Dynamic Modeling Cross- and Self-Lattice Attention Network for Chinese NER

Shan Zhao[1], Minghao Hu[2*], Zhiping Cai[1†] Haiwen Chen[1], Fang Liu[3]

[1]College of Computer, National University of Defense Technology, Changsha, China
[2]Information Research Center of Military Science, PLA Academy of Military Science, Beijing, China
[3]School of Design, Hunan University, Changsha, Hunan
{zhaoshan18, zpcai, chenhaiwen13}@nudt.edu.cn, {huminghao16}@gmail.com,{fangl}@hnu.edu.cn

Abstract

Word-character lattice models have been proved to be effective for Chinese named entity recognition (NER), in which word boundary information is fused into character sequences for enhancing character representations. However, prior approaches have only used simple methods such as feature concatenation or position encoding to integrate word-character lattice information, but fail to capture fine-grained correlations in word-character spaces. In this paper, we propose DCSAN, a Dynamic Cross- and Self-lattice Attention Network that aims to model dense interactions over word-character lattice structure for Chinese NER. By carefully combining cross-lattice and self-lattice attention modules with gated word-character semantic fusion unit, the network can explicitly capture fine-grained correlations across different spaces (e.g., word-to-character and character-to-character), thus significantly improving model performance. Experiments on four Chinese NER datasets show that DCSAN obtains state-of-the-art results as well as efficiency compared to several competitive approaches.

Introduction

Named Entity Recognition (NER), which aims to automatically detect named entities from giving text and identify their categories (Zhao et al. 2020), is one of the most important tasks in information extraction. Due to the additional word segmentation process of Chinese (Zhao et al. 2019), Chinese NER is more difficult compared to English NER.

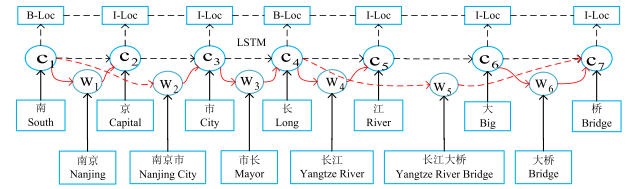
Traditionally, the task of Chinese NER is decoupled into a pipeline of two separated subtasks, namely word segmentation and word sequence labeling (Yang et al. 2016). The major disadvantage of this method is error propagation: word segmentation errors negatively impact the identification of named entities (Peng and Dredze 2015; He and Sun 2016). Character-based models, on the other hand, can naturally avoid word segmentation errors, thus outperforming word-based methods. Moreover, to explicitly inform each character about its related word information, previous works (Zhang and Yang 2018; Liu et al. 2019; Yan et al. 2019) have proposed to integrate word information into character sequences via word-character lattice structure, as shown in Figure 1(a).

Prior approaches have attempted to utilize different network architectures to integrate word-character lattice information, such as RNN-based (Zhang and Yang 2018), CNN-based (Gui et al. 2019a), Graph-based (Gui et al. 2019b), and Transformer-based models (Li et al. 2020). However, these lattice methods have not considered capturing fine-grained correlations between each character and its corresponding matched word. For example, WC-LSTM (Liu et al. 2019) adopts simple feature concatenation to fuse word-character lattice information, while FLAT (Li et al. 2020) utilizes position encoding to propagate information in a flat-lattice structure. We argue that, for the Chinese NER task, effectively modeling dense interactions between each character and each matched word is crucial. Taking the sentence in Figure 1(c) as an example, it is beneficial that the character "长 (Long)" or 江 (River) is aware of "桥 (Bridge)" being matched with the "长江大桥 (Yangtze River Bridge)" word.

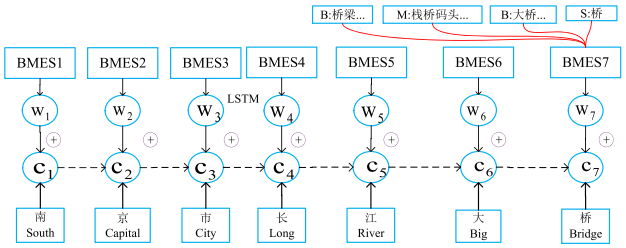
To address the above issue, we propose a Dynamic Cross- and Self-lattice Attention Network (DCSAN) for Chinese NER. The key insight comes from multimodal learning in computer vision (Gao et al. 2019; Yu et al. 2019), where the character and word sequences are viewed as two different modalities. To model dense interactions over word-character lattice structure, we first design a cross-lattice attention module that aims to capture fine-grained correlations between two input feature spaces. Then, we further construct a dynamic self-lattice attention module that is capable of dynamically fusing word-character features and building direct connections between two arbitrary characters despite of their distances. Given the word-character embeddings and the aligned lattice structure, DCSAN first utilizes the cross-lattice attention module to generate word-aware character features, and then adopts the dynamic self-lattice attention module to combine character and word features, eventually obtaining self-aware character features. In this way, our network can fully capture dense interactions over word-character lattice structure, thus providing rich representations for Chinese NER prediction.

