3.5 Fusion layer

We design a char-subword-word fusion layer on top of hierarchical encoding layer to fuse the information of characters, subwords, and words. We extract the output hidden states of the four sub-layers of the hierarchical encoder and use weighed sum to form the char-subword-word hybrid representation, which is denoted as the green circles in Figure 2:

$$\mathbf{rep}_i^{\text{hier}} = \alpha_c \cdot \boldsymbol{h}_i^c + \alpha_{\text{sub}} \cdot \boldsymbol{h}^{\text{sub}}(c_i) + \alpha_{\text{fine}} \cdot \boldsymbol{h}^{\text{fine}}(c_i) + \alpha_{\text{coarse}} \cdot \boldsymbol{h}^{\text{coarse}}(c_i), \tag{7}$$

where α_c, α_{sub}, α_{fine}, α_{coarse} are trainable softmax weights. $\boldsymbol{h}^{\text{sub}}(c_i)$, $\boldsymbol{h}^{\text{fine}}(c_i)$, $\boldsymbol{h}^{\text{coarse}}(c_i)$ are the output hidden states of the subword, fine-grained word and coarse-grained word which end with the i-th character respectively. If there is no subword/fine-grained word/coarse-grained word ends with c_i, we use a padding vector to represent $\boldsymbol{h}^{\text{sub}}(c_i)/\boldsymbol{h}^{\text{fine}}(c_i)/\boldsymbol{h}^{\text{coarse}}(c_i)$.

Moreover, to address the issue that character, subword and word embeddings are diluted with the stacking of hierarchical layers, we fuse these embeddings with the above hierarchical representation so that the semantic information in embeddings is strengthened. We also integrate lexicon words in the fusion layer since they can provide additional word information. We use the blue circles in Figure 2 to denote the embedding representation.

$$\begin{aligned} \mathbf{rep}_i^{\text{emb}} &= \mathbf{emb}^c(c_i) \oplus \mathbf{emb}^{\text{sub}}(\text{sub}(c_i)) \oplus \mathbf{rep}_i^{\text{lex}} \oplus \mathbf{rep}_i^{\text{seg}}, \\ \mathbf{rep}_i^{\text{lex}} &= \text{avgpool}(\{\mathbf{emb}^{\text{word}}(l) | l \in \text{lex}(c_i)\}), \\ \mathbf{rep}_i^{\text{seg}} &= \text{avgpool}(\{\mathbf{emb}^{\text{word}}(s) | s \in \text{seg}(c_i)\}), \end{aligned} \tag{8}$$

where $\text{lex}(c_i)$ are the matched lexicon words which end with c_i. We feed the concatenation of $\mathbf{rep}_i^{\text{hier}}$ and $\mathbf{rep}_i^{\text{emb}}$ to a BiLSTM to obtain the final representation:

$$\boldsymbol{h}_i^{\text{fus}} = \text{fuseBiLSTM}(\mathbf{rep}_i^{\text{hier}} \oplus \mathbf{rep}_i^{\text{emb}}). \tag{9}$$

Table 1 Data statistics

Dataset	Train (#Sent)	Dev (#Sent)	Test (#Sent)
OntoNotes	15724	4301	4346
MSRA	41728	4636	4365
Weibo NER	1350	270	270

3.6 CRF layer

In order to fully consider the dependencies of adjacent labels and make sequence labeling decisions jointly, we employ a CRF layer for training and decoding. For a predicted label sequence $\boldsymbol{y} = y_1 y_2 \cdots y_n$ of **sent**, we define its prediction score as

$$\texttt{score}(\mathbf{sent}, \boldsymbol{y}) = \sum_{i=1}^{n} (T_{y_{i-1}, y_i} + S_{i, y_i}), \tag{10}$$

where T_{y_{i-1}, y_i} is the transition score jumping from label y_{i-1} to y_i. S_{i, y_i} denotes the score of the i-th character labeled as y_i:

$$\boldsymbol{S}_i = \boldsymbol{W}_s \boldsymbol{h}_i^{\text{fus}} + \boldsymbol{b}_s. \tag{11}$$

