IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 28, 2020, accepted December 11, 2020, date of publication December 21, 2020,
date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3046253

A Residual BiLSTM Model for Named Entity Recognition

GANG YANG AND HONGZHE XU
School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China
Corresponding author: Hongzhe Xu (xuhz@xjtu.edu.cn)

ABSTRACT As one of the most powerful neural networks, Long Short-Term Memory (LSTM) is widely used in natural language processing (NLP) tasks. Meanwhile, the BiLSTM-CRF model is one of the most popular models for named entity recognition (NER), and many state-of-the-art models for NER are based on it. In this paper, we propose a new residual BiLSTM model and perform it with a conditional random field (CRF) layer together on NER tasks. Based on the most popular BiLSTM-CRF model, we replace the BiLSTM with our residual BiLSTM blocks to encode words or characters. We evaluate our model on Chinese and English datasets. We utilize both word2vec and BERT to generate word or character vectors. Furthermore, we conduct experiments to compare the performance of NER by using different structures of residual blocks. The experimental results show that our model can improve the performance of both Chinese and English NER effectively without introducing any external knowledge.

INDEX TERMS Natural language processing, named entity recognition, residual bi-directional LSTM.

I. INTRODUCTION

As a fundamental task of natural language processing tasks, named entity recognition (NER) aims to identify the named entities from unlabeled sentences or texts. Named entities are a series of special semantic types such as person (PER), organization (ORG) and location (LOC), etc. Thus, NER is a typical classification task that trains a model with texts in which named entities have been labeled rightly, and then predicts the named entities in other unlabeled texts. NER has received much attention for it will impact the performance of other downstream NLP tasks, such as relation extraction [1], entity linking [2], etc.

In recent years, deep learning technologies have been widely used in a variety of NLP and computer vision (CV) tasks. Compared with convolutional neural network (CNN), recurrent neural network (RNN) is more used for NLP tasks because it can capture the semantic features about a long sequence. The most popular RNN model is Long Short-Term Memory (LSTM) [3] that has achieved success in many NLP tasks [4], [5]. For NER, the state-of-the-art models are usually based on BiLSTM-CRF [6] which uses BiLSTM to extract the features of input sentences and connect them to a conditional random field (CRF) layer to jointly predict target labels. Among these models, many of them introduced external knowledge to the BiLSTM-CRF model, such as radical features [7], character features [8], segmentation features [9], sentence features [10], etc. Some other models employed multi-task learning to perform NER tasks with other related tasks jointly [11], [12]. However, almost all above works are the combination of existing models or methods. A few researchers [13]–[15] have attempted to improve the structure of BiLSTM-CRF.

Meanwhile, in the area of CV, residual networks have achieved state-of-the-art accuracy on image recognition and some other related tasks. For instance, ResNets [16] use identity mappings as the skip connections in layers to train a very deep network with over 100 layers. DenseNets [17] connect each layer to its all subsequent layers to build a deep residual network. Compared with a convolution kernel as the basic unit in CNNs, the LSTM kernel is more complex. Hence, it is a challenge to design a residual network based on LSTMs. Inspired by residual CNNs, we introduce a new type of residual block to BiLSTM to build a residual BiLSTM model for NER.

In this paper, we propose a new residual BiLSTM model and connect it to a CRF layer to perform NER tasks. To demonstrate the effectiveness of our model, we conduct experiments on both Chinese and English NER datasets with fixed vectors generated by word2vec [18] and GloVe [19] as inputs respectively. Furthermore, we use BERT [20] the generate higher quality input vectors for both Chinese and

The associate editor coordinating the review of this manuscript and approving it for publication was Constantinos Marios Angelopoulos.

English NER to demonstrate our model can also benefit from BERT. We choose BiLSTM-CRF, stack BiLSTM-CRF and other state-of-the-art models that introduce external knowledge or multi-task learning approaches as our baseline models. Furthermore, we conduct experiments to investigate the impacts of residual block structures and the number of layers on the performance. The experimental results show that our proposed residual BiLSTM model can improve the performance of both Chinese and English NER effectively.

The contributions of this paper can be summarized as follows:

- We propose a new residual BiLSTM model which introduces a new type of residual block to improve the capability of feature extraction of BiLSTM.
- We apply our model to NER tasks on English and Chinese datasets without introducing any external knowledge. The experimental results demonstrate the effectiveness of our model.
- As a feature extractor, our model has the potential to be applied to other NLP tasks.

The structure of our paper includes four main parts. In the first part, we introduce the existing state-of-the-art models for NER and some related works. In the second part, we elaborate the motivation and the architecture of our model. In the third part, we perform our model on a variety of English and Chinese datasets to evaluate our model. In the fourth part, we conduct ablation study to investigate the impact of each component in the residual block and the number of layers.