Finally, we conducted extensive experiments on four NER datasets to evaluate the proposed model. Experimental results show that DCSAN can achieve state-of-the-art performance and efficiency compared to a variety of compet-

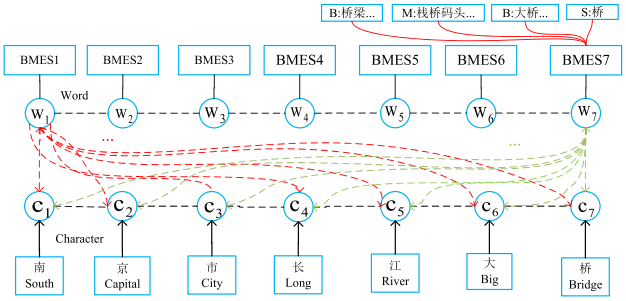
*Corresponding Author.
†Corresponding Author.



(a) Original Lattice LSTM



(b) Soft-lexicon feature strategy



(c) Dense interactions of lattice inputs

Figure 1: (a) An example of the original lattice LSTM model (Zhang and Yang 2018); (b) Soft-lexicon strategy used (Ma et al. 2020); (c) Dense interactions of lattice inputs in our model.

itive approaches. In particular, we obtain 96.67%, 96.21%, 71.27%, and 82.62% F1 on Resume, MSRA, Weibo, and E-commerce datasets respectively. We will release the source code to facilitate future research in this field. [1]

Related Work

Chinese NER with Lattice Structure

Since word sequence information is potentially useful for character-based sequence learning, neural networks with word-character lattice structures have outperformed both word-based and character-based approaches by a large margin on the Chinese NER task. Specifically, Zhang and Yang (Zhang and Yang 2018) first proposed a Lattice LSTM model to explicitly leverage word boundary information, in which matched lexical words are encoded into character sequences with a directed acyclic graph (DAG) structure. Yet, this DAG structure fails to choose the right path sometimes, which may cause the lattice model to degenerate into a partial word-based model. Later, Liu et al. (Liu et al. 2019) explored four different words encoding strategies to alleviate this issue. Gui et al. (Gui et al. 2019a) proposed a CNN-based NER model (LR-CNN) that encodes matched words at different window sizes. Moreover, Gui et al. (Gui et al. 2019b) and Sui et al. (Sui et al. 2019) converted lattice into graph and use graph neural networks (GNNs) for encoding. However, as NER is very sensitive to sentence structure, these methods still need to use LSTMs as backbone encoder, which makes the models complicated. Recently, Yan et al. (Yan et al. 2019) proposed an adapted Transformer encoder for Chinese NER. Ma et al. (Ma et al. 2020) constructed the soft-lexicon feature to encoding the matched words, obtained from the lexicon, into the representations of characters. Li et al. (Li et al. 2020) leveraged a flat lattice structure so that Transformer can capture word information via position encoding. The main difference between our network and the above methods is that our network consists of cascaded attention modules to model dense interactions across different feature spaces (e.g., word-to-character and character-to-character).

Multimodal Learning

Multimodal learning is widely explored in computer vision and natural language processing. A typical task is visual question answering (VQA) (Antol et al. 2015), which requires the model to perform fine-grained semantic understanding of both the image and the question. For example, Nguyen and Okatani (2018) proposed a dense symmetric co-attention architecture to form a hierarchy for multi-step interactions between an image-question pair. Yu et al. (2019) introduced a VQA model that consists of multiple modular co-attention layers cascaded in depth. Gao et al. (2019) proposed to dynamically fuse multi-modal features with intra- and inter-modality information flow. Inspired by these advancements in this field, we aim to model dense interactions over word-character lattice structure using cascaded attention units and gating mechanism.

The Proposed Model

In this section, we introduce the proposed Dynamic Cross- and Self-lattice Attention Network (DCSAN) in details, as illustrated in Figure 2. We first construct the word-character lattice structure by applying a soft-lexicon feature strategy, and then obtain fixed-dimensional representations of both character and word sequences. Next, we utilize a cross-lattice attention module and a dynamic self-lattice attention module to explicitly model dense interactions across different feature spaces. Finally, we apply a conditional random field (CRF) (Lafferty, McCallum, and Pereira 2001) layer to perform the decoding for Chinese NER.

Character-Word Feature Representations

Since character sequences and matched words are viewed as two different modalities, therefore they are represented as two sets of distributed representations. Below we give detailed explanations on the construction of these representations.