Given a sentence, the probability of its label sequence $\boldsymbol{y}$ is

$$p(\boldsymbol{y}|\mathbf{sent}) = \frac{e^{\texttt{score}(\mathbf{sent}, \boldsymbol{y})}}{\sum_{\boldsymbol{y}' \in \boldsymbol{Y}_{\mathbf{sent}}} e^{\texttt{score}(\mathbf{sent}, \boldsymbol{y}')}}. \tag{12}$$

Here, $\boldsymbol{Y}_{\mathbf{sent}}$ represents all the possible label sequences.

During training, we use the log-likelihood loss to maximize the probability of the gold label sequence $\boldsymbol{y}^*$:

$$\mathcal{L} = -\log(p(\boldsymbol{y}^*|\mathbf{sent})). \tag{13}$$

At test time, we adopt the dynamic programming Viterbi algorithm to find the highest-scoring label sequence $\hat{\boldsymbol{y}}$:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}' \in \boldsymbol{Y}_{\mathbf{sent}}} \texttt{score}(\mathbf{sent}, \boldsymbol{y}'). \tag{14}$$

4 Experiments

In this section, we conduct experiments to show the effectiveness of our proposed HiLSTM CNER model.

4.1 Experimental settings

Data. We evaluate our model on OntoNotes [29], MSRA [30], and Weibo NER [24]. We adopt the same data split as Zhang and Yang [15] on OntoNotes and Weibo NER. As for the MSRA, which does not have development data, we randomly sample 10% of the training data as the development data. Table 1 shows the statistics of the data. OntoNotes and MSRA are newswire data, Weibo NER is social media data collected from Sina Weibo.

Embeddings. We utilize the same character embeddings and word embeddings as Zhang and Yang [15], which are pretrained with word2vec model on Chinese Gigaword. The subword embeddings are pretrained by Heinzerling and Strube [20] with word2vec. We use the word embedding dictionary as word lexicon in our model, containing 704368 words. All the embeddings are fine-tuned during training.

Model settings. We set the embedding size to 50 and the hidden size of LSTM to 200. The dropout is set to 0.1 for Weibo and 0.5 for the other two datasets. We use stochastic gradient descent (SGD) as the optimizer with a learning rate of 0.015 initially and decays at a rate of 0.05. Early stopping is triggered when the peak performance on the development data does not increase in 30 consecutive iterations.

Evaluation metrics. We use the standard precision ($\frac{\#\texttt{Entity}_{\texttt{correct}}}{\#\texttt{Entity}_{\texttt{pred}}}$), recall ($\frac{\#\texttt{Entity}_{\texttt{correct}}}{\#\texttt{Entity}_{\texttt{gold}}}$) and $F1$ ($\frac{2PR}{P+R}$) score to measure the NER performance. $\#\texttt{Entity}_{\texttt{correct}}$ represents the number of entities that are

Table 2 Development results on OntoNotes. For the "with Lex words only" results of Lattice LSTM and WC-LSTM, we re-run the codes released by Zhang and Yang [15] and Liu et al. [17]. We also modify their models by encoding char-subword-word and give corresponding results in "with char-subword-word hybrid"[a)]

Models	P (%)	R (%)	$F1$ (%)
Char-based LSTM	67.12	58.42	62.47
Lattice LSTM			
with Lex words only	74.64	68.83	71.62
with char-subword-word hybrid	76.20	72.26	**74.17**
WC-LSTM			
with Lex words only	73.08	68.62	70.31
with char-subword-word hybrid	79.73	69.31	**74.15**
HiLSTM			
with Lex words only	74.12	67.41	70.60
with char-subword-word hybrid	76.84	73.06	**74.90**

a) We use bold font to mark the best $F1$ score in each major row.

predicted correctly by the model. $\#\texttt{Entity}_{\texttt{pred}}$ and $\#\texttt{Entity}_{\texttt{gold}}$ are the number of predicted entities and the number of gold entities respectively. We adopt Dan Bikel's randomized parsing evaluation comparator for significance test [31].

