Enhancing Chinese Character Representation With Lattice-Aligned Attention

Shan Zhao, Minghao Hu, Zhiping Cai, Zhanjun Zhang, Tongqing Zhou, and Fang Liu, *Member, IEEE*

***Abstract*— Word–character lattice models have been proved to be effective for some Chinese natural language processing (NLP) tasks, in which word boundary information is fused into character sequences. However, due to the inherently unidirectional sequential nature, prior approaches have only learned sequential interactions of character–word instances but fail to capture fine-grained correlations in word–character spaces. In this article, we propose a lattice-aligned attention network (LAN) that aims to model dense interactions over word–character lattice structure for enhancing character representations. By carefully combining cross-lattice module, gated word–character semantic fusion unit, and self-lattice attention module, 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. Experimental results on three Chinese NLP benchmark tasks demonstrate that LAN obtains state-of-the-art results compared to several competitive approaches.**

***Index Terms*— Attention, Chinese, information extraction, interactions, lattice.**

I. Introduction

MANY Chinese information extraction tasks, including Chinese word segmentation (CWS), Chinese named entity recognition (NER), and Chinese relation extraction (RE), require determining and categorizing word boundary, which are fundamental tasks in the field of natural language processing (NLP). These basic tasks have attracted increasing attention due to their important role in many downstream NLP tasks, such as knowledge base population [1]–[3] and question answering [4], [5]. However, Chinese cannot use explicit delimiters (e.g., white space) to separate words in written text, which is very different from the English writing system.

Generally, there are two major methodologies for these Chinese information extraction tasks: word-based models [6]–[8] and character-based models [9]–[12]. The major disadvantage of word-based models is that word information can be utilized only for readily recognized words, namely those that are already in the output candidates. Character-based models, on the other hand, regard each input sentence as a character sequence, which can naturally avoid word segmentation errors, thus outperforming word-based methods. However, in most cases, the semantic of a single Chinese character is ambiguous. For example, the character “处” in word “处理 (Handle)” and “处长 (Director)” has entirely different meanings. Moreover, several recent works [13]–[16] have demonstrated that integrating word information into character sequences via word–character lattice structure can lead to better language understanding and accordingly benefits various Chinese NLP tasks. For example, Zhang and Yang [13] proposed a lattice LSTM structure, which can utilize the words information in the NER task. Yang *et al.* [16] extended the lattice LSTM structure using subword encoding in the CWS task. Li *et al.* [15] introduced a multigrained lattice framework (MG lattice) for Chinese RE task to take advantage of multigrained language information.

Prior approaches mainly enhance character-level LSTM encoder with a directed acyclic graph (DAG) structure by adding word level as external knowledge, referred to as word–character lattice LSTM structure. However, these lattice methods can only learn sequential interactions of character–word instances but fail to model dense interactions between each character and each matched word for enhancing character representations. Taking the sentence in Fig. 1(a) as an example, the character “京 (Capital)” has only access to its self-matched words “南京 (Nanjing)” in the lattice LSTM. Yet, we argue that it is beneficial for enhancing character representations when the character “京 (Capital)” can be aware of “市 (City)” being matched with “南京市 (Nanjing City)”.

To address the above issue, we propose a lattice-aligned attention network (LAN) for enhancing character representations. The key insight comes from multimodal learning in computer vision [17], [18], 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

Manuscript received 21 October 2020; revised 10 February 2021 and 29 June 2021; accepted 17 September 2021. Date of publication 5 October 2021; date of current version 7 July 2023. This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1405702. *(Corresponding author: Fang Liu.)*

Shan Zhao is with the School of Design, Hunan University, Changsha 410082, China, and also with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: zs50910@mail.ustc.edu.cn).

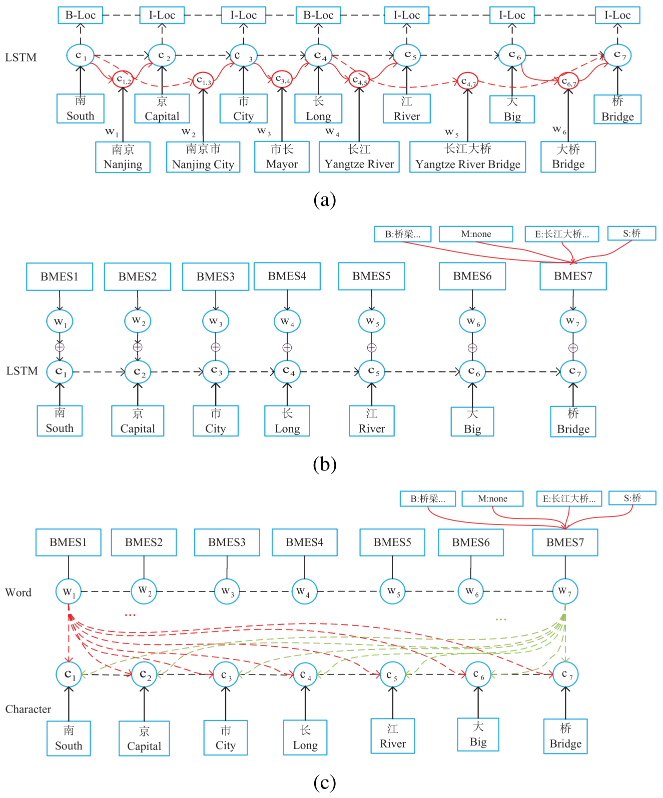
Minghao Hu is with the PLA Academy of Military Science, Beijing 100000, China (e-mail: huminghao16@@gmail.com).

Zhiping Cai, Zhanjun Zhang, and Tongqing Zhou are with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: zpcai@nudt.edu.cn; zhangzhanjun@nudt.edu.cn; zhoutongqing@nudt.edu.cn).

Fang Liu is with the School of Design, Hunan University, Changsha, Hunan 410082, China (e-mail: fangl@hnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3114378.