[1] https://github.com/zs50910/DCSAN-for-Chinese-NER

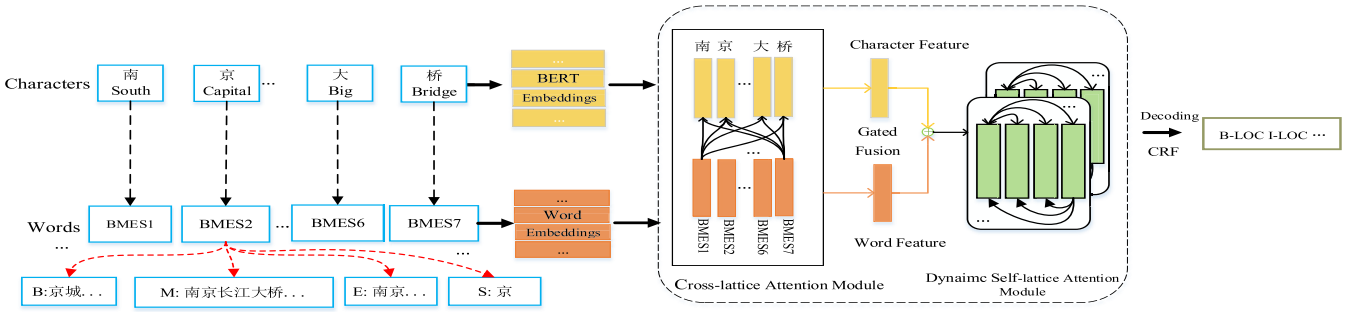


Figure 2: Overall flowchart of DCSAN. Characters and aligned words are first represented as distributed representations. The cross-lattice and dynaimc self-lattice attention modules are designed to explicitly capture dense interactions over word-character lattice structure. Finally, a conditional random field (CRF) (Lafferty, McCallum, and Pereira 2001) layer is used to perform decoding for the Chinese NER task.

Character Representations Character embeddings are used to map discrete characters into continuous input vectors. Given a Chinese input sentence $s = [c_1, c_2..., c_n]$, where c_i represents the i-th character, we map each character into a real-valued embedding to express its semantic and syntactic meaning. Each character c_i is represented as:

$$x_i = \mathrm{B}^c(c_i), x_i \in \mathbb{R}^d \tag{1}$$

where B^c denotes BERT embeddings (Cui et al. 2019). The character feature representations can be obtained as:

$$X = [x_1, x_2, x_3, ..., x_n] \in \mathbb{R}^{n*d} \tag{2}$$

Word Representations To unify the word-character representation space, we use $c_{i,j}$ to denote a word in s, which begins from the i-th character to the j-th character. Taking the sentence in Figure 1(a) for example, $c_{1,3}$ refers to the word "南京市 (Nanjing City)". In the original lattice model, the i-th character is aligned with a set of matched words $w_i = [c_{k,i}, ..., c_{j,i}]$, where $k, j < i$. For instance, the set of matched words for the character "桥 (Bridge)" is $w_7 = [c_{4,7}, c_{6,7}]$, which refers to "长江大桥 (Yangtze River Bridge)" and "大桥 (Big Bridge)" respectively. However, as the number of matched words for each character is dynamically changed (the character "大 (Big)" has no matching word), such lattice structure is deprived of batch training, which makes the model inefficient and difficult to deploy. To address this issue, we use the soft-lexicon feature strategy (Ma et al. 2020), as shown in Figure 1(b). This strategy selects a fixed-dimensional vector which is composed of four word sets marked by the four segmentation labels "BMES" , as the aligned word for each character c_i. Specifically, the word set $B(c_i)$ consists of all lexicon matched words on s that begin with c_i. Similarly, $M(c_i)$ consists of all lexicon matched words in the middle of which c_i occurs, $E(c_i)$ consists of all lexicon matched words that end with c_i, and $S(c_i)$ is the single-character word comprised of c_i. When a word set is empty, we will set a special word "none" to it to indicate this situation. Next, the aligned word w_i for each corresponding character c_i is represented as:

$$y_i = [v(B(c_i)); v(M(c_i)); v(E(c_i)); v(S(c_i))], y_i \in \mathbb{R}^{4d} \tag{3}$$

where v denotes the function that maps a single word set to a dense vector. The function works as:

$$v(p) = \frac{1}{Z} \sum_{w \in p} (z(w) + b) e^w \tag{4}$$

where $z(w)$ denote the frequency of w_c occurring in the statistic data set; w_c is the character sequence constituting w; e^w represents a pre-trained word embedding lookup table; b denotes the value that there are 10% of training words occurring less than b times within the statistic data set. Z can be computed by:

$$Z = \sum_{w \in (B \sqcup M \sqcup E \sqcup S)} z(w) + b \tag{5}$$

To facilitate calculation, we utilize a linear projections to transform dimension, and finally word feature representations can be obtained as:

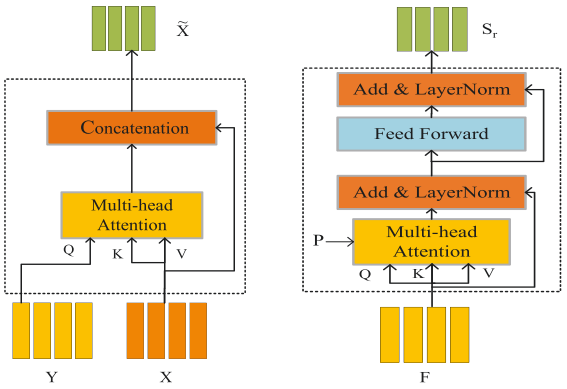
$$Y = \mathrm{Linear}[y_1, y_2, y_3, ..., y_n] \in \mathbb{R}^{n*d} \tag{6}$$