4.2 Benchmark methods

In order to verify the effectiveness of our proposed HiLSTM model, we adopt the following benchmark methods for comparison.

Char-based LSTM uses an LSTM-CRF model on the character sequence to predict the label of each character in the sentence.

Lattice LSTM [15] is proposed to encode lexicon words using a lattice structure. It extends the original char-based LSTM by adding extra LSTM memory cells. For each memory cell, it takes the embedding of a matched lexicon word and the hidden state of the word start character as inputs. Shortcut paths are introduced to link the memory cell between the start and the end characters of a lexicon word. We can also integrate segmentor words and subwords into the Lattice LSTM by adding extra memory cells for segmentor words and subwords.

WC-LSTM [17] integrates lexicon words into CNER model by assigning each character with a corresponding lexicon word representation. Specifically, for character c_i, an average pooling operation is performed on the matched lexicon words end with c_i to obtain the lexicon word representation of c_i. The concatenation of character embedding and its corresponding lexicon word representation is then fed into the BiLSTM-CRF. We can obtain the representation of segmentor words and subwords for each character in a similar way and concatenate their representations with character embeddings and lexicon word representation to encode the hybrid information.

4.3 Development results

To learn the influence of adding extra segmentor words and subwords information to lexicon-enhanced models and the performance of different model architectures, we compare the development results on OntoNotes. In Table 2, "with Lex words only" means the model is only integrated with lexicon words, "with char-subword-word hybrid" is the model enhanced with all the three sources, i.e., lexicon words, segmentor words and subwords.

As shown in Table 2, incorporating lexicon words into Lattice LSTM, WC-LSTM, and HiLSTM brings significant improvement over the Char-based model. Further encoding the char-subword-word hybrid into the three models, instead of only integrating lexicon words, outperforms the results of "with Lex words only" by large margin. This demonstrates that segmentor words and subwords contain different morphology knowledge from lexicon words thus can provide additional information for CNER model.

Table 3 Final results on OntoNotes, MSRA, and Weibo NER test data[a)]

Model	OntoNotes (%)			MSRA (%)			Weibo NER (%)		
	P	R	$F1$	P	R	$F1$	$F1$ (NE)	$F1$ (NM)	$F1$ (All)
Maximum entropy [32]	–	–	–	92.20	90.18	91.18	–	–	–
Global linear [33]	–	–	–	91.86	88.75	90.28	–	–	–
Radical-level LSTM [14]	–	–	–	91.28	90.62	90.95	–	–	–
Unified model [22]	–	–	–	–	–	–	54.50	62.17	58.23
MTL [26]	–	–	–	–	–	–	55.28	62.97	58.99
Adversarial MTL [27]	–	–	–	91.73	89.58	90.64	54.34	57.35	58.70
GNN† [16]	76.13	73.68	**74.89**	94.19	92.73	93.46	55.34	64.98	60.21
Collaborative GNN⋆ [18]	75.06	**74.52**	74.79	94.01	**92.93**	93.47	**56.45**	**68.43**	**63.09**
Collaborative GNN† [18]	74.42	72.60	73.50	92.52	90.29	91.39	50.57	64.50	56.33
Lattice LSTM† [15]	**76.35**	71.56	73.88	93.57	92.79	93.18	53.04	62.25	58.79
WC-LSTM† [17]	76.09	72.85	74.43	**94.58**	92.91	**93.74**	53.19	67.41	59.84
Char-based LSTM	68.79	60.35	64.30	90.74	86.96	88.81	46.11	55.29	52.77
Our HiLSTM	**77.77**	**76.32**	**77.04**	**94.83**	**93.61**	**94.22**	**60.94**	**68.89**	**63.79**

a) We use bold font to mark the best result in each major row.