Digital Object Identifier 10.1109/TNNLS.2021.3114378


B-Loc
I-Loc
LSTM
c_1 c_2 c_3 c_4 c_5 c_6 c_7
$c_{1,2}$ $c_{1,3}$ $c_{3,4}$ $c_{4,5}$ $c_{4,7}$ $c_{6,7}$
南 South
京 Capital
市 City
长 Long
江 River
大 Big
桥 Bridge
w_1 南京 Nanjing
w_2 南京市 Nanjing City
w_3 市长 Mayor
w_4 长江 Yangtze River
w_5 长江大桥 Yangtze River Bridge
w_6 大桥 Bridge
B:桥梁...
M:none
E:长江大桥...
S:桥
BMES1 BMES2 BMES3 BMES4 BMES5 BMES6 BMES7
w_1 w_2 w_3 w_4 w_5 w_6 w_7
Word
Character
...

(a)

(b)

(c)

Fig. 1. (a) Example of the original lattice LSTM model [13]; (b) Soft-lexicon strategy used [19]; (c) Dense interactions of lattice inputs in our model. “BMESi” denotes the aligned word for the ith character. B, M, and E mean all lexicon matched words on a sentence that begin, middle, and end with the ith character, respectively. S is the single-character word. “none” denotes the corresponding word set that is empty.

spaces. Then, we propose a gated fusion unit to dynamically integrate character and word features. After that, we further construct a self-lattice attention module that is capable of building direct connections between two arbitrary characters despite their distances. Given the word–character embeddings and the aligned lattice structure, LAN first utilizes the cross-lattice attention module to generate word-aware character features and then adopts the gated fusion unit and 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 a rich semantic feature for enhancing character representations.

Finally, we conducted extensive experiments on three Chinese NLP tasks, including CWS, Chinese NER, and Chinese RE, and nine public benchmark datasets to evaluate the proposed model. Experimental results show that LAN can achieve the state-of-the-art performance compared to a variety of competitive approaches.

II. Related Work

A. Word–Character Lattice Structure

Lattice RNNs have been first used to model speech tokenization lattice [12], [20] and multigranularity segmentation for NMT [21]. Then, since word sequence information is potentially useful for character-based sequence learning, Zhang and Yang [13] proposed a lattice LSTM model to explicitly leverage word boundary information, in which matched lexical words are encoded into character sequences with a DAG structure. Lattice LSTM model has outperformed both word- and character-based approaches by a large margin on the Chinese NER task. Later, Yang *et al.* [16] extended the lattice LSTM structure by using subword encoding that does not rely on any external segment on CWS tasks, which gives competitive results with previous state-of-the-art methods on four segmentation benchmarks. Recently, Tian *et al.* [22] proposed a neural framework, WMSEG, which uses memory networks to incorporate wordhood information with several popular encoder–decoder combinations for CWS task and achieve the state-of-the-art performance on several datasets. Tian *et al.* [23] proposed a neural model named TWASP for joint CWS and POS tagging following the character-based sequence labeling paradigm, where a two-way attention mechanism is used to incorporate both context feature and their corresponding syntactic knowledge for each input character. Moreover, Li *et al.* [14] exploited lattice LSTM, which comprehensively utilizes both internal information and external knowledge, to conduct the Chinese RE task. 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.* [14] explored four different words encoding strategies to alleviate this issue. Gui *et al.* [24] proposed a CNN-based NER model (LR-CNN) that encodes matched words at different window sizes. Moreover, Gui *et al.* [25] and Sui *et al.* [26] converted lattice into graph and used graph neural networks (GNNs) for encoding. However, as sequence labeling tasks are very sensitive to sentence structure, these methods still need to use LSTMs as backbone encoder, which makes the models complicated. Recently, Yan *et al.* [27] proposed an adapted transformer encoder for Chinese NER. Ma *et al.* [19] constructed the soft-lexicon feature to encoding the matched words, obtained from the lexicon, into the representations of characters. More similar to our work is the recent approach FLAT by [28], which also applies multihead attention mechanism as their core model. FLAT leveraged a flat lattice structure so that transformer can capture word information via position encoding. Compared with FLAT, our proposed method view character and word sequences as two modalities and can dynamically fuse multimodal features with intramodality and intermodality information.

B. Multimodal Learning

Multimodal learning is widely explored in computer vision and NLP. A typical task is visual question answering (VQA) [29], [30], which requires the model to perform fine-grained semantic understanding of both the image and the question. For example, Nguyen and Okatani [31] proposed a dense symmetric co-attention architecture to form a hierarchy for multistep interactions between an image–question pair. Yu *et al.* [18] introduced a VQA model that consists of multiple modular co-attention layers cascaded in depth. Gao *et al.* [17] proposed to dynamically fuse multimodal features with intramodality and intermodality 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.

III. Methodology

The primary goal of this work is to model dense interactions over word–character lattice for enhancing character representations in several Chinese NLP tasks. Fig. 2 shows the overall architecture of our LAN. We first construct the word–character lattice structure by applying the soft-lexicon feature strategy and then obtain fixed-dimensional representations of both character and word sequences (Section III-A). Next, we utilize lattice-aligned attention (Section III-B) to explicitly model dense interactions across different feature spaces. Finally, we apply a conditional random field (CRF) and a relation classifier to perform the decoding for several Chinese NLP tasks (Section III-C).

A. Word–Character Lattice Representations

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

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

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

where e^c denotes a pretrained BERT character embedding lookup table. The character feature representations for NER and CWS tasks can be obtained as follows:

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

For RE task, as pointed out in some previous studies [15], [32], entity position embeddings are important for relation classification. Therefore, we incorporate position embeddings if experiments are carried out on the RE task. Here, entity position embeddings denoted relative distances from the current character to head and tail entities. These position embeddings aim to specify entity pairs. Specifically, the relative distances from the ith character c_i to the two marked entities are denoted as p_i^1 and p_i^2, respectively. They are calculated as follows:

$$p_i^1 = \begin{cases} i - b^1, & i < b^1 \\ 0, & b^1 \le i \le e^1 \\ i - e^1, & i > e^1 \end{cases} \tag{3}$$

where b^1 and e^1 are the start and end indices of the head entity, respectively. The computation of p_i^2 i is similar to (3). In our work, we concatenate x_i, p_i^1, and p_i^2 as character feature representations for RE task. Moreover, we exploit a linear projection to transform dimension for facilitating calculation. The final character feature representations for RE task are calculated as follows:

$$X = \text{linear}\big[x_1', x_2', x_3', \ldots, x_n'\big], \quad \in \mathbb{R}^{n*d} \tag{4}$$

$$x_i' = \big[x_i, p_i^1, p_i^2\big]. \tag{5}$$

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 ith character to the jth character. Taking the sentence in Fig. 1(a) for example, $c_{1,3}$ refers to the word "南京市 (Nanjing City)." In the original lattice model, the ith character is aligned with a set of matched words $w_i = [c_{k,i}, \ldots, c_{j,i}]$, where $k, j < i$. For instance, the set of matched words for the character "桥 (Bridge)" is $w = [c_{4,7}, c_{6,7}]$, 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 [19], as shown in Fig. 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

$$y_i = [v(B(c_i)); v(M(c_i)); v(E(c_i)); v(S(c_i))], \quad y_i \in \mathbb{R}^{4d} \tag{6}$$

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

$$v(p) = \frac{1}{Z} \sum_{w \in p} (z(w) + b) e^w \tag{7}$$

where $z(w)$ denotes the frequency of w_c occurring in the statistic data set, w_c is the character sequence constituting w, e^w represents a pretrained word embedding, and 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:

$$Z = \sum_{w \in (B \sqcup M \sqcup E \sqcup S)} z(w) + b. \tag{8}$$

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

$$Y = \text{Linear}[y_1, y_2, y_3, \ldots, y_n] \in \mathbb{R}^{n*d}. \tag{9}$$

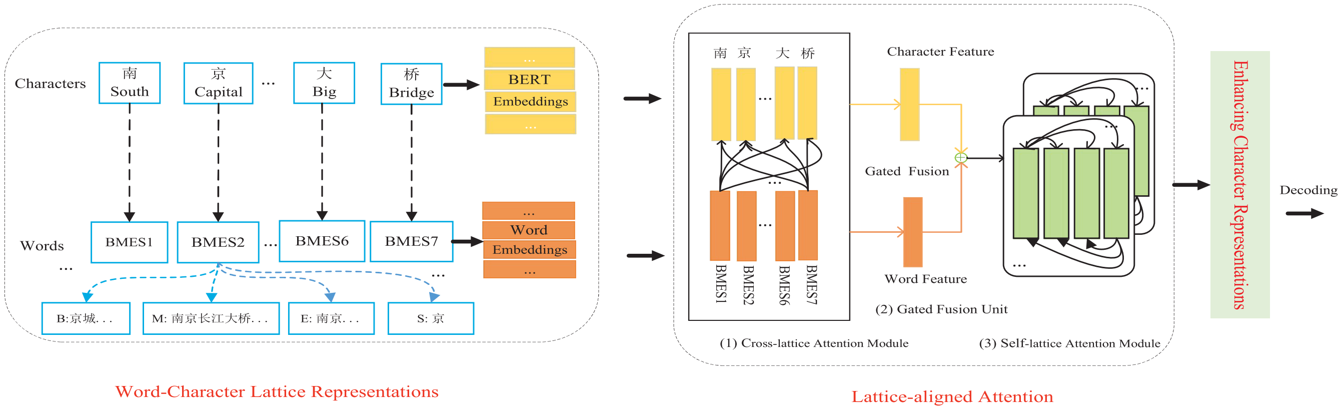


Fig. 2. Overall flowchart of LAN. Word–character lattice structure is first constructed by characters and aligned words distributed representations. Then, lattice-aligned attention, which contains three main components: 1) cross-lattice attention module; 2) a gated fusion unit; and 3) a self-lattice module, is designed to explicitly capture dense interactions over word–character lattice structure. Finally, enhancing character representations is fed to decode layer for several Chinese NLP tasks.

B. Lattice-Aligned Attention

In this section, we present our proposed lattice-aligned attention for enhancing character representations, which contains three main components: 1) cross-lattice attention module; 2) a gated fusion unit; and 3) a self-lattice module.

1) Cross-Lattice Attention: Cross-lattice attention module [see Fig. 3(a)] aims to capture fine-grained correlations between character and word feature representations, which is a variant of recently proposed m-head cross-modal attention mechanism [33], by treating X as queries and Y as keys and values. It is capable of modeling dense interactions between each pair of character and word feature, which is calculated as

$$\alpha_{wi}(Y, X, X) = \text{softmax}\left(\frac{\left[Y W_i^Q\right]\left[X W_i^K\right]^T}{\sqrt{d}}\right)\left[X W_i^V\right] \quad (10)$$

$$\text{Inter}_{X\to Y} = [\alpha_{w1}(Y, X, X), \ldots, \alpha_{wm}(Y, X, X)]W_m \quad (11)$$

where α_{wi} refers to the ith head of cross-modal attention and $W_i^Q \in \mathbb{R}^{d*d/m}$, $W_i^K \in \mathbb{R}^{d*d/m}$, $W_i^V \in \mathbb{R}^{d*d/m}$, and $W_m \in \mathbb{R}^{d*d}$ are trainable parameter matrices. Then, 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:

$$\widetilde{X} = \text{Linear}[X; \text{Inter}_{X\to Y}]. \quad (12)$$

Now, cross-lattice attention learns the pairwise relationship between each paired sample $\langle x_i, y_j\rangle$ within X and Y and fuses feature representations to generate word-aware character features. Compared with original Lattice structures, which have only access to its self-matched words, each character can directly interact with all matched words in cross-lattice attention.

2) Gated Fusion of Character–Word Pairs: We design a gated fusion unit to integrate character and word features. 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:

$$h_c = \tanh\left(\widetilde{X} W_c + b_c\right) \quad (13)$$

$$h_w = \tanh(Y W_w + b_w) \quad (14)$$

$$g = \sigma\left(([h_c; h_w])W_g\right) \quad (15)$$

$$F = g\widetilde{X} + (1 - g)Y \quad (16)$$

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

3) Self-Lattice Attention: Self-lattice attention with relative position encoding [see Fig. 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 $\langle f_i, f_j\rangle$ within F, and outputs attended self-aware character features by using weighted summation across all instances. This module is a variant of multihead attention mechanism, which is calculated as follows:

$$\text{head}_i = \text{softmax}\left(\left(Q W_i^Q\right)K[i]^T + P[i]\right)\left(V W_i^V\right) \quad (17)$$

$$O = \left[\text{head}_1; \ldots; \text{head}_z\right]W^o \quad (18)$$

where Q, K, and 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 ith partition of K, and $P[i] \in \mathbb{R}^{n*n}$ contains relative position information of the ith partition.