Cross-lattice Attention Module

Cross-lattice attention module (see Figure 3(a)) aims to capture fine-grained correlations between character and word feature representations, which is a variant of the recently-proposed multi-head attention mechanism (Vaswani et al. 2017). Cross-lattice attention is capable of modeling dense interactions between each pair of character and word feature. Fisrt, scaled dot-product is chosen as the similarity scoring function in this module. Given word feature representations Y as queries, and character feature representations X as keys and values, the matrix of outputs is computed using the following equation:

$$\alpha(Y, X, X) = \mathrm{softmax}(\frac{YX^T}{\sqrt{d}})X \tag{7}$$

Then, cross-lattice attention allows the model to jointly attend to information from different representation subspaces at different positions. It maps the matrix of input vectors to updating query, updating key, and updating value matrices by using different linear projections. And z parallel



(a) Cross-lattice Attention (b) Self-lattice Attention

Figure 3: Two proposed attention modules. Cross-lattice attention aims to model dense interactions between each pair of character and word features, while self-lattice attention is used to capture character-to-character self-correlations.

heads are employed to perform attention operation in different parts of channels:

$$\text{head}_i = \alpha(YW_i^Y, XW_i^X, XW_i^{X'}) \quad (8)$$

$$\text{Inter}_{X\to Y} = [\text{head}_1; ...; \text{head}_z]W^t \quad (9)$$

where $W_i^Y \in \mathbb{R}^{d*d/z}, W_i^X \in \mathbb{R}^{d*d/z}, W_i^{X'} \in \mathbb{R}^{d*d/z}$, and $W^t \in \mathbb{R}^{d*d}$ are trainable parameter matrices.

Finally, we concatenate $\text{Inter}_{X\to Y}$ with original character features, which are transformed into the original dimension by a linear projections. The information flow for updating character features $\widetilde{X}$ is obtained as follows:

$$\widetilde{X} = \text{Linear}[X; \text{Inter}_{X\to Y}] \quad (10)$$

Now, cross-lattice attention learns the pairwise relationship between each paired sample $< x_i, y_j >$ within X and Y, and fuses feature representations to generate word-aware character features.

Dynamic Self-lattice Attention Module

After acquiring dense interactions across lattice structure (word-to-character), the word-aware character features $\widetilde{X}$ already contain rich information over lattice feature spaces. Yet, we argue that capturing self-correlations inside character sequences is also important for Chinese NER and should be taken into account to generate contextualized representations. Therefore, we first propose a gated fusion unit to dynamically integrate character and word features. Then, we introduce a self-lattice attention layer with relative position encoding for modeling self-correlations inside character sequences.

Gated Fusion of Character-Word Pairs We design a gated fusion unit to integrate character and word features. For the NER task, this unit trades off how much information the network is taking from either word features or character features. This is achieved by first computing a gating vector $g \in \mathbb{R}^n$, and then using it to calculate the weighted-sum result from $\widetilde{X}$ and Y. The fused representation of character-word pairs is obtained as follows:

$$h_c = \tanh(\widetilde{X}W_c + b_c) \quad (11)$$

$$h_w = \tanh(YW_w + b_w) \quad (12)$$

$$g = \sigma([h_c; h_w])W_g) \quad (13)$$

$$F = g\widetilde{X} + (1-g)Y \quad (14)$$

where $W_c \in \mathbb{R}^{d*d}, W_w \in \mathbb{R}^{d*d}, W_g \in \mathbb{R}^{2d}, b_c \in \mathbb{R}^d, b_w \in \mathbb{R}^d$ are trainable parameters, and σ is the sigmoid activation function.

Self-lattice Attention Self-lattice attention with relative position encoding (see Figure 3(b)) is designed to model character-level self-correlations, which takes the fused features F and relative position encoding P as inputs. It learns the pairwise relationship between the paired sample $< f_i, f_j >$ within F, and outputs attended self-aware character features by using weighted summation across all instances. This module is a variant of multi-head attention mechanism, which is calculated as follows:

$$\text{head}_i = \text{softmax}((QW_i^Q)K[i]^T + P[i])(VW_i^V) \quad (15)$$

$$O = [\text{head}_1; ...; \text{head}_z]W^o \quad (16)$$

where Q, K, V are all set as F, $W_i^Q \in \mathbb{R}^{d*d/z}, W_i^K \in \mathbb{R}^{d*d/z}, W^o \in \mathbb{R}^{d*d}$ are trainable parameters, $K[i] \in \mathbb{R}^{n*d/z}$ is the i-th partition of K, and $P[i] \in \mathbb{R}^{n*n}$ contains relative position information of the i-th partition.

To explicitly inform the module with positional information, we utilize the relative position encoding method of which details can be found from (Yan et al. 2019). Suppose that t is the index of target token, j is the index of context token, and R_{t-j} is the bias term for certain distance and direction, then the relative position encoding $P[i]$ can be calculated as:

$$m = (2b * z)/d \quad (17)$$

$$R_{t-j} = [\cdots \sin(\frac{t-j}{10000^m})\cos(\frac{t-j}{10000^m})\cdots]^T \quad (18)$$

$$P[i]_{t,j} = (QW_i^Q)R_{t-j} + uK[i]_j^T + vR_{t-j}^T \quad (19)$$

where $u, v \in \mathbb{R}^{d/z}$ are learnable parameters. b in Eq.(17) is in the range $[0; d/(2z)]$, and z is the number of heads.