We also observe that the HiLSTM model enhanced with char-subword-word hybrid representation achieves the best $F1$-score of 74.90%, which is consistently higher than the Lattice LSTM and WC-LSTM by 0.73% and 0.75% in $F1$-score ($p < 0.001$). This shows the superiority of encoding the char-subword-word hybrid information in a hierarchical way. The probable reason is that the characters, subwords and words can be seen as a hierarchical structure naturally and the lower levels usually provide rich semantic knowledge for the upper levels. For example, sometimes we can even obtain the meaning of a word by simply composing its subwords meanings. Our proposed HiLSTM model makes full use of the implicit knowledge contained in the hierarchical structure through the propagations from the lower character level to the upper subword and word levels, whereas the Lattice LSTM and WC-LSTM are non-hierarchical models and unable to capture the relations between character-level, subword-level and word-level.

From Table 2 we can conclude that (1) incorporating segmentor words and subwords into lexicon-enhanced models helps to further improve the CNER performance consistently, (2) encoding char-subword-word hybrid in a hierarchical way is superior to other two non-hierarchical methods, i.e., Lattice LSTM and WC-LSTM.

4.4 Final results

Table 3 shows the final results on OntoNotes, MSRA, and Weibo NER test. † denotes the models using the lexicon over automatically segmented Chinese GigaWord (contains 704.4 k words) [15], which is the same with the lexicon used in our HiLSTM. ⋆ means the corresponding model uses the lexicon obtained from 7 corpora of different sizes and domains, including Weibo, People's Daily News, Baidu Encyclopedia, etc (contains 1.3 million words) [34].

OntoNotes. As shown in Table 3, our proposed HiLSTM model encoded with char-subword-word hybrid representation gives the performance of 77.04 % in $F1$-score, outperforming the Char-based LSTM model by a large margin. Compared with the previous best result achieved by Gui et al. [16], who utilize lexicon information with GNN, our model gains a 2.15 % improvement in $F1$-score, reaching state-of-the-art performance.

MSRA. Previous studies leverage hand-crafted features [32, 33], radical features [14], and lexicon words [15–18] for CNER or learn word segmentation and CNER jointly based on an adversarial multi-task learning (MTL) framework [27] and achieve good performance. Compared with previous studies and the Char-based LSTM, our HiLSTM model gives the best results of 94.22 % in $F1$-score.

Weibo NER. Peng and Dredze [26] consider CNER and word segmentation as MTL to make full use of task-shared information. He and Sun [22] propose a unified model to improve the CNER performance with large-scale semi-supervised and cross-domain data. Our HiLSTM model outperforms all the previous

Table 4 Ablation study. "Lex words" and "Seg words" represent lexicon words and segmentor words[a)]

Model	OntoNotes $F1$ (%)	MSRA $F1$ (%)	Weibo $F1$ (%)
Complete HiLSTM	**74.90**	**94.87**	**69.71**
w/o Seg words	72.91	94.81	67.28
w/o Subwords	73.55	94.38	66.67
w/o Lex words	73.78	94.31	67.72
w/o Seg words & Subwords	70.60	94.39	65.86
w/o Seg words & Lex words	71.17	92.09	67.81
w/o Subwords & Lex words	70.46	93.93	65.47
Char-based LSTM	62.47	90.59	59.43

a) We use bold font to mark the best result.

works on named entities, nominal entities and both. Please note that our HiLSTM model even outperforms the "Collaborative GNN⋆" model proposed by Sui et al. [18], which uses additional domain-specific Weibo lexicon[1)]. Compared with their model, our HiLSTM does not rely on domain-specific lexicon and achieves better results.

Overall, our proposed HiLSTM model encoded with char-subword-word outperforms both the char-based LSTM and lexicon-enhanced methods, achieving the best performance on all the three datasets.