To explicitly inform the module with positional information, we utilize the relative position encoding method of which details can be found from [27]. 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, and then,

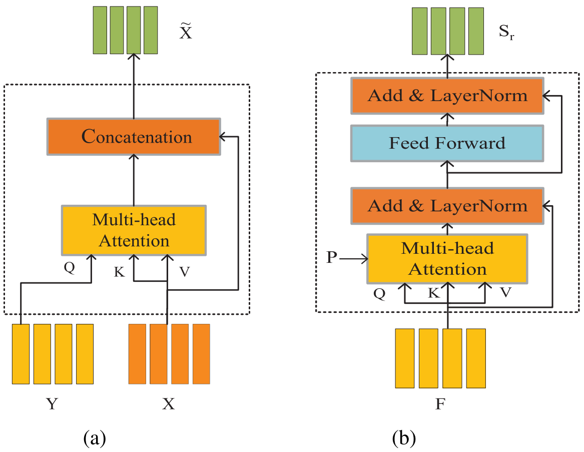


Fig. 3. Two proposed attention modules. Cross-lattice attention aims to model dense interactions between each pair of character and word features, while self-lattice attention is used to capture character-to-character self-correlations. (a) Cross-lattice attention. (b) Self-lattice attention.

the relative position encoding $P[i]$ can be calculated as

$$d = \text{Linear}\left[\mathrm{d_p}\right] \tag{19}$$

$$m = (2b * z)/d \tag{20}$$

$$R_{t-j} = \left[\cdots \sin\left(\frac{t-j}{10000^m}\right)\cos\left(\frac{t-j}{10000^m}\right)\cdots\right]^T \tag{21}$$

$$P[i]_{t,j} = \left(QW_i^Q\right)R_{t-j} + uK[i]_j^T + vR_{t-j}^T \tag{22}$$

where $u, v \in \mathbb{R}^{d/z}$ are learnable parameters. b in (17) is in the range $[0; d/(2z)]$, and z is the number of heads. To facilitate calculation, we utilize a linear projections to transform position embeddings dimension (d_p).

In our network, the output O of the multihead attention will be further processed by residual connection [34] and layer normalization [35] followed by position-wise feedforward networks, which can be computed as follows:

$$Rc = \text{LayerNorm}(O + F) \tag{23}$$

$$\text{FFN}(Rc) = \max(0; RcW_1 + b_1)W_2 + b_2 \tag{24}$$

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

$$Sr = \text{LayerNorm}(\text{FFN}(Rc) + Rc). \tag{25}$$

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

C. Decoding and Training for Different Tasks

In this section, we describe how enhancing character representations can be used for different NLP tasks. The enhancing character representations are carried out on three Chinese NLP tasks.

1) Chinese NER and CWS: Chinese NER and CWS can be formalized as character-level sequence labeling tasks, in which we need to predict a label for each character. A standard CRF layer is used to predict character taggings, which takes Sr^l as inputs and outputs a sequence of predicted tagging probabilities $A = [a_1, \ldots, a_n]$. Let A' denotes an arbitrary label distribution sequence (i.e., B-begin, I-inside, O-outside (BIO) tagging scheme), the probability of the label sequence A can be calculated using a softmax function

$$Pr\left(A|Sr^l\right) = \frac{\prod_{i=1}^{n} \varphi_n\left(a_{n-1}, a_n, Sr^l\right)}{\sum_{a' \in A'} \prod_{i=1}^{n} \varphi_n\left(a'_{n-1}, a'_n, Sr^l\right)} \tag{26}$$

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, respectively. During training, we optimize model parameters by minimizing the following conditional likelihood:

$$\mathcal{L}(\theta) = -\log Pr\left(A|Sr^l\right) \tag{27}$$

where θ indicates all parameters of our model.

2) Chinese RE: Chinese RE aims to extract semantic relations between entity pairs in natural language sentences. A relation classifier is used to predict the single label for the entire sentence. We first adopt a character-level attention to integrate the enhancing character representations Sr^l into a sentence representations V_h

$$Sr^{l'} = \tanh\left(Sr^l\right) \tag{28}$$

$$\alpha = \text{softmax}\left(Sr^{l'} W_h\right) \tag{29}$$

$$V_h = \alpha Sr^l. \tag{30}$$

Then, to compute the conditional probability of each relation, the sentence representations V_h is fed into a softmax classifier

$$R = W_o V_h + b \tag{31}$$

$$Pr(T|S) = \text{softmax}(R) \tag{32}$$

where $W_o \in \mathbb{R}^{k*d}$ is the transformation matrix and $b \in \mathbb{R}^k$ is a bias vector. k indicates the total number of relation types, and T is the estimated probability for each type. During training, given all (M) training examples $(S^i; T^i)$, we optimize the parameters of the model by minimizing the following cross-entropy for RE

$$\mathcal{L}(\theta) = \sum_{i=1}^{M} \log Pr\left(T^i|S^i, \theta\right) \tag{33}$$

where θ indicates all parameters of our model.

IV. Experiments

A. Experimental Setup

To evaluate the performance of our model, we conduct experiments on three Chinese NLP tasks and nine public benchmark datasets, of which detailed statistics are shown in Table I.