In our network, the output O of the multi-head attention will be further processed by residual connection (He et al. 2016) and layer normalization (Ba, Kiros, and Hinton 2016) followed by position-wise feedforward networks, which can be computed as follows:

$$Rc = \text{LayerNorm}(O + F) \quad (20)$$

$$\text{FFN}(Rc) = \max(0; RcW_1 + b_1)W_2 + b_2 \quad (21)$$

where W_1, W_2, b_1, b_2 are learnable parameters. Similarly, residual connection along with layer normalization is further applied on $\text{FFN}(Rc)$ to produce the final output features. Thus, the self-lattice attention output can be denoted as:

$$Sr = \text{LayerNorm}(\text{FFN}(Rc) + Rc) \quad (22)$$

Dataset	Type	Train	Dev	Test
Weibo	Char	73.8K	14.5K	14.8K
E-commerce	Char	119.1K	14.9K	14.7K
Resume	Char	124.1K	139K	15.1K
MSRA	Char	2169.9K	-	172.6K

Table 1: Statistics of four Chinese NER datasets.

To increase model capacity, we stack l layers of self-lattice attention operation to form a cascaded architecture. Finally, the encoding output is denoted as $Sr^l \in \mathbb{R}^{n*d}$, which is sent to the decoding layer for prediction.

Decoding and Training

A standard CRF layer is used to predict NER taggings, which takes Sr^l as inputs, and outputs a sequence of predicted tagging probabilities $A = [a_1, ..., a_n]$. Let $A^{'}$ denotes an arbitrary label distribution sequence (i.e., BIO tagging scheme), the probability of the label sequence A can be calculated using a softmax function:

$$Pr(A|Sr^l) = \frac{\prod_{i=1}^{n} \varphi_n(a_{n-1}, a_n, Sr^l)}{\sum_{a' \in A'} \prod_{i=1}^{n} \varphi_n(a'_{n-1}, a'_n, Sr^l)} \quad (23)$$

where $\varphi_n(a_n, a_{n-1}, L) = \exp(W_n Sr^l + b_n)$ is the scoring function and W_n and b_n are the weight vector and bias. During training, we optimize model parameters by minimizing the following conditional likelihood:

$$\mathcal{L}_{ner} = -\log Pr(A|Sr^l) \quad (24)$$

Experiments

Experimental Setup

To evaluate the performance of our model, we conduct experiments on four datasets, including Weibo NER (Peng and Dredze 2015), MSRA (Levow 2006), Chinese resume dataset (Zhang and Yang 2018), and E-commerce NER (Ding et al. 2019). These datasets involve in social media, financial, news, and e-commerce domains, of which detailed statistics are shown in Table 1.

Implementation Details

Following (Li et al. 2020), We utilize the BERT embedding as our character embeddings. The BERT in the experiment is "BERT-wwm" released by (Cui et al. 2019). We use the word embedding dictionary (Song et al. 2018) that contains over 8000k Chinese character and words as default lexicon in our model. As for hyper-parameter configurations, the sizes of character embeddings is 768 and word embeddings is 200 by default, and the dimensionality of hidden size is 768. For attention settings, the head number of cross-lattice attention and dynamic self-lattice attention are 8 and 4 respectively for all datasets. We set the number of self-lattice attention layers l as 2 by default. To avoid overfitting, we regularize our network using dropout with a rate tuned on the development set. To train the model, we use SGD optimizer with a learning rate of 0.0007 on Resume, MSRA, and E-commerce datasets and 0.001 on the Weibo dataset. The training takes 100 epochs until convergence. We adopt standard Precision (P), Recall (R) and F1 score to evaluate the model.

Models	Resume	MSRA	Weibo	E-commerce
Lattice LSTM[1]	94.46	93.18	58.79	-
LR-CNN[2]	95.11	93.71	59.92	-
LGN[3]	95.37	93.46	59.84	-
TENER[4]	95.00	92.74	58.39	-
FLAT[5]	95.45	94.35	63.42	-
FLAT+BERT[5]	95.86	96.09	68.55	-
Multi-Digraph[6]	-	-	-	75.20
DCSAN (ours)	**96.67**	**96.41**	**71.27**	**82.62**

Table 2: Main results (F1) on Resume, MSRA, Weibo and E-commerce datasets. Zhang et al.(2018)[1], Gui et al.(2019a)[2], Gui et al. (2019b)[3], Yan et al.(2019)[4], Li et al. (2020)[5], Ding et al. (2019)[6].