5 Analysis

In order to better understand the improvements introduced by the HiLSTM, we conduct detailed analyses from different perspectives.

5.1 Ablation study

We conduct ablation study to analyze the contribution of lexicon words, segmentor words, and subwords in our proposed HiLSTM model. Table 4 shows the results on OntoNotes, MSRA, and Weibo development datasets. We observe that the performance of the HiLSTM model is degraded when removing any of the three sources. This demonstrates that the information in lexicon words, segmentor words, and subwords are all beneficial for CNER. We also find that integrating any one of the sources can improve the HiLSTM model by a large margin, and fully exploiting all the three sources leads to the best result. Overall, we can conclude that lexicon words, segmentor words and subwords can provide complementary contributions to the model.

5.2 Entity coverage

Figure 3 shows the entity coverage ratio in segmentor words, subwords, lexicon words, and the hybrid of above three sources. From Figure 3, we can draw the following findings. First, it can be seen that the entity coverage ratio in segmentor words is higher than that in lexicon and subwords, because the entities such as location and organization names often have the same boundaries with coarse-grained words as discussed in Section 1, whereas they rarely exist in lexicon or subwords. Therefore, the segmentor words can provide CNER model with abundant boundary information. Second, as dipicted by the red bars (denoted as "Hybrid") in Figure 3, the hybrid of segmentor words, subwords and lexicon words covers the most number of entities. This explains why the three sources can make complementary contributions to the HiLSTM model. Third, when turning to the final results in Table 3, we find the improvements of the final HiLSTM model over the char-based LSTM model (by 13.26, 5.41, and 11.02 $F1$-score on OntoNotes, MSRA, and Weibo NER, respectively) are correlated to the entity coverage, higher entity coverage usually contributes to greater progress in performance. On OntoNotes, the complete HiLSTM model encoding

1) We also replace their lexicon (obtained from 7 corpora of different domains) with the same lexicon as ours (obtained from Chinese GigaWord) and re-run the Collaborative GNN model using their released code according to their model settings. We observe that the performance of the "Collaborative GNN †" is degraded without domain-specific lexicon, which is shown in Table 3.

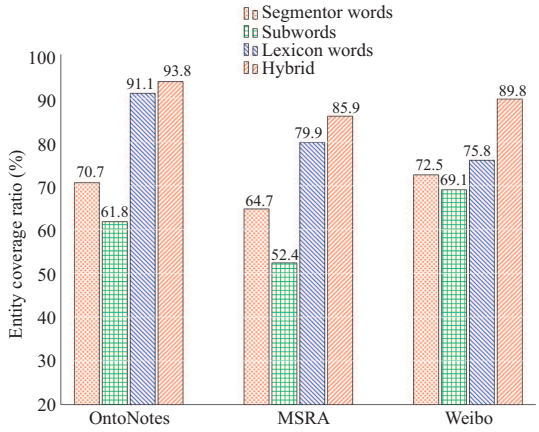

Figure 3 (Color online) The entity coverage ratio in segmentor words, subwords, lexicon words and the hybrid of the three sources on OntoNotes, MSRA, and Weibo NER dataset.

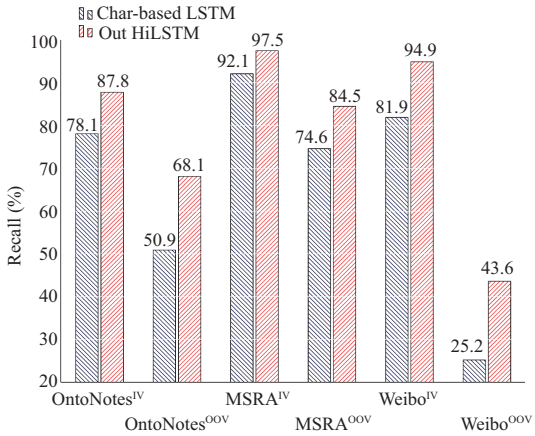

Figure 4 (Color online) The recall of in-vocabulary (IV) and out-of-vocabulary (OOV) entities on OntoNotes, MSRA, and Weibo NER dataset.

the hybrid of segmentor words, subwords and lexicon words brings the maximum improvement compared with other two datasets, since the entity coverage in this dataset is the highest and thus can offer rich boundary information to CNER model. The improvement on MSRA is limited due to the relatively low entity coverage.