TABLE I
STATISTICS OF NINE CHINESE NLP DATASETS

Tasks	Dataset	Type	Train	Dev	Test
NER	Weibo	Char	73.8K	14.5K	14.8K
	Resume	Char	124.1K	139K	15.1K
	MSRA	Char	2,169.9K	-	172.6K
CWS	PKU	Char	1,826K	-	173K
	MSR	Char	4,050K	-	184K
	AS	Char	8,368K	-	198K
	CITYU	Char	2,403K	-	68K
RE	SanWen	Char	515K	55K	68K
	FinRE	Char	727K	81K	203K

TABLE II
MAIN RESULTS (F1) FOR NER TASK

Models	Resume	MSRA	Weibo
LR-CNN [24]	95.11	93.71	59.92
LGN [25]	95.37	93.46	59.84
TENER [27]	95.00	92.74	58.39
FLAT [28]	95.45	94.35	63.42
FLAT +BERT [28]	95.86	96.09	68.55
Char-level LSTM [25]	93.48	88.81	52.77
Lattice LSTM [13]	94.46	93.18	58.79
LAN (ours)	**96.67**	**96.41**	**71.27**

TABLE III
MAIN RESULTS (F1 AND AUC) FOR RE TASK. † DENOTES THE RESULTS THAT ARE PRODUCED BY IMPLEMENTATION IN [14]

Method	FinRE		SanWen	
	F1	AUC	F1	AUC
CNN [32]†	41.47	27.12	59.42	47.81
PCNN [42] †	45.51	30.49	61.00	48.26
PCNN-Att [43]†	46.13	31.89	60.55	50.41
MG Lattice [14]	49.26	38.74	65.61	57.33
Char-level LSTM [44] †	42.87	28.80	61.04	50.21
Lattice LSTM [13]†	47.41	36.58	63.88	56.88
LAN (ours)	**51.35**	**40.68**	**69.85**	**66.32**

1) Chinese NER Task: The Weibo NER dataset [36] is drawn from the Chinese social media network Sina Weibo. The MSRA dataset [37] comes from news written in simplified Chinese. The Chinese resume dataset [13] contains resumes crawled from SinaFinance text.

2) CWS Task: We evaluate our model on four standard CWS datasets: PKU, MSR, AS, and CITYU. They are taken from the SIGHAN 2005 bake-off [38] with standard data split. AS and CITYU are in traditional Chinese characters. Following previous studies [22], [39], we convert traditional Chinese characters in AS and CITYU into simplified ones.

3) Chinese RE Task: We take two publicly available Chinese RE datasets to evaluate the performance of our model. The first one is the Chinese SanWen [8], which contains sentences with annotated relations extracted from 837 Chinese literature articles. The second one is the FinRE dataset [15]. This dataset aims to extract 44 distinguished relationships from financial news in Sina Finance.

B. Implementation Details

We utilize the BERT embedding as our character embeddings. The BERT in the experiment is "BERT-wwm" released in [40]. We use the word embedding dictionary [41] that contains over 8000k Chinese characters and words as default lexicon in our model. As for hyperparameter configurations, the sizes of character embeddings are 768, position embeddings are 25, and word embeddings are 200 by default, and the dimensionality of hidden size is 768. For attention settings, the head number of cross-lattice attention and self-lattice attention is 8 and 4, respectively, for all datasets. We set the number of self-lattice attention layers l as 2 by default. To train the model, we use the stochastic gradient descent (SGD) optimizer with a learning rate of 0.0007 on all datasets. The training takes 100 epochs until convergence. We adopt standard F1-score and area under the curve (AUC) to evaluate the model. All experiments were conducted on a single NVIDIA 1080Ti GPU.

C. Overall Results

Tables II–IV show the performances on three Chinese NLP tasks with nine public datasets. Generally, our method LAN consistently outperforms all baselines on all three tasks, which demonstrates the effectiveness and universality of the proposed approach. Moreover, lattice-based models significantly outperform character-level models on all datasets from different tasks, which indicates that incorporating the word information plays a vital role in the Chinese NLP tasks.

In detail, on the three datasets of NER task, it can be seen that our model achieves the state-of-the-art performance by obtaining 96.67, 96.41, and 71.27 F1. 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. When compared to the lattice LSTM, we find stronger performance improvement with respect to Resume (+2.21%), MSRA (+3.23%), and Weibo (+12.48%). Besides, introducing word boundary information into the encoding of character sequences improves the RE performance on FinRE and SanWen datasets by 2.09 and 4.24 points and 1.94 and 8.99 points in F1 and AUC, respectively. For the CWS task, we obtain 96.69%, 98.61%, 97.02%, and 98.13% F1 on PKU, MSR, AS, and CITYU datasets, respectively. Compared to the lattice LSTM, our approach significantly increases by 1.68%, 1.49%, 1.39%, and 1.18% on four datasets.

D. Ablation Study

We conduct an ablation study on the Weibo, SanWen, and PKU test sets to investigate the influence of different modules in our proposed LAN model in Table V. Modules are tested in five ways.

TABLE IV
MAIN RESULTS (F1) FOR CWS TASK. ∗ DENOTES THE RESULTS THAT ARE PRODUCED BY OUR IMPLEMENTATION

Models	PKU	MSR	AS	CITYU
Lattice+Subword [16]	95.80	97.80	-	-
Switch-LSTMs [45]	96.15	97.78	95.22	96.22
Unified model [39]	96.41	98.05	96.44	96.44
WMSEG (BERT-CRF) [22]	96.53	98.40	96.62	97.93
Char-level LSTM [46]	94.32	96.04	94.75	95.55
Lattice LSTM∗ [13]	95.01	97.12	95.63	96.95
LAN (ours)	**96.69**	**98.61**	**97.02**	**98.13**

TABLE V
ABLATIONS ON WEIBO, SANWEN, AND PKU TEST SETS

Model	Weibo	SanWen	PKU
LAN	71.27	69.85	96.69
- Cross-lattice attention	69.31	68.33	96.03
- Gated fusion	69.13	68.12	95.96
- Self-lattice attention	69.92	69.02	96.23
- Both attention	67.66	67.13	95.53
- BERT embeddings	67.33	65.42	95.98

1) We remove the cross-lattice attention module and only use the gate fusion unit and self-lattice attention module for encoding. We find that the F1-score obviously decreases by 1.96, 1.52, and 0.66 on three datasets, showing the beneficial effect of modeling dense interactions among word–character feature spaces.
2) 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 69.02 (-2.14%), 88 (-1.73%), and 96.21 (-0.73%) F1 on three datasets. We think that the reason is that too much unrelated information hinders the learning process.
3) We attempt to delete the self-lattice attention and directly use the fused representation of character–word pairs (F) for decoding. We observe that the F1 significantly drops by 1.35%, 0.83%, and 0.46% on three datasets, indicating that capturing self-correlations among characters is critical for enhancing character representations.
4) Removing both attention modules and using character representations (X) for decoding leads to further worse results on Weibo (-3.61%), SanWen (-2.72%), and PKU (-1.16%), which suggests that the proposed attention modules play a vital role in enhancing character representations.
5) We utilize the pretrained character embeddings used in [41] instead of BERT embeddings. It leads to significantly worse results on Weibo (-3.94%), SanWen (-4.43%), and PKU (-0.71%), which suggests that BERT embeddings can provide better semantic representations of character sequences.