II. RELATED WORK

A. NAMED ENTITY RECOGNITION

Recurrent neural network such as LSTM has shown its advantages in a variety of NLP tasks. [6] proposed the BiLSTM-CRF model that is widely used for NER in various languages. Most of the state-of-the-art models for NER are based on the BiLSTM-CRF model. There are several approaches to improve the performance. The first approach is to introduce external knowledge or some other existing models to the BiLSTM-CRF model, such as character, segmentation, context, etc. Both Chinese and English characters contain semantic information that can contribute to NER tasks. For English NER, [8] used BiLSTM to encode characters in OOV words. [21] and [22] combined LSTM with CNN which is used to encode English characters. For Chinese NER, [7] introduced radical features as the input of a BiLSTM-CRF model instead of Chinese words. [23] proposed a lattice LSTM model which utilized both characters and words as the inputs. In this model, it utilized the results of segmentation by using a lexicon as the extra input for LSTM. [24] can be treated as an improved version of lattice LSTM model that integrated the results of segmentation into characters as the final input of LSTM. [25] combined CNNs and self-attention mechanism with BiLSTM-CRF together which used global self-attention layer to capture the information form characters and sentence contexts. Besides character features, segmentation features [9] and sentence features [10] can also be utilized as external knowledge. The second approach is to adopt a multi-task learning strategy to train the NER task and other related tasks together. [11] trained the NER task with Chinese word segmentation task jointly. [12] introduced adversarial transfer learning framework and self-attention mechanism to learn NER tagging and Chinese word segmentation jointly. [26] incorporated coreferential relation to enrich CNN-BiLSTM-CRF. The third approach is to improve the representations of inputs. Language models such as BERT [20] and ELMo [27] achieved state-of-the-art results in a variety of NLP tasks. They can generate dynamic vectors according to different contexts rather than fixed vectors.

B. RESIDUAL NEURAL NETWORKS

[28] proposed Highway Network that first trained very deep end-to-end networks. ResNets [16] improved the Highway Network by adding identity mappings as the skip connections in layers. DenseNets [17] made shortcut connections between each layer and its subsequent layer to build a deep residual network. Furthermore, [29] analyzed the impacts of various usages of activation. In addition to the above works based on CNNs, there are also several works that built residual networks based on LSTMs. [14] proposed stack residual LSTM networks to generate paraphrase. [15] proposed residual LSTMs for distant speech recognition. A similar work to ours is [13] which employed stack residual LSTMs for NER.

III. OUR MODEL

In this section, we first introduce the motivation and the difference between the residual structure we proposed and that of ResNets. Then we elaborate the architecture of our model in three sections. As shown in Figure 3, we take a 3-layer residual BiLSTM model as an example to illustrate the residual structure. Note that the number of layers can be changed. Figure 3 shows the overall architecture of 3-layers residual BiLSTMs with a CRF layer. The whole model consists of three main parts. The bottom part is an input layer and the top part is a CRF layer, which is similar to most of the models based on BiLSTM-CRF. Our innovation is the residual BiLSTM blocks in the middle part.

A. COMPARISONS OF STRUCTURES BETWEEN RESNETS AND OUR MODEL

In this section, we compare the structure of our model with the model in [13] that uses the same structure of ResNets. Then we analyze the impact of the residual structure on LSTMs and show the motivation of our model. From the results of [13]–[15] we can see that applying the same residual structure of ResNets to BiLSTM is not as effective as CNN. The reason is that residual LSTMs are not only deep networks, but each LSTM layer contains the information of long-term dependencies, which is the main difference from residual CNNs. Figure 1 and Figure 2 show two different structures of residual LSTMs respectively. The structure of

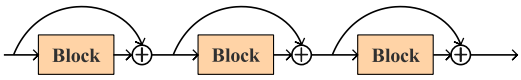

FIGURE 1. Residual LSTMs with the same structure of ResNets.

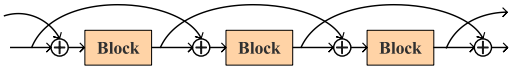

FIGURE 2. Residual LSTMs with the structure we proposed.

the network in Figure 1 is the same with that of ResNets, and Figure 2 shows the structure we proposed. Given a loss function J of the network in Figure 1:

$$J = \mathcal{F}(\mathbf{h}_t^n + \mathbf{h}_t^{n-1} + \ldots + \mathbf{h}_t^1) \tag{1}$$

We take the first LSTM layer as an example, the derivative of J with respect to $\mathbf{c}_t^1$ can be written as follows:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{c}_t^1} &= \mathcal{F}'\left(\prod_{i=0}^{n-2}\frac{\partial \mathbf{h}_t^{n-i}}{\partial \mathbf{h}_t^{n-i-1}}\right)\frac{\partial \mathbf{h}_t^1}{\partial \mathbf{c}_t^1} \\ &\quad + \mathcal{F}'\left(\prod_{i=1}^{n-2}\frac{\partial \mathbf{h}_t^{n-i}}{\partial \mathbf{h}_t^{n-i-1}}\right)\frac{\partial \mathbf{h}_t^1}{\partial \mathbf{c}_t^1} \\ &\quad + \ldots + \mathcal{F}'\frac{\partial \mathbf{h}_t^2}{\partial \mathbf{h}_t^1}\frac{\partial \mathbf{h}_t^1}{\partial \mathbf{c}_t^1} + \mathcal{F}'\frac{\partial \mathbf{h}_t^1}{\partial \mathbf{c}_t^1}\end{aligned} \tag{2}$$