Overall Results

We compare the proposed model with several competing approaches and show the results in Table 2. On Resume, MSRA and Weibo datasets, it can be seen that our model achieves state-of-the-art performance by obtaining 96.67, 96.41, and 71.27 F1 respectively. Compared with the best result among Lattice LSTM, LR-CNN and LGN, our approach gets absolute F1 improvements of 1.3%, 2.7% and 11.35% on three datasets respectively. When compared to the TENER model, we find stronger performance improvement with respect to Resume (+1.67%), MSRA(+3.67%) and Weibo (+12.88%). Compared to the latest FLAT+BERT model, our approach slightly increases by 0.81% and 0.32% on Resume and MSRA datasets respectively. However, it can be found that our proposed model significantly outperforms FLAT+BERT by 2.72% F1 on Weibo. Since E-commerce dataset is released recently, we can find that only Multi-Digraph model, which utilizes graph neural networks with a multi-digraph structure that captures the information of gazetteers offers, has been evaluated on this dataset. Compared to Multi-Digraph, our proposed model significantly outperforms it by 7.42% F1. The above results indicate the effectiveness of our model and suggest that DCSAN is able to better leverage word-character lattice structure.

Ablation Study

We conduct an ablation study to investigate the effectiveness of our attention modules in Table 3. Firstly, we remove the cross-lattice attention module and only use the dynamic self-lattice attention module for encoding. We find that the F1 score obviously decreases by 1.66, showing the beneficial effect of modeling dense interactions among word-character feature spaces. Secondly, to test the effectiveness of gated fusion, we replace the gating mechanism with simple feature addition ($\widetilde{X} + Y$ is fed to self-lattice attention instead of F) and find that the performance drops to 81.22 (-1.40%) F1.

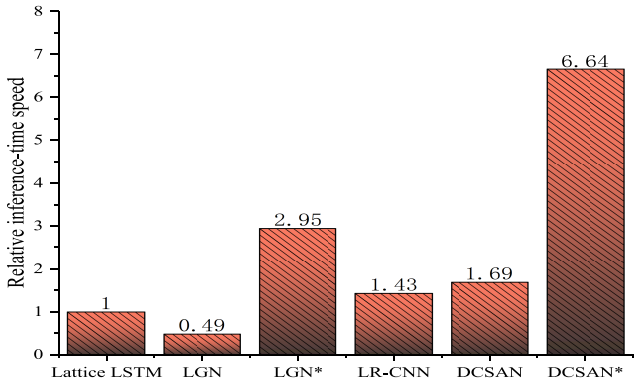


Figure 4: Relative inference-time speed of different models, compared with Lattice LSTM. The default batch size is 1, while * denotes the model is run with 16 batch size.

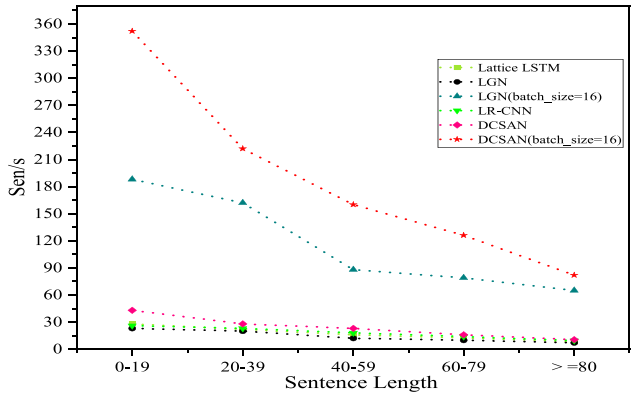


Figure 5: Speed against sentence length. Sen/s denotes the number of sentences processed per second.

Model	P	R	F1
DCSAN	79.63	85.83	82.62
- Cross-lattice attention	78.72	83.34	80.96
- Gated fusion	76.73	86.28	81.22
- Self-lattice attention	78.64	84.56	81.49
- Both attention	78.09	76.89	77.49
- BERT embeddings	77.46	82.90	80.09

Table 3: Ablations on E-commerce test set.

We think the reason is that too much unrelated information hinders the learning process. Then, we attempt to delete the self-lattice attention, and directly use fused representation of character-word pairs (F) for CRF decoding. We observe that the F1 significantly drops by 1.13%, indicating that capturing self-correlations among characters is critical for the Chinese NER task. Moreover, removing both attention modules and using character representations (X) for CRF decoding leads to further worse results on NER (-5.13%) , which suggests that the proposed attention modules play a vital role in the NER task. Finally, we utilize the pre-trained character embeddings used in (Song et al. 2018) instead of BERT embeddings. It leads to significantly worse results on NER (-2.53%), which suggests that BERT embeddings can provide better semantic representations of character sequences.

Performance against Efficiency

To explore the efficiency of our model, we conducted experiments of inference time on the Weibo dataset, as shown in Figure 4. Due to the restriction of DAG structure and variable-sized set of matched words, Lattice LSTM and LR-CNN are non-batch parallel, while LGN and DCSAN can leverage parallel computation of GPU. As we can see, when batch size is set as 16, DCSAN runs 6.64, 4.46, and 2.25 times faster than lattice LSTM, LR-CNN, and LGN respectively. This is due to the multi-head attention that can make better use of GPU parallelism than other baseline models.