E. Performance Against Efficiency

To explore the efficiency of our model, we conducted experiments of inference time on all datasets. Tables VI and VII list the relative decoding time on all test sets, compared to the lattice LSTM. As we can see, our LAN not only achieves better F1-score results than the baseline model but also is much faster. Specifically, LAN runs an average of 6.28 times faster than lattice LSTM. The reason is that the lattice LSTM extends the already slow LSTM to a DAG structure by adding word level. However, our model is based on multihead attention, which can make better use of GPU parallelism. Moreover, due to the restriction of DAG structure and variable-sized set of matched words, lattice LSTM is nonbatch parallel, while LAN can leverage parallel computation of GPU.

TABLE VI
RELATIVE DECODING-TIME SPEED OF DIFFERENT MODELS ON CWS TASK. LATTICE LSTM CAN ONLY RUN WITH BATCH SIZE = 1, WHILE OUR LAN MODEL RUNS WITH BATCH SIZE = 16

Method	PKU		MSR		AS		CITYU	
	F1	Speedup	F1	Speedup	F1	Speedup	F1	Speedup
Lattice LSTM	95.11	1x	97.12	1x	95.63	1x	96.95	1x
LAN (ours)	96.69	5.6	98.61	6.2	97.02	6.3	98.13	5.8

To further investigate the influence of sentence length, we analyze the performance of our LAN model and other baseline approaches with respect to different grouped sentence lengths on the Weibo dataset, which is shown in Fig. 4. We partition the sentence length into five groups ([0–19], [20–39], [40–59], [60–79], [≥80]). We can observe that LAN consistently runs faster than compared baselines under different sentence lengths. Especially, when the sentence length is less than 20, LAN (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 that the reason is that the longest sentence becomes an outlier during batch prediction and it slows down the whole decoding process. Moreover, we can find that FLAT outperforms our proposed method in terms of efficiency. The reason is that our proposed method cascaded attention units and gating mechanisms. In summary, the LAN model firmly outperforms current RNN-based (lattice LSTM), CNN-based (LR-CNN), and graph-based methods (LGN) in terms of efficiency.

F. Lexicon and Embeddings

To analyze the influence of lexicon and pretrained word embeddings (direction skip-gram model training), we evaluate some comparative experiments by using the same word lexicon with and without pretrained embeddings on the Weibo test set. Moreover, to further analyze the contribution from word lexicon, we also conduct experiments with character-only information by running self-lattice attention. To make the comparison fair, we set all hyperparameters unchanged. The result is shown in Fig. 5. As can be seen from the figure, replacing pretrained word embeddings with randomly initialized embeddings changes the performance: the F1-score decreases by 2.94%. Compared with models incorporating lexicons, the performance of the model without lexicons is seriously degraded, and the F1-score decreases by 4.05%. These results demonstrate the excellent contribution from word

TABLE VII

RELATIVE DECODING-TIME SPEED OF DIFFERENT MODELS ON NER AND RE TASKS. LATTICE LSTM CAN ONLY RUN WITH BATCH SIZE = 1, WHILE OUR LAN MODEL RUNS WITH BATCH SIZE = 16

Method	Resum		MSRA		Weibio		FinRE		SanWen	
	F1	Speedup	F1	Speedup	F1	Speedup	F1	Speedup	F1	Speedup
Lattice LSTM	94.46	1x	93.18	1x	58.79	1x	47.41	1x	63.88	1x
LAN (ours)	96.67	5.3	96.47	7.1	71.27	6.4	51.35	6.8	69.85	7.1

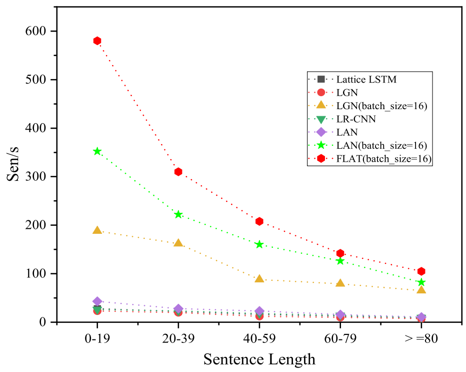


Fig. 4. Speed against sentence length. Sen/s denotes the number of sentences processed per second. Due to the restriction of the DAG structure or variable-sized lexical words set, lattice LSTM and LR-CNN are nonbatch parallel.

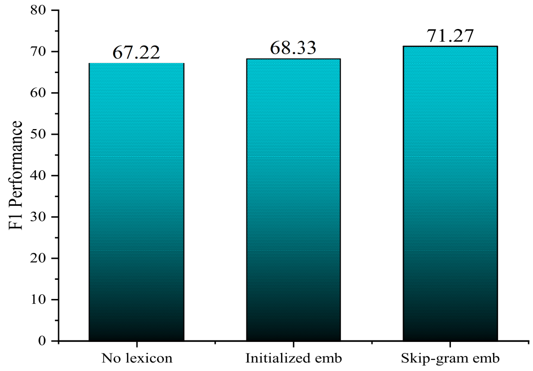


Fig. 5. Comparison F1-scores between LAN with and without pretrained word embeddings and lexicon on the Weibo test set. "No lexicon" denotes only character information that is fed to self-lattice attention. "Initialized emb" and "Skip-gram emb" denote the word embeddings that are obtained by randomly initialized and direction skip-gram model training, respectively.