In (2) we can see that $\mathcal{F}'\frac{\partial \mathbf{h}_1^n}{\partial \mathbf{c}_t^1}$ which have fewer product terms than $\mathcal{F}'\left(\prod_{i=0}^{n-2}\frac{\partial \mathbf{h}_t^{n-i}}{\partial \mathbf{h}_t^{n-i-1}}\right)\frac{\partial \mathbf{h}_t^1}{\partial \mathbf{c}_t^1}$, will yield bigger value than the items with more product terms. Thus, the value of (2) mainly depends on the last several items. As we know, $\mathcal{F}'$ contains the addition of direct outputs from each layer. If we use the result of (2) to update the weight $\mathbf{c}_t^1$ in the process of back propagation, the weight $\mathbf{c}_t^1$ will contain much direct information of each layers. As a result, the all long-term dependencies of each layer are confused and distributed in each layer. However, the cell state of each LSTM layer should be relatively independent from other layers, which allow the residual LSTMs to learn more information. Thus, the motivation of our model is to make the long-term dependencies of each layer more different to improve the capability of residual LSTMs. We adopt a ''local residual'' strategy to build the structure. We suppose that two adjacent LSTM layers have relatively strong correlation, and only keep the shortcut connections between them, which is shown in Figure 2. The loss function $\tilde{J}$ and the derivative of $\tilde{J}$ of our model can be written as follows:

$$\tilde{J} = \mathcal{F}(\mathbf{h}_t^n + \mathbf{h}_t^{n-1}) \tag{3}$$

$$\begin{aligned}\frac{\partial \tilde{J}}{\partial \mathbf{c}_t^1} &= \mathcal{F}'\left(\prod_{i=0}^{n-2}\frac{\partial \mathbf{h}_t^{n-i}}{\partial \mathbf{h}_t^{n-i-1}}\right)\frac{\partial \mathbf{h}_t^1}{\partial \mathbf{c}_t^1} \\ &\quad + \mathcal{F}'\left(\prod_{i=1}^{n-2}\frac{\partial \mathbf{h}_t^{n-i}}{\partial \mathbf{h}_t^{n-i-1}}\right)\frac{\partial \mathbf{h}_t^1}{\partial \mathbf{c}_t^1}\end{aligned} \tag{4}$$

We can observe that our model is simpler and mitigate the direct influence of the outputs of each layers on $\mathbf{c}_t^1$.

B. INPUT LAYER

In this section we illustrate the approach of encoding the input for our model. We denote an input sentence as $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, where x_i represents the i-th token in the sentence. A token represents an English word or a Chinese character in our model. In the look-up layer, we map each token to a vector as inputs of the BiLSTM. Word2vec, GloVe and BERT are all available and adopted to generate input vectors. Specifically, BERT can generate dynamic vectors of English word and Chinese characters. Word2vec can generate fixed vectors of Chinese characters. GloVe can generate fixed vectors for English words. On account of that many English named entities are out of vocabulary words, we use the same method proposed by [8] to help generating fixed vectors, which uses BiLSTM to encode each character in a word. Thus, the final fixed vector of an English word is the concatenation of the word vector and each character vectors. Note that we use only independent tokens as inputs and do not introduce any external knowledge.

C. RESIDUAL BiLSTM BLOCKS

In this section we illustrate the structure of the residual BiLSTM blocks we proposed. As shown in Figure 3, a residual BiLSTM block consists of a BiLSTM layer, a shortcut connection and four additional layers which are fully-connected layer, layer normalization [30], ReLU [31] and dropout [32] which are used to prevent overfitting and the vanishing gradient problem. Inspired by ResNets and DenseNets, we design a type of BiLSTM-based residual block which refers to an order of ''BN-ReLU-Weight'' recommend by [29]. We take the l-th blocks as an example to illustrate the structure of residual BiLSTM blocks.

In the l-th block, we denote the output of $(l-1)$-th block as $\mathbf{r}_t^{l-1}$ for position t. Then the output of the dropout layer can be written as follows:

$$\mathbf{d}_t^l = \mathcal{G}^l(\mathbf{W}_{fc} \cdot \mathbf{r}_t^{l-1}) \tag{5}$$

where $\mathbf{W}_{fc}^l$ denotes the weight matrix of the fully-connected layer, and $\mathcal{G}^l$ denotes the composite function of layer normalization, ReLU and dropout. Then the vector $\mathbf{d}_t^l$ is used as the input to the BiLSTM in this block. The basic LSTM function can be written as follows:

$$\begin{aligned}\begin{bmatrix}\tilde{\mathbf{c}}_t^l \\ \mathbf{o}_t^l \\ \mathbf{i}_t^l \\ \mathbf{f}_t^l\end{bmatrix} &= \begin{bmatrix}\tanh \\ \sigma \\ \sigma \\ \sigma\end{bmatrix}\left(\mathbf{W}_p^l\begin{bmatrix}\mathbf{d}_t^l \\ \tilde{\mathbf{h}}_{t-1}^l\end{bmatrix} + \mathbf{b}_p^l\right) \\ \mathbf{c}_t^l &= \mathbf{i}_t^l \odot \tilde{\mathbf{c}}_t^l + \mathbf{f}_t^l \odot \mathbf{c}_{t-1}^l \\ \tilde{\mathbf{h}}_t^l &= \mathbf{o}_t^l \odot \tanh(\mathbf{c}_t^l)\end{aligned} \tag{6}$$

where $\mathbf{c}_t^l, \mathbf{i}_t^l, \mathbf{o}_t^l, \mathbf{f}_t^l$ denote cell state, input gate, output gate and forget gate respectively. $\mathbf{W}_p^l, \mathbf{b}_p^l, \sigma, \odot$ denote weight matrices, bias matrices, sigmoid function and element-wise product