5.3 In/out-of-vocabulary (IV/OOV) analysis

Figure 4 investigates how the HiLSTM model improves performance from the perspective of in-vocabulary (IV) and out-of-vocabulary (OOV) named entities. We divide the entities in the test set of each dataset into two categories: entities exist in the training data (IV) and entities out of the training data (OOV). As shown in Figure 4, the recall of both IV and OOV entities increases significantly after integrating the char-subword-word hibrid representation with the HiLSTM structure. The absolute improvement is especially large for OOV entities (17.2%, 9.9%, and 18.4% on OntoNotes, MSRA, and Weibo NER, respectively). After conducting detailed statistical analysis, we find that the coverage ratio of OOV entities in the hybrid of segmentor words, subwords and lexicon words is more than three quarters (88.65%, 81.80%, and 76.07% on OntoNotes, MSRA, and Weibo NER datasets). This means the segmentor words, subwords and lexicon words covers a lot of entities that are missing in the training data and thus can provide additional entity information and alleviate OOV problems by integrating the hybrid representation into the HiLSTM model.

5.4 Case study

Table 5 shows a case study comparing the results of the HiLSTM model without segmentor words, the model without subwords and the complete HiLSTM model.

In the first case, there is an entity "中国进出口银行 (the export-import bank of China)" with nested "中国 (China)" and "进出口银行 (the export-import bank)". The word "中国 (China)" is included in all the three sources, i.e., lexicon words, segmentor words and subwords, whereas the organization entity "中国进出口银行 (the export-import bank of China)" is only covered in the context-aware segmentor words. Without segmentor words, the model ignores the sentence context and incorrectly predicts "中国 (China)" as an entity. In contrast, after integrating segmentor words, the complete HiLSTM is able to obtain context information and successfully detects the organization name thanks to the coarse-grained segmentor word "中国进出口银行 (the export-import bank of China)".

In the second case, without subwords, the model recognizes "港澳 (Hong Kong and Macao)" as a location entity, affected by the word "港澳 (Hong Kong and Macao)" exists in both lexicon words and

Table 5 Case study[a)]

Id		Cases
1	Sentence	... 对中国进出口银行有较深的了解 (have a relatively deep understanding of the Export-Import Bank of China)...
	Lex words	... 中国 (China), 进出 (in and out), 进出口 (imports and exports), 出口 (exit), 银行 (bank), 了解 (understand)...
	Seg words	... 对 (to), 中国 (China), 中国进出口银行 (the export-import bank of China), 进出口 (imports and exports), 银行 (bank), 有 (have), 较深 (relatively deep), 较 (relatively), 深 (deep), 的 (de), 了解 (understand)...
	Subwords	... 对 (to), 中国 (China), 进 (in), 出 (out), 口 (entrance), 银行 (bank), 有 (have), 较 (relatively), 深 (deep), 的 (de), 了解 (understand)...
	Gold result	... 对 (to) 中国进出口银行 (the export-import bank of China) [ORG] 有了较深的了解 (have a relatively deep understanding)...
	w/o Seg words predicted result	... 对 (to) 中国 (China) [GPE] 进出口银行 (the export-import bank) 有了较深的了解 (have a relatively deep understanding)...
	with Seg words predicted result	... 对 (to) **中国进出口银行 (the export-import bank of China) [ORG]** 有了较深的了解 (have a relatively deep understanding)...
2	Sentence	... 充分利用毗邻港澳的优势 (make full use of the advantages of adjoining Hong Kong and Macao)...
	Lex words	... 充分 (full), 利用 (use), 毗邻 (adjoin), 港澳 (Hong Kong and Macao), 优势 (advantage)...
	Seg words	... 充分 (full), 利用 (use), 毗邻 (adjoin), 港澳 (Hong Kong and Macao), 的 (de), 优势 (advantage)...
	Subwords	... 充分 (full), 利用 (use), 毗 (connect), 邻 (neighbour), 港 (Hong Kong), 澳 (Macao), 的 (de), 优势 (advantage)...
	Gold result	... 充分利用毗邻 (make full use of the adjoining) 港 (Hong Kong) [GPE] 澳 (Macao) [GPE] 的优势 (advantage)...
	w/o Subwords predicted result	... 充分利用毗邻 (make full use of the adjoining) 港澳 (Hong Kong and Macao) [LOC] 的优势 (advantage)...
	with Subwords predicted result	... 充分利用毗邻 (make full use of the adjoining) **港 (Hong Kong) [GPE] 澳 (Macao) [GPE]** 的优势 (advantage)...