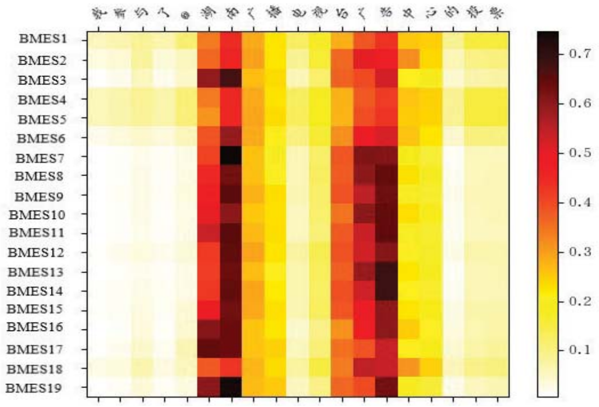


Fig. 6. Visualizations of the learned attention maps of the cross-lattice attention over character–word pair on case 1.

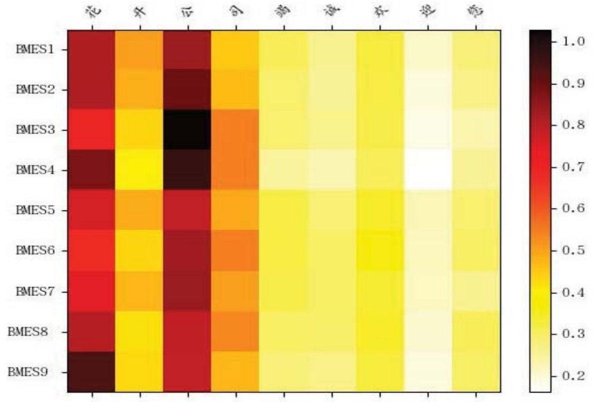


Fig. 7. Visualizations of the learned attention maps of the cross-lattice attention over character–word pair on case 2.

lexicon and pretrained word embeddings and also explains the effectiveness of our model in different domains.

G. 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 VIII. 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, LAN 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 LAN 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 nonentity. However, LAN can exploit cross-modal information. For example, the fourth 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 indicate that dense interactions between each pair of character and word feature are indispensable and can help model better understand the contextual semantics.

TABLE VIII

EXAMPLES OF WEIBO DATASET. CONTENTS WITH RED AND BLUE COLORS REPRESENT CORRECT AND INCORRECT ENTITIES, RESPECTIVELY

	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
LAN	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
LAN	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 IX

ERROR ANALYSIS ON WEIBO DATASET. CONTENTS WITH RED AND BLUE COLORS REPRESENT CORRECT AND INCORRECT ENTITIES, RESPECTIVELY

	Example 1
Sentence	分手大师贵仔邓超四大名捕围观话筒转发 Breaking up master Guizai Deng Chao four famous catch onlookers microphone forwarding
Gold labels	分 手 大 师 贵 仔 邓 超 四 大 名 捕 围 观 话 筒 转 发 O O O O O O B-PER E-PER O O O O O O O O O O
Predicted	分 手 大 师 贵 仔 邓 超 四 大 名 捕 围 观 话 筒 转 发 O O O O O O B-PER E-PER O O B-PER E-PER O O O O O O
	Example 2
Sentence	哈哈米八吴够历史要的陈小奥丁丁我爱小肥肥一族 Haha Mi Ba Wu is enough in history, Chen Xiao AO Ding Ding, I love the little fat family
Gold labels	哈 哈 米 八 吴 够 历 史 要 的 陈 小 奥 丁 丁 我 爱 小 肥 肥 一 族 O O O O O O O O O O B-PER I-PER E-PER O O O O O O O O O O
Predicted	哈 哈 米 八 吴 够 历 史 要 的 陈 小 奥 丁 丁 我 爱 小 肥 肥 一 族 O O O O O O O O O O B-PER I-PER E-PER B-PER E-PER O O O O O O O

Moreover, we visualize the cross-lattice attention weights on two cases in Figs. 6 and 7. It is first observed that the attention map of case 1 forms vertical stripes, and the organization entity "湖南广播电视台广告中心 (Hunan Radio and Television Advertising Center)" involves characters to 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.

H. Error Analysis

To gain further insights about our best-performing model, we conducted an error analysis (see Table IX). We can observe that "邓超 (Deng Chao)" and "陈小奥 (Chi Xiao Ao)" can be can correctly be detected as person entities by our LAN on examples 1 and 2. However, our LAN incorrectly recognizes "名捕 (Famous arrest)" and "丁丁 (Ding ding)" as person entities. Moreover, the overall performance on WeiboNER dataset is relatively low. We think that the reason may be that social media texts do not follow strict syntactic rules. Besides, too many words of information still bring interference to a certain extent, although we used the gated fusion unit.

V. CONCLUSION

In this article, we propose an LAN for enhancing character representations, 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 gated fusion unit and a self-lattice attention module to model self-correlations inside character sequences.

We evaluate the proposed model on three Chinese NLP tasks. The results show that LAN achieves new state-of-the-art performance compared to other competing approaches.

References

[1] S. N. Tran and A. S. d'Avila Garcez, "Deep logic networks: Inserting and extracting knowledge from deep belief networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 246–258, Feb. 2018.

[2] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," in *Proc. ACL*, 2019, pp. 241–251.

[3] S. Zhao, M. Hu, Z. Cai, and F. Liu, "Dynamic modeling cross-modal interactions in two-phase prediction for entity-relation extraction," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 25, 2021, doi: 10.1109/TNNLS.2021.3104971.

[4] D. Diefenbach, V. Lopez, K. Singh, and P. Maret, "Core techniques of question answering systems over knowledge bases: A survey," *Knowl. Inf. Syst.*, vol. 55, no. 3, pp. 529–569, Jun. 2018.

[5] K. Tolias and S. P. Chatzis, "t-exponential memory networks for question-answering machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 8, pp. 2463–2477, Aug. 2019.

[6] J. Yang, Y. Zhang, and F. Dong, "Neural word segmentation with rich pretraining," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 839–849.