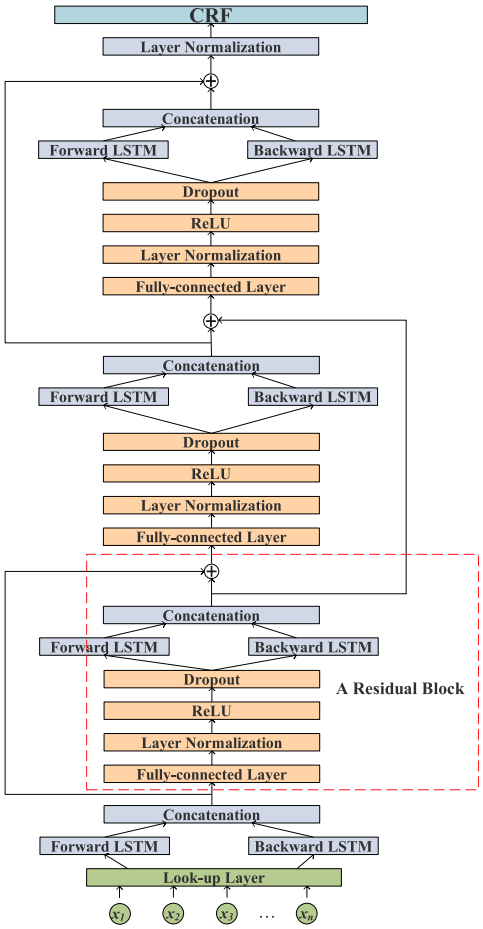

FIGURE 3. The architecture of 3-layer residual LSTMs with a CRF layer.

respectively. We denote the hidden state of the forward LSTM as $\overrightarrow{\mathbf{h}_t^l}$ and that of the backward LSTM as $\overleftarrow{\mathbf{h}_t^l}$. Then we concatenate the two hidden states and get the vector $\mathbf{h}_t^l = [\overrightarrow{\mathbf{h}_t^l}; \overleftarrow{\mathbf{h}_t^l}]$. Then the final output of the l-th residual block can be written as follows:

$$\mathbf{r}_t^l = \mathbf{h}_t^{l-1} + \mathbf{h}_t^l \tag{7}$$

Thus, we introduce a new type of identity shortcut connection to BiLSTMs to build a residual BiLSTM model. In order to illustrate the structure of the residual BiLSTM blocks more clearly, we re-written the output of l-th residual block as follows:

$$\mathbf{r}_t^l = \mathbf{h}_t^{l-1} + \mathcal{H}^l(\mathbf{r}_t^{l-1}) \tag{8}$$

where $\mathcal{H}^l$ is the composite function of all operations in the l-th residual block.

D. CRF LAYER

The CRF layer is usually used as the top layer in each model for NER. Compared with LSTMs that predict output labels independently, CRF can capture the dependency information across the output labels. For example, a label B-PER cannot follow B-PER. As BiLSTM-CRF, we use a CRF layer to predict output labels with residual BiLSTMs together. We denote an output label sequence as $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$, the score of the sequence can be written as follows:

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{n} \mathbf{A}_{y_i, y_{i+1}} + \sum_{i=1}^{n} \mathbf{P}_{i, y_i} \tag{9}$$

where $\mathbf{A}$ denotes the transition score matrix and $\mathbf{P}$ denotes a score matrix of the probabilities of labels predicted by residual BiLSTMs. Thus, the probability for the sequence $\mathbf{y}$ is:

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X},\mathbf{y})}}{\sum_{\widetilde{\mathbf{y}} \in \mathbf{Y_X}} e^{s(\mathbf{X},\widetilde{\mathbf{y}})}} \tag{10}$$

where $\mathbf{Y_X}$ denotes all possible label sequences. We use Viterbi algorithm to calculate the highest score label sequence as the result of prediction, which can be written as follow:

$$\mathbf{y}* = \arg\max_{\widetilde{\mathbf{y}} \in \mathbf{Y_X}} s(\mathbf{X}, \widetilde{\mathbf{y}}) \tag{11}$$

TABLE 1. Statistics of the datasets.

Dataset	Train	Dev	Test	Language
CoNLL-2003	15.0k	3.5k	3.7k	English
OntoNotes 5.0	81.8k	11.0k	11.2k	English
MSRA	46.4k	N/A	4.4k	Chinese
OntoNotes 4.0	15.7k	4.3k	4.3k	Chinese
Weibo	1.4k	0.27k	0.27k	Chinese

IV. EXPERIMENT RESULTS

A. DATASETS AND EXPERIMENTAL SETTINGS

We use four most widely used datasets which are CoNLL-2003 [33], MSRA [34], Weibo [11], OntoNotes 4.0 [35] and OntoNotes 5.0 [36] to evaluate our model on English and Chinese NER tasks respectively. The statistics of sentences of the datasets is shown in Table 1. We apply the schema of BIOES (B-begin, I-inside, O-outside, E-end, S-single) to for all NER datasets as baselines did. For example, the entity "Kurdistan Democratic Party" with 3 words is labeled as "B-ORG I-ORG E-ORG", where "ORG" denotes the entity type as organization. The entity "Ramallah" with single word is labeled as "S-LOC", where "LOC" denotes the entity type as location. Our model has nearly the same type of hyper-parameters as that of BiLSTM-CRF, which is much simpler than previous sophisticated models.

In the experiments, we adopt pre-trained English word vectors published by GloVe [19] and Chinese character vectors published by [37] as the fixed input for all datasets. Furthermore, we utilize BERT as the dynamic input for CoNLL-2003 and MSRA to demonstrate the robustness of our mode. The hyper-parameters of fixed input and BERT are shown in Table 2 and 3 respectively. We use Adam optimizer [38] with a gradient clipping of 5.0. Compared the hyper-parameters in Table 3, we increase the batch size and LSTM hidden size in Table 2, because our model has less parameters than BERT that we can increase these parameters to accelerate the training speed.