To further investigate the influence of sentence length, we analyze the performance of our DCSAN model and other baseline approaches with respect to different grouped sentence lengths on the Weibo dataset, which is shown in Figure 5. We partition the sentence length into five groups ([0-19], [20-39], [40-59], [60-79],[≥80]). We can observe that DCSAN consistently runs faster than compared baselines under different sentence lengths. Especially, when the sentence length is less than 20, DCSAN (batch size=16) runs 12.57, 13.53, and 1.87 times faster than lattice LSTM, LR-CNN, and LGN (batch size=16) respectively. However, the speed gap becomes smaller as the sentence length increases. We think the reason is that the longest sentence becomes an outlier during batch prediction and it slows down the whole decoding process. In summary, the DCSAN model firmly outperforms current RNN-based, CNN-based, and Graph-based methods in terms of efficiency.

Qualitative Analysis

To intuitively verify that our model can better utilize fine-grained correlations in word-character spaces, we analyze two examples from the Weibo test set, as shown in Table 4. In the first case, due to the inherently sequential nature, the character “南 (Nan) ” has only access to its self-matched words “湖南 (Hunan) ” in the Lattice LSTM. Hence, the Lattice LSTM incorrectly recognizes “湖南 (Hunan) ” as a geo-political entity. However, DCSAN can correctly detects the organization entity “湖南广播电视台广告中心 (Hunan Radio and Television Advertising Center)”. The reason is that DCSAN can fully capture fine-grained correlations between characters and matched words, such as a word “广告中心 (Advertising Center) ” corresponds to the character “广 (Guang) ”. In the second case, there is an organization entity “花开公司(Huakai Company) ”. It is difficult for Lattice LSTM to detect the uncommon entity “花开公司(Huakai Company) ” since it lacks cross-modal information, which wrongly recognizes “花开公司(Huakai Company) ” as non-entity. However, DCSAN can exploit cross-modal information. For example, the 4-th character “司 (Division) ” has access to words “开公司(Establish a company) and “花开 (Flowers bloom) ” in “BMES2”, and model close interaction among them. These results indicates that dense interactions between each pair of character and word feature are indispensable and can help model better understand the

	Case 1
Sentence	我参与了@湖南广播电视台广告中心的投票 I participated in the voting of @ Hunan Radio and Television Advertising Center
Gold labels	我 参 与 了 @ 湖 南 广 播 电 视 台 广 告 中 心 的 投 票 O O O O O B-ORG I-ORG I-ORG I-ORG I-ORG I-ORG I-ORG I-ORG I-ORG I-ORG E-ORG O O O
Lattice LSTM	参与 湖南 广播 电视 电视台 广告 中心 投票 我 →参 →与 →了 →@ →湖 →南 →广 →播 →电 →视 →台 →广 →告 →中 →心 →的 →投 →票 O O O O O B-GPE I-GPE O O O O O O O O O O O O
DCSAN	BMES1 ... BMES6 BMES7 BMES8 BMES9 BMES10 ... BMES13 (B:广告中心...; M:湖南广播...; E:南广...; S:广) ... 我 — 参 —与 —了 —@ — 湖 — 南 — 广 — 播 — 电 — 视 — 台 — 广 — 告 — 中 — 心 — 的 —投 —票 O O O O O B-ORG I-ORG I-ORG I-ORG I-ORG I-ORG I-ORG I-ORG I-ORG I-ORG E-ORG O O O
	Case 2
Sentence	花开公司竭诚欢迎您 Huakai company sincerely welcomes you
Gold labels	花 开 公 司 竭诚欢迎您 B-ORG I-ORG I-ORG E-ORG O O O O O
Lattice LSTM	花开 公司, 开公司 竭诚 欢迎 花 →开 → 公 →司 →竭 →诚→欢→迎→您 O O O O O O O O O
DCSAN	BMES1 BMES2 (B:开公司...; M:none; E:花开; S:开) ... 花 — 开 — 公 — 司 — 竭 — 诚 — 欢 — 迎 — 您 B-ORG I-ORG I-ORG E-ORG O O O O O

Table 4: Examples of Weibo dataset. Contents with red and blue colors represent correct and incorrect entities, respectively.

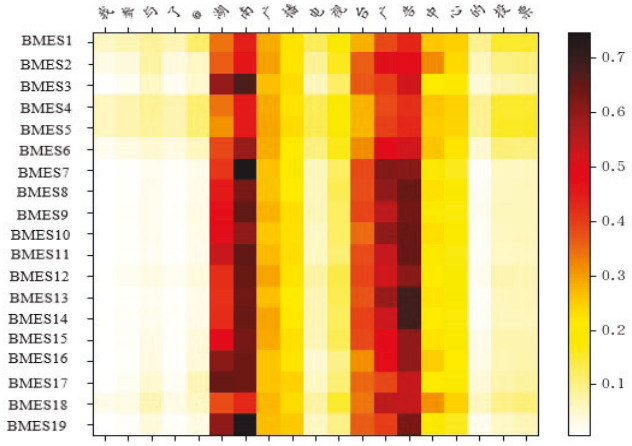


Figure 6: Visualizations of the learned attention maps of the cross-lattice attention over character-word pair on case 1.