[7] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. NAACL*, 2016, pp. 260–270.

[8] J. Xu, J. Wen, X. Sun, and Q. Su, "A discourse-level named entity recognition and relation extraction dataset for Chinese literature text," 2017, *arXiv:1711.07010*. [Online]. Available: http://arxiv.org/abs/1711.07010

[9] H. He and X. Sun, "F-score driven max margin neural network for named entity recognition in Chinese social media," 2016, *arXiv:1611.04234*. [Online]. Available: http://arxiv.org/abs/1611.04234

[10] Q.-Q. Zhang, M.-D. Chen, and L.-Z. Liu, "An effective gated recurrent unit network model for Chinese relation extraction," *DEStech Trans. Comput. Sci. Eng.*, Mar. 2018, pp. 262–267.

[11] S. Zhao, Z. Cai, H. Chen, Y. Wang, F. Liu, and A. Liu, "Adversarial training based lattice LSTM for Chinese clinical named entity recognition," *J. Biomed. Informat.*, vol. 99, Nov. 2019, Art. no. 103290.

[12] S. Zhao, M. Hu, Z. Cai, H. Chen, and F. Liu, "Dynamic modeling cross- and self-lattice attention network for Chinese NER," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 16, pp. 14515–14523.

[13] Y. Zhang and J. Yang, "Chinese NER using lattice LSTM," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Jul. 2018, pp. 1554–1564.

[14] W. Liu, T. Xu, Q. Xu, J. Song, and Y. Zu, "An encoding strategy based word-character LSTM for Chinese NER," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 2379–2389.

[15] Z. Li, N. Ding, Z. Liu, H. Zheng, and Y. Shen, "Chinese relation extraction with multi-grained information and external linguistic knowledge," in *Proc. ACL*, 2019, pp. 4377–4386.

[16] J. Yang, Y. Zhang, and S. Liang, "Subword encoding in lattice LSTM for Chinese word segmentation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 2720–2725.

[17] P. Gao *et al.*, "Dynamic fusion with intra-and inter-modality attention flow for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6639–6648.

[18] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proc. CVPR*, Jun. 2019, pp. 6281–6290.

[19] R. Ma, M. Peng, Q. Zhang, Z. Wei, and X.-J. Huang, "Simplify the usage of lexicon in Chinese NER," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5951–5960.

[20] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Neural lattice-to-sequence models for uncertain inputs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1380–1389.

[21] J. Su, Z. Tan, D. Xiong, R. Ji, X. Shi, and Y. Liu, "Lattice-based recurrent neural network encoders for neural machine translation," 2016, *arXiv:1609.07730*. [Online]. Available: http://arxiv.org/abs/1609.07730

[22] Y. Tian, Y. Song, F. Xia, T. Zhang, and Y. Wang, "Improving Chinese word segmentation with wordhood memory networks," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 8274–8285.

[23] Y. Tian *et al.*, "Joint Chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 8286–8296.

[24] T. Gui, R. Ma, Q. Zhang, L. Zhao, Y.-G. Jiang, and X. Huang, "CNN-based Chinese NER with lexicon rethinking," in *Proc. 28th Int. Joint Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, Aug. 2019, pp. 4982–4988.

[25] T. Gui *et al.*, "A lexicon-based graph neural network for Chinese NER," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1039–1049.

[26] D. Sui, Y. Chen, K. Liu, J. Zhao, and S. Liu, "Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3821–3831.

[27] H. Yan, B. Deng, X. Li, and X. Qiu, "TENER: Adapting transformer encoder for named entity recognition," 2019, *arXiv:1911.04474*. [Online]. Available: http://arxiv.org/abs/1911.04474

[28] X. Li, H. Yan, X. Qiu, and X. Huang, "FLAT: Chinese NER using flat-lattice transformer," 2020, *arXiv:2004.11795*. [Online]. Available: http://arxiv.org/abs/2004.11795

[29] S. Antol *et al.*, "VQA: Visual question answering," in *Proc. ICCV*, Dec. 2015, pp. 2425–2433.

[30] Y. Yu, S. Tang, K. Aizawa, and A. Aizawa, "Category-based deep CCA for fine-grained venue discovery from multimodal data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1250–1258, Apr. 2019.

[31] D.-K. Nguyen and T. Okatani, "Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6087–6096.

[32] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. 25th Int. Conf. Comput. Linguistics (COLING)*, Aug. 2014, pp. 2335–2344.

[33] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2019, pp. 6558–6569.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.

[35] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450

[36] N. Peng and M. Dredze, "Named entity recognition for Chinese social media with jointly trained embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 548–554.

[37] G.-A. Levow, "The third international Chinese language processing bakeoff: Word segmentation and named entity recognition," in *Proc. 5th SIGHAN Workshop Chin. Lang. Process.*, 2006, pp. 108–117.

[38] T. Emerson, "The second international Chinese word segmentation bakeoff," in *Proc. 4th SIGHAN Workshop Chin. Lang. Process.*, 2005, pp. 2335–2344.

[39] X. Qiu, H. Pei, H. Yan, and X. Huang, "A concise model for multi-criteria Chinese word segmentation with transformer encoder," 2019, *arXiv:1906.12035*. [Online]. Available: http://arxiv.org/abs/1906.12035

[40] Y. Cui *et al.*, "Pre-training with whole word masking for Chinese BERT," 2019, *arXiv:1906.08101*. [Online]. Available: http://arxiv.org/abs/1906.08101

[41] Y. Song, S. Shi, J. Li, and H. Zhang, "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2018, pp. 175–180.

[42] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1753–1762.

[43] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 2124–2133.

[44] D. Zhang and D. Wang, "Relation classification via recurrent neural network," 2015, *arXiv:1508.01006*. [Online]. Available: http://arxiv.org/abs/1508.01006

[45] J. Gong, X. Chen, T. Gui, and X. Qiu, "Switch-LSTMs for multi-criteria Chinese word segmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 6457–6464.

[46] X. Chen, Z. Shi, X. Qiu, and X.-J. Huang, "Adversarial multi-criteria learning for Chinese word segmentation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Apr. 2017, pp. 1193–1203.