a) We use underline and bold fonts to denote the wrong and correct predicted labels.

segmentor words. However, "港澳 (Hong Kong and Macao)" is a compound word composed of two geographical political entities "港 (Hong Kong)" and "澳 (Macao)" and these two entities are not covered in lexicon words and segmentor words. With the help of subwords "港 (Hong Kong)" and "澳 (Macao)", the complete HiLSTM model is able to recognize the components of the compound word "港澳 (Hong Kong and Macao)" and finally predicts the correct labels.

6 Conclusion

We propose an effective hierarchical LSTM architecture to integrate char-subword-word hibrid representation into CNER model and further improve the performance of CNER. Specifically, we first construct a char-subword-word tree structure to represent each sentence for NER by making use of lexicon words, segmentor words and subwords. Then, we propose a HiLSTM model to capture the linguistic information of different levels in the char-subword-word tree-structure representation. Finally, we conduct experiments on three widely-used datasets to verify the effectiveness of our proposed HiLSTM model and give detailed analysis. Experiments on three datasets show that the lexicon words, segmentor words and subwords can provide complementary contributions to the CNER model. Our proposed HiLSTM model encoded with char-subword-word representation achieves the performance of 77.04%, 63.79%, and 94.22% in $F1$-score on OntoNotes, Weibo NER, and MSRA datasets respectively, reaching new state-of-the-art results on the three datasets.

In this paper, we focus on using our HiLSTM model enhanced with char-subword-word tree-structure representations to improve performance for the task of CNER. In fact, the proposed hierarchical char-subword-word encoder can also be adapted to other Chinese NLP tasks, such as sentiment analysis, intent classification, and question answering. In the future, we will try to apply the char-subword-word

tree-structure representation and the hierarchical LSTM architecture on other NLP tasks.