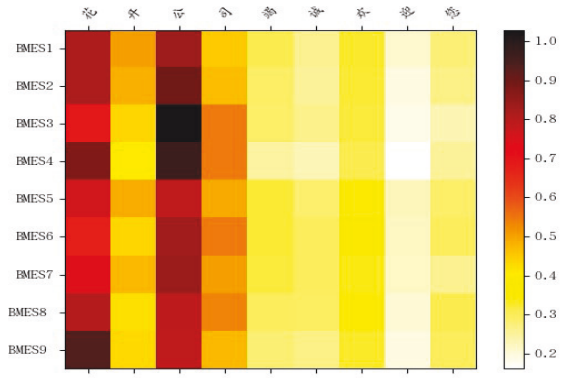


Figure 7: Visualizations of the learned attention maps of the cross-lattice attention over character-word pair on case 2.

contextual semantics.

Moreover, we visualize the cross-lattice attention weights on two cases in Figure 6 and 7. It is first observed that the attention map of case 1 form vertical stripes, and the organization entity "湖南广播电视台广告中心 (Hunan Radio and Television Advertising Center)" involve characters obtain large attention weights. This reveals that the attended features tend to use the feature of "湖南广播电视台广告中心 (Hunan Radio and Television Advertising Center)" for reconstruction. Then, we can find that the attention map of case 2 tend to focus on columns of characters "花 (Flower) " , "开 (Open) ","公 (Public) " and "司 (Division) " . This can be explained by the fact that "花开公司(Huakai Company) " have been reconstructed as the most important information in input features.

Conclusion

In this paper, we propose a Dynamic Cross- and Self-lattice Attention Network (DCSAN) for Chinese NER, which aims to model dense interactions over word-character lattice structure. To achieve this, we introduce a cross-lattice attention module to capture fine-grained correlations between each pair of character and word feature, and present a dynamic self-lattice attention module to model self-correlations inside character sequences. We evaluate the proposed model on four Chinese NER datasets. The results show that DCSAN achieves new state-of-the-art performance compared to other competing approaches, with highly competitive efficiency.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62072465) and the National Key Research and Development Program of China (2018YFB0204301,2020YFC2003400).

References

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of ICCV*, 2425–2433.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; and Hu, G. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101* .

Ding, R.; Xie, P.; Zhang, X.; Lu, W.; Li, L.; and Si, L. 2019. A Neural Multi-digraph Model for Chinese NER with Gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1462–1467.

Gao, P.; Jiang, Z.; You, H.; Lu, P.; Hoi, S. C.; Wang, X.; and Li, H. 2019. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6639–6648.

Gui, T.; Ma, R.; Zhang, Q.; Zhao, L.; Jiang, Y.-G.; and Huang, X. 2019a. Cnn-based chinese ner with lexicon rethinking. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 4982–4988. AAAI Press.

Gui, T.; Zou, Y.; Zhang, Q.; Peng, M.; Fu, J.; Wei, Z.; and Huang, X.-J. 2019b. A Lexicon-Based Graph Neural Network for Chinese NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1039–1049.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of CVPR,2016*, 770–778.

Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *In Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289.

Levow, G.-A. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 108–117.

Li, X.; Yan, H.; Qiu, X.; and Huang, X. 2020. FLAT: Chinese NER Using Flat-Lattice Transformer. *In Proceedings of ACL,2020* .

Liu, W.; Xu, T.; Xu, Q.; Song, J.; and Zu, Y. 2019. An Encoding Strategy Based Word-Character LSTM for Chinese NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2379–2389.

Ma, R.; Peng, M.; Zhang, Q.; Wei, Z.; and Huang, X.-J. 2020. Simplify the Usage of Lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5951–5960.

Nguyen, D.-K.; and Okatani, T. 2018. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6087–6096.

Peng, N.; and Dredze, M. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 548–554.

Song, Y.; Shi, S.; Li, J.; and Zhang, H. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 175–180.

Sui, D.; Chen, Y.; Liu, K.; Zhao, J.; and Liu, S. 2019. Leverage Lexical Knowledge for Chinese Named Entity Recognition via Collaborative Graph Network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3821–3831.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Yan, H.; Deng, B.; Li, X.; and Qiu, X. 2019. TENER: Adapting Transformer Encoder for Name Entity Recognition. *arXiv preprint arXiv:1911.04474* .

Yang, J.; Teng, Z.; Zhang, M.; and Zhang, Y. 2016. Combining discrete and neural features for sequence labeling. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 140–154. Springer.

Yu, Z.; Yu, J.; Cui, Y.; Tao, D.; and Tian, Q. 2019. Deep Modular Co-Attention Networks for Visual Question Answering. In *Proceedings of the CVPR,2019*, 6281–6290.

Zhang, Y.; and Yang, J. 2018. Chinese NER Using Lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1554–1564.

Zhao, S.; Cai, Z.; Chen, H.; Wang, Y.; Liu, F.; and Liu, A. 2019. Adversarial training based lattice LSTM for Chinese clinical named entity recognition. *Journal of biomedical informatics* 99: 103290.

Zhao, S.; Hu, M.; Cai, Z.; and Liu, F. 2020. Modeling Dense Cross-Modal Interactions for Joint Entity-Relation Extraction. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 4032–4038. ijcai.org. doi:10.24963/ijcai.2020/558. URL https://doi.org/10.24963/ijcai.2020/558.