TABLE 2. Hyper-parameters of NER with fixed input vectors.

Parameter	Value
Chinese character vectors size	300
English word vectors size	300
English character vectors size	100
batch size	64
residual LSTM layer size	4
residual LSTM hidden size	300
fully-connected layer hidden size	600
dropout rate	0.5
learning rate	0.001

TABLE 3. Hyper-parameters of NER with BERT.

Parameter	Value
max sequence length of BERT	128
vector size of BERT	768
batch size	16
residual LSTM layer size	4
residual LSTM hidden size	128
fully-connected layer hidden size	256
dropout rate	0.5
learning rate	0.001

B. RESULTS ON ENGLISH NER DATASETS

In this section, we perform our model on English NER Datasets ConLL-2003 and OntoNotes 5.0. We take the same approach proposed by [8] to generate English input vectors for English NER, where the inputs of English NER are composed of pre-trained word vectors from GloVe[1] and character vectors learned by a BiLSTM network. The results are shown in Table 4 and Table 5. Our model achieves a F1-score of 92.22% and 89.65% on CoNLL-2003 and OntoNotes 5.0 respectively, which outperform the baselines on the both datasets. Our model also outperforms the residual LSTM model in [13] significantly. Meanwhile, we can observe that stacked BiLSTM model performs worse that [13]. It demonstrates that shortcut connection can improve the performance of stacked BiLSTM, and the residual structure in our model is more effective and reasonable than [13] which uses the same structure of ResNets.

Since most NLP task can benefit from BERT, we also adopt BERT [2] to generate dynamic input vector for our model on the ConLL-2003 dataset. We use the official BERT tools[3] offered by Google to program which adopts AdamW [39] algorithm for optimization. On account of that our model is more complex to fine tune with BERT, we use the method proposed by [40] which contains two steps to fine tune a complex model with BERT. Table 6 shows the F1-scores on ConLL-2003. The baselines also adopt BERT or ELMo as the input. We can see that our model work with BERT more effectively than baselines, which again shows the effectiveness and robustness of our model.

[1]http://nlp.stanford.edu/data/glove.6B.zip

[2]https://storage.googleapis.com/bert_models/2018_10_18/cased_L-12_H-768_A-12.zip

[3]https://github.com/google-research/bert

TABLE 4. Results of the CoNLL-2003 dataset with fixed input vectors.

Model	F1
Chiu and Nichols [22]	91.95
Tran et al. [13]	91.64
Yang et al. [10]	91.62
Xin et al. [49]	91.81
Zhang et al. [50]	91.57
Liu et al. [44]	91.54
Qian et al. [51]	91.74
Dai et al. [26]	91.96
Our model	**92.22**

TABLE 5. Results of the OntoNotes 5.0 dataset with fixed input vectors.

Model	F1
Strubell et al. [52]	86.99
Tran et al. [13]	87.22
Li et al. [53]	87.21
Ghaddar et al. [54]	87.95
Clark et al. [55]	88.81
Chen et al. [56]	87.67
Dai et al. [26]	88.95
Our model	**89.68**

TABLE 6. Results of the CoNLL-2003 dataset with BERT.

Model	F1
Devlin et al. [20] (BERT)	92.80
Akbik et al. [43]	93.18
Liu et al. [44]	93.23
Dai et al. [26]	93.29
Our model + BERT	**93.31**

C. RESULTS ON CHINESE NER DATASETS

In this section, we perform our model on Chinese NER Datasets MSRA, Weibo and OntoNotes 4.0. We use pre-trained Chinese character embeddings proposed by [37] for all datasets. The results of the 3 datasets are shown in Table 7 to Table 9 respectively. Our model achieves a F1-score of 92.17% on MSRA, which gains 1.67% improvement in F1-score compared with BiLSTM-CRF. And it also outperforms baselines on OntoNotes 4.0. For the Weibo dataset, the F1-score of our model is slightly worse than [25]. The reason is that the Weibo dataset is a relatively small dataset that our model use only character as inputs, but [25] utilizes sentences as external information. Nonetheless, the performance of our model is still better than most baselines. Meanwhile, we can observe that the F1-scores of the model in [13] on the 3 Chinese datasets are all lower than our model, which is consistent with the results of English NER.

For Chinese NER, we also adopt BERT and BERT-based language model to generate dynamic input vectors to evaluate our model on the MSRA dataset. We use Chinese BERT-Base,[4] BERT-wwm[5] [41] and ERNIE 1.0 Base[6] [42] to

[4]https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

[5]https://pan.iflytek.com/#/link/A2483AD206EF85FD91569B498A3C3879

[6]https://ernie.bj.bcebos.com/ERNIE_stable.tgz

TABLE 7. Results of the MSRA dataset with fixed input vectors.

Model	F1
Levow, [34]	91.18
Lample et al. [8] (BiLSTM+CRF)	90.55
Tran et al. [13]	91.65
Dong et al. [7]	90.95
Cao et al. [12]	90.64
Yang et al. [45]	91.67
Our model	**92.17**

TABLE 8. Results of the OntoNotes 4.0 dataset with fixed input vectors.

Model	F1
Tran et al. [13]	72.33
Yan et al. [46]	72.43
Zhang and Yang [23]	73.88
Dai et al. [25]	73.64
Liu et al. [24]	74.43
Our model	**74.54**

TABLE 9. Results of the Weibo dataset with fixed input vectors.