Acknowledgements This work was supported by National Science Fund for Distinguished Young Scholars (Grant No. 61525205), National Natural Science Foundation of China (Grant No. 61876116), and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1 Chen Y B, Xu L H, Liu K, et al. Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2015. 167–176
2 Miwa M, Bansal M. End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2016. 1105–1116
3 Diefenbach D, Lopez V, Singh K, et al. Core techniques of question answering systems over knowledge bases: a survey. Knowl Inform Syst, 2018. https://hal.archives-ouvertes.fr/hal-01637143/document
4 Yang J F, Guan Y, He B, et al. Corpus construction for named entities and entity relations on Chinese electronic medical records. J Softw, 2016, 27: 2725–2746
5 Song L F, Zhang Y, Gildea D, et al. Leveraging dependency forest for neural medical relation extraction. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2019. 208–218
6 Tian Y H, Ma W C, Xia F, et al. ChiMed: a Chinese medical corpus for question answering. In: Proceedings of the 18th BioNLP Workshop and Shared Task, 2019. 250–260
7 Saito K, Nagata M. Multi-language named-entity recognition system based on HMM. In: Proceedings of Annual Meeting of the Association for Computational Linguistics Workshop on Multilingual and Mixed-language Named Entity Recognition, 2003
8 Yu H K, Zhang H P, Liu Q, et al. Chinese named entity identification using cascaded hidden Markov model. J Commun, 2006, 27: 87–94
9 Solorio T, López A. Learning named entity classifiers using support vector machines. In: Proceedings of Conference on Computational Linguistics and Natural Language Processing (CICLing), 2004. 158–167
10 Mccallum A, Li W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), 2003. 188–191
11 Chiu J P C, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. Trans Assoc Comput Linguist, 2016, 4: 357–370
12 Lample G, Ballesteros M, Subramanian S, et al. Neural architectures for named entity recognition. In: Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), 2016. 260–270
13 Liu L Y, Shang J B, Xu F, et al. Empower sequence labeling with task-aware neural language model. In: Proceedings of Association for the Advance of Artificial Intelligence (AAAI), 2018. 5253–5260
14 Dong C H, Zhang J J, Zong C Q, et al. Character-based LSTM-CRF with radical-level features for Chinese named entity recognition. In: Proceedings of International Conference on Computer Processing of Oriental Languages (ICCPOL), 2016. 239–250
15 Zhang Y, Yang J. Chinese NER using lattice LSTM. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2018. 1554–1564
16 Gui T, Zou Y C, Zhang Q, et al. A lexicon-based graph neural network for Chinese NER. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2019. 1040–1050
17 Liu W, Xu T G, Xu Q H, et al. An encoding strategy based word-character LSTM for Chinese NER. In: Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL), 2019. 2379–2389
18 Sui D B, Chen Y B, Liu K, et al. Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2019. 3828–3838
19 Gong C, Li Z H, Zhang M, et al. Multi-grained Chinese word segmentation. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2017. 703–714
20 Heinzerling B, Strube M. BPEmb: tokenization-free pre-trained subword embeddings in 275 languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC), 2018. 2989–2993
21 Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch. J Mach Learn Res, 2011, 12: 2493–2537
22 He H F, Sun X. A unified model for cross-domain and semi-supervised named entity recognition in Chinese social media. In: Proceedings of Association for the Advance of Artificial Intelligence (AAAI), 2017. 3216–3222
23 Zhao H, Kit C Y. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and namedentity recognition. In: Proceedings of SIGHAN Workshop on Chinese Language Processing, 2008
24 Peng N Y, Dredze M. Named entity recognition for Chinese social media with jointly trained embeddings. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2015. 548–554
25 He H F, Sun X. F-score driven max margin neural network for named entity recognition in Chinese social media. In: Proceedings of European Chapter of the Association for Computational Linguistics (EACL), 2017. 713–718
26 Peng N Y, Dredze M. Improving named entity recognition for Chinese social media with word segmentation representation learning. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2016.

149–155
27 Cao P F, Chen Y B, Liu K, et al. Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2018. 182–192
28 Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2016. 1715–1725
29 Weischedel R, Palmer M, Marcus M, et al. OntoNotes Release 4.0. Philadelphia: Linguistic Data Consortium, 2011
30 Levow G A. The third international Chinese language processing backoff: word segmentation and named entity recognition. In: Proceedings of SIGHAN Workshop on Chinese Language Processing, 2006. 108–117
31 Noreen E. Computer-intensive methods for testing hypotheses. Biometrics, 1990, 46: 540–541
32 Zhang S X, Qin Y, Wen J, et al. Word segmentation and named entity recognition for SIGHAN bakeoff3. In: Proceedings of SIGHAN Workshop on Chinese Language Processing, 2006. 158–161
33 Zhou J S, Qu W G, Zhang F. Chinese named entity recognition via joint identification and categorization. Chinese J Electron, 2013, 22: 225–230
34 Li S, Zhao Z, Hu R F, et al. Analogical reasoning on Chinese morphological and semantic relations. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2018. 138–143