Model	F1
Peng and Dredze [11]	56.05
Peng and Dredze [9]	58.99
He and Sun [47]	54.82
He and Sun [48]	58.23
Tran et al. [13]	57.64
Cao et al. [12]	58.70
Zhang and Yang [23]	58.79
Zhu et al. [25]	59.31
Our model	**58.91**

generate Chinese character vectors. We take the same tool and optimization method in the previous section, and the fine-tune learning rates are $3e^{-5}$, $4e^{-5}$ and $5e^{-5}$ for BERT-Base, BERT-wwm and ERNIE respectively. The results in shown in Table 10. We can see that the results of using BERT or BERT-based models are much better than the models [23]–[25] using external knowledge, which again demonstrates that our model can benefit from BERT and achieves better performance than baselines. It also shows that a good pre-trained model can make an improvement on NER tasks significantly.

TABLE 10. Results of the MSRA dataset with BERT and ERNIE.

Model	F1
Zhang and Yang [23]	93.18
Liu et al. [24]	93.74
Zhu et al. [25]	92.97
Devlin et al. [20] (BERT)	95.30
Cui et al. [41] (BERT-wwm)	95.40
Zhang et al [42] (ERNIE)	95.40
Our model + BERT	95.41
Our model + BERT-wwm	**95.74**
Our model + ERNIE	95.66

V. ANALYSIS AND DISCUSSION

A. IMPACT OF RESIDUAL BLOCK STRUCTURE

In this section, we perform Chinese NER on the MSRA dataset with several different types of residual block structures. We use the same hyper-parameters shown in Table 2. Figure 4 shows four different representative structures of the

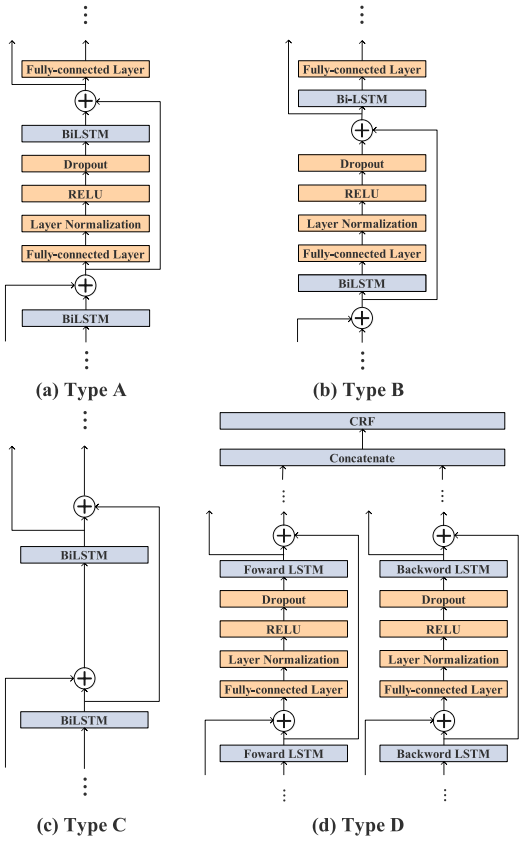

FIGURE 4. Four different structures of residual blocks.

residual block. Type A builds shortcut connections in a way that is similar to ResNets, where the element-wise addition is before the identity mapping. Note that Type A has the same structure with the model in [13]. Like Type A, Type B just moves the element-wise addition before the BiLSTM layer. Type C can be treated as a simplified version of our model. It removes fully-connected layer, layer normalization, ReLU and dropout from the residual block. Type D builds residual LSTMs for forward LSTMs and backward LSTMs separately. Furthermore, we conduct further experiments by removing the fully-connected layer and layer normalization. We choose Chinese NER for that the model for English NER needs an extra BiLSTM network to encode characters, which may influence the results of the experiment.

From Table 11 we can observe that the performance of Type C is better than Type A and Type B, which demonstrates that it is more reasonable and effective to build a residual BiLSTM model with the structure we proposed. Meanwhile, it also shows that the order of "BN-ReLU-Weight" proposed by [29] for the block also works in residual LSTMs.

B. IMPACT OF NUMBER OF RESIDUAL BiLSTM LAYERS

In this section, we repeat the Chinese NER tasks on the MSRA dataset by changing the number of residual BiLSTM blocks. Meanwhile, we choose the traditional stacked BiLSTM-CRF models as the baseline model which is

TABLE 11. Results of the MSRA dataset with different structures of residual blocks.

Model	F1
Type A	90.82
Type B	90.25
Type C	91.24
Type D	90.66
Our model	**92.22**

TABLE 12. Results of the MSRA dataset with different number of BiLSTM layers.

Model	Layer	F1
stacked BiLSTM-CRF	2	90.55
	3	90.69
	4	90.53
residual BiLSTM-CRF	2	91.38
	3	91.85
	4	**92.22**
	5	91.80
	6	91.62

without shortcut connections and uses the output of previous LSTM as the input of next LSTM directly. The results are shown in Table 12. We can observe that increasing the number of stacked BiLSTM-CRF contributes slightly to the performance compared to the BiLSTM-CRF model. For our model, the highest F1-score is achieved by the 4-layer residual BiLSTM-CRF. The F1-score begins to drop when the number of layers is more than 4, which is consistent with the results of [13]. The reason might be that the structure of a LSTM kernel is more complex that has much more parameters that a CNN kernel. It is easier to overfit for a deep LSTM network with multiple layers. Hence, it is relatively difficult to train it effectively and to learn high-quality long-term dependencies information for each LSTM layer. Nevertheless, our model with 4 layers still outperforms stacked BiLSTM-CRF models significantly.

TABLE 13. Results of ablation study on the MSRA dataset.

Model	F1	Difference
Full model	92.22	-
w/o dense layer	92.02	-0.20
w/o layer normalization	91.58	-0.64
w/o ReLU	91.83	-0.39
w/o dropout	91.78	-0.44
w/ ELU	92.09	-0.13
w/ GRU	91.55	-0.67

C. ABLATION STUDY

In this section, we investigate the effectiveness of each layer in the residual BiLSTM blocks by the ablation study on MSRA. The results are shown in Table 13. Obviously, layer normalization makes the most contribution. Meanwhile, ReLU and dropout which are used to prevent overfitting both contribute to our model. And we introduce a dense layer to further improve the performance slightly. We conduct two extra experiments where we replace the ReLU and LSTM with ELU [57] and GRU [58] respectively, but it makes little contribution to the F1-score. In particular, GRU is not suitable for the residual BiLSTM units. Because the hidden state and cell state of GRU are the same state, which will easily break the long-term dependencies of each layer when using GRU to build a residual network. Compared to GRU, LSTM has two different states to keep hidden state and cell state respectively that LSTM is more suitable for the residual structure.

TABLE 14. Results of case study on the CoNLL-2003 dataset.

Model	Sentence 1	Sentence 2
stacked BiLSTM-CRF	[Defender Hassan Abbas]$_{PER}$ rose to ...	... at the [Melbourne Cricket Ground]$_{ORG}$ after ...
Tran et al. [13]	[Defender Hassan Abbas]$_{PER}$ rose to ...	... at the [Melbourne Cricket Ground]$_{LOC}$ after ...
Our Model	Defender [Hassan Abbas]$_{PER}$ rose to ...	... at the [Melbourne Cricket Ground]$_{LOC}$ after ...

D. CASE STUDY

In this section, we conduct a case study on the stacked BiLSTM, residual BiLSTM and our model with CRF. The number of layers is set to 3. The results are show in Table 14. We can see our model predict the entities in the two sentences correctly. By contrast, the stacked BiLSTM model does not predict either the boundary in the first sentence or the entity type in the second sentence correctly. The residual BiLSTM model in [13] only predicts the entity type correctly. The results show that our model can capture richer semantic information from texts for NER.

VI. CONCLUSION AND FUTURE WORK

We present a novel residual BiLSTM model for NER tasks. We introduce a new type of residual block based on BiLSTMs. Being different from most other state-of-the-art models that introduce external knowledge or multi-task learning, we make efforts to innovate on the structure of residual network based on BiLSTMs. We evaluate our model on Chinese and English NER datasets. The experimental results show that our model can improve the performance of both Chinese and English NER effectively without introducing external knowledge. Meanwhile, our model performs well with both fixed and dynamic inputs, which demonstrates the robustness of our model. Furthermore, we conduct experiments with several different structures of residual blocks. The results demonstrate the effectiveness of the structure of the residual block we proposed.

In the future, we will also try to combine our model with attention mechanism. For example, we can use attention layers to control the weight of each layer. And we will try to introduce external knowledge such as contextual information as the extra input to enhance our model. On the other hand, we will apply our model to other NLP tasks. For example, our model can be used to encode sentences for relation extraction and extract the features of texts for text classification instead of BiLSTM.

REFERENCES

[1] 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.*, Sep. 2015, pp. 1753–1762.

[2] N. Gupta, S. Singh, and D. Roth, "Entity linking via joint encoding of types, descriptions, and context," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Copenhagen, Denmark, Sep. 2017, pp. 2681–2690.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*. [Online]. Available: http://arxiv.org/abs/1409.3215

[5] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Oct. 2017, pp. 207–212.

[6] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: http://arxiv.org/abs/1508.01991

[7] C. Dong, J. Zhang, C. Zong, M. Hattori, and H. Di, "Character-based LSTM-CRF with radical-level features for Chinese named entity recognition," in *Proc. Int. Conf. Comput. Process. Oriental Lang.*, 2016, pp. 239–250.

[8] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 260–270.

[9] N. Peng and M. Dredze, "Improving named entity recognition for chinese social media with word segmentation representation learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2016, pp. 149–155.

[10] J. Yang, Y. Zhang, and F. Dong, "Neural reranking for named entity recognition," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process.*, Sep. 2017, pp. 784–792.

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

[12] P. Cao, Y. Chen, K. Liu, J. Zhao, and S. Liu, "Adversarial transfer learning for Chinese named entity recognition with selfattention mechanism," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Nov. 2018, pp. 182–192.

[13] Q. Tran, A. Mackinlay, and A. Jimenoyepes, "Named entity recognition with stack residual LSTM and trainable bias decoding," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, Dec. 2017, pp. 566–575.

[14] A. Prakash, S. A. Hasan, K. Lee, V. V. Datla, A. Qadir, J. Liu, and O. Farri, "Neural paraphrase generation with stacked residual LSTM networks," in *Proc. 26th Int. Conf. Comput. Linguistics*, Dec. 2016, pp. 2923–2934.

[15] J. Kim, M. El-Khamy, and J. Lee, "Residual LSTM: Design of a deep recurrent architecture for distant speech recognition," 2017, *arXiv:1701.03360*. [Online]. Available: http://arxiv.org/abs/1701.03360

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[17] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2261–2269.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2013, pp. 3111–3119.

[19] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 1532–1543.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. ACL Conf. NAACL HLT*, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[21] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2016, pp. 1064–1074.

[22] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 357–370, Dec. 2016.

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

[24] W. Liu, T. Xu, Q. Xu, J. Song, and Y. Zu, "An encoding strategy based word-character lstm for chinese ner," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Jun. 2019, pp. 2379–2389.

[25] Y. Zhu, G. Wang, and B. Karlsson, "Can-ner: Convolutional attention network for Chinese named entity recognition," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Jun. 2019, pp. 3384–3393.

[26] Z. Dai, H. Fei, and P. Li, "Coreference aware representation learning for neural named entity recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4946–4953.

[27] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. NAACL*, Jun. 2018, pp. 2227–2237.

[28] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. Conf. Neural Inf. Process. Syst.*, Dec. 2015, pp. 2377–2385.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Zürich, Switzerland, Oct. 2016, pp. 630–645.

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

[31] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Apr. 2011, pp. 315–323.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[33] E. T. K. Sang and F. D. Meulder, "Introduction to the conll-2003 shared task: language-independent named entity recognition," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, May 2003, pp. 142–147.

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

[35] R. Weischedel, S. Pradhan, L. Ramshaw, M. Palmer, N. Xue, M. Marcus, A. Taylor, C. Greenberg, E. Hovy, R. Belvin, and A. Houston, *Ontonotes Release 4.0*, document LDC2011T03. DVD. Philadelphia, Linguistic Data Consortium, 2011.

[36] R. Weischedel, S. Pradhan, L. Ramshaw, M. Palmer, N. Xue, M. Marcus, A. Taylor, C. Greenberg, E. Hovy, R. Belvin, and A. Houston, *Ontonotes Release 5.0*, document LDC2013T19. Web Download. Philadelphia: Linguistic Data Consortium, 2013.

[37] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, "Analogical reasoning on chinese morphological and semantic relations," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2018, pp. 138–143.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, May 2015, pp. 1–15.

[39] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, May 2019.

[40] R. Wang, H. Su, C. Wang, K. Ji, and J. Ding, "To tune or not to tune? How about the best of both worlds?" 2019, *arXiv:1907.05338*. [Online]. Available: http://arxiv.org/abs/1907.05338

[41] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, and G. Hu, "Pre-training with whole word masking for chinese BERT," 2019, *arXiv:1906.08101*. [Online]. Available: http://arxiv.org/abs/1906.08101

[42] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2019, pp. 1441–1451.

[43] A. Akbik, T. Bergmann, and R. Vollgraf, "Pooled contextualized embeddings for named entity recognition," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Jun. 2019, pp. 724–728.

[44] Y. Liu, F. Meng, J. Zhang, J. Xu, Y. Chen, and J. Zhou, "Gcdt: A global context enhanced deep transition architecture for sequence labeling," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Jun. 2019, pp. 2431–2441.

[45] F. Yang, J. Zhang, G. Liu, J. Zhou, C. Zhou, and H. Sun, "Deep contextualized word representations," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Jun. 2018, pp. 184–195.

[46] 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

[47] H. He and X. Sun, "F-score driven max margin neural network for named entity recognition in chinese social media," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 713–718.
[48] H. He and X. Sun, "A unified model for cross-domain and semi-supervised named entity recognition in chinese social media," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 713–718.
[49] Y. Xin, V. Mahajan, E. Hart, and J. D. Ruvini, "Learning better internal structure of words for sequence labeling," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Aug. 2018, pp. 2584–2593.
[50] Y. Zhang, Q. Liu, and L. Song, "Sentence-state lstm for text representation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2018, pp. 317–327.
[51] Y. Qian, E. Santus, Z. Jin, J. Guo, and R. Barzilay, "Graphie: A graph-based framework for information extraction," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Jun. 2019, pp. 751–761.
[52] E. Strubell, P. Verga, D. Belanger, and A. McCallum, "Fast and accurate entity recognition with iterated dilated convolution," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2017, pp. 2670–2680.
[53] P.-H. Li, R.-P. Dong, Y.-S. Wang, J.-C. Chou, and W.-Y. Ma, "Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2017, pp. 2664–2669.
[54] A. Ghaddar and P. Langlais, "Robust lexical features for improved neural network named-entity recognition," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2018, pp. 1896–1907.
[55] K. Clark, M.-T. Luong, C. Manning, and Q. Le, "Semi-supervised sequence modeling with cross-view training," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Nov. 2018, pp. 1914–1925.
[56] H. Chen, Z. Lin, G. Ding, J. Lou, Y. Zhang, and B. Karlsson, "GRN: Gated relation network to enhance convolutional neural network for named entity recognition," in *Proc. 33st AAAI Conf. Artif. Intell.*, Feb. 2019, pp. 6236–6243.
[57] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. Int. Conf. Learn. Represent.*, Nov. 2016, pp. 1–15.
[58] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

GANG YANG received the bachelor's degree in information engineering from Xi'an Jiaotong University, Xi'an, in 2011, where he is currently pursuing the Ph.D. degree in computer technology. His main research directions are natural language processing and deep learning.

HONGZHE XU received the Ph.D. degree in mechanical engineering, in 2004. From 1999 to 2012, she was a Vice Professor with the Research School of Computing. Since 2012, she has been a Professor with the Research School of Computing, Xi'an Jiaotong University, Xi'an. She is the author of ten books, more than 50 articles, and more than five inventions. Her research interests include intelligent platform in cloud environment, object-oriented big data analysis, application of data mining, and medical big data analysis.