Investigating Self-Attention Network for Chinese Word Segmentation

Leilei Gan and Yue Zhang

***Abstract*—Neural network has become the dominant method for Chinese word segmentation. Most existing models cast the task as sequence labeling, using BiLSTM-CRF for representing the input, and making output predictions. Recently, attention-based sequence models have emerged as a highly competitive alternative to LSTMs, which allow better running speed by parallelization of computation. We investigate self-attention network (SAN) for Chinese word segmentation, making comparisons between BiLSTM-CRF models. In addition, the influence of contextualized character embeddings is investigated using BERT, and a method is proposed for integrating word information into SAN segmentation. Results show that SAN gives highly competitive results compared with BiLSTMs, with BERT, and word information further improving segmentation for in-domain, and cross-domain segmentation. Our final models give the best results for 6 heterogenous domain benchmarks.**

***Index Terms*—Chinese word segmentation, contextualized word embedding, domain adaptation, self-attention network.**

I. Introduction

WORD segmentation is a necessary pre-processing step for Chinese natural language processing tasks [3], [10], [25], [30], [38]. The dominant method treats Chinese Word Segmentation (CWS) as a sequence labeling problem [31], and neural network models [2], [32], [36], [39] have achieved the state-of-the-art results. A representative model [4], [5] takes LSTM [9] as a feature extractor, and a standard CRF [13] layer is used on top of BiLSTM layers to predict the label sequences.

As an alternative representation learning model, self-attention network (SAN) [28] has been shown highly effective for a range of natural language processing tasks, such as machine translation [27], constituency parsing [12], semantic role labeling [26], and language modeling [6], [21], [42]. Compared with recurrent neural networks (RNNs) [7], SAN has advantages of capturing long-term dependencies and supporting parallel computing more easily. It has become a dominant approach for learning sequence representation in the research literature in recent years. However, its effectiveness on CWS has not been fully investigated in the literature.

We empirically investigate SAN for CWS by building a SAN-CRF word segmentor, comparing its effectiveness with a state-of-the-art BiLSTM-CRF baseline. In particular, we take the Transformer [28] architecture for building our model. The default Transformer framework performs *global attention*, where each input character learns its representation by attention over all the other characters in the input sentence. On the other hand, it has been shown that a *local attention* mechanism has its unique advantages over global attention for certain NLP tasks [43]–[45]. Observing that Chinese word segmentation can rely heavily on local information, we compare global attention and local attention in the model.

Based on the SAN-CRF segmentation model, we investigate two further questions. First, in Chinese, characters are highly polysemantic, where the same character can have different meanings in different contexts. As a result, contextualized character representations, which vary according to sentence context, can be potentially more useful for Chinese word segmentation compared with static embeddings, which most previous work uses [5], [33], [37]. SAN has also been shown to be a useful method for training contextualized word representations [6], [21]. We compare context-independent character representations [17], [18] with contextualized character representations using our SAN framework in both in-domain and cross-domain CWS evaluation.

Second, out-of-vocabulary (OOV) words, especially domain-specific noun entities, raise a challenge for cross-domain CWS. To solve this problem, domain lexicons can be used [35], [37] for cross-domain CWS tasks. We consider a novel method for integrating lexicons to the SAN framework for cross-domain CWS, using attention to integrate word information by generalizing words into POS tags, resulting in end-to-end neural type-supervised domain adaptation.

Results on three benchmarks show that SAN-CRF can achieve competitive performance compared with BiLSTM-CRF. In addition, BERT character embeddings are used for both in-domain and cross-domain evaluation. One important observation is that local attention plays a crucial role in the competitiveness of SAN segmentation model. In cross-domain evaluation, our proposed neural type-supervised method gives an averaged error reduction of 30.32% on three cross-domain datasets. Our method gives the best results on standard benchmarks including CTB, PKU, MSR,

Manuscript received September 28, 2019; revised April 8, 2020 and August 3, 2020; accepted September 9, 2020. Date of publication October 13, 2020; date of current version November 13, 2020. This work was supported by the Grant NSFC 61976180 and a Grant from Rxhui Inc. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dilek Hakkani-Tur. *(Corresponding author: Yue Zhang.)*

Leilei Gan is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310007, China (e-mail: 11921071@zju.edu.cn).

Yue Zhang is with the School of Engineering, Westlake University, Hangzhou 310024, China, and also with the Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou 310024, China (e-mail: zhangyue@westlake.edu.cn).

Digital Object Identifier 10.1109/TASLP.2020.3030487

2329-9290 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.

ZX, FR and DL. To the best of our knowledge, we are the first to investigate SAN for CWS.[1]

II. Related Work

Our work is related with three strands of existing work, including: i) methods on neural Chinese word segmentation, ii) recent investigation on self-attention network and iii) the use of contextualized word representations for improving NLP tasks.

A. Chinese Word Segmentation

There has been a long line of work on neural word segmentation. [4], [5] extract features based on character representation by using LSTM or GRU, while [46] investigate using convolution neural network as a feature extractor. [36] propose a transition-based neural model, which can make use of the word-level features. [39] train character embedding with word-based context information on auto-segmented data. [32] exploit the effectiveness of rich external resources through multi-task learning. [47] jointly train Chinese word segmentation and dependency parsing using a graph-based model. While the above work shows that CNN and LSTM are useful for CWS, we show that SAN with local attention can give strong accuracies. For cross-domain CWS, [35] propose a type-supervised domain adaptation approach for joint CWS and POS-tagging, which shows competitive results compared to token-supervised methods. [20] investigate CWS for Chinese novels, proposing a method to automatically mine noun entities for novels using a double-propagation algorithm. [37] investigate how to integrate external dictionaries into CWS models. Similar to [35] and [37], our work uses a domain lexicon. To our knowledge, we are the first to investigate contextualized character embeddings and lexicon integration for neural CWS under a SAN framework.

B. Self-Attention Network

Self-attention network [28] was first proposed for machine translation. [26] and [24] use SAN for the task of semantic role labeling, which can directly capture the relationship between two arbitrary tokens in the sequence. [24] incorporate linguistic information through multi-task learning, including dependency parsing, part-of-speech and predicate detection. [22] propose multi-dimensional attention as well as directional information, achieving the state-of-the-art results on natural language inference and sentiment analysis. [12] show that a novel encoder based on self-attention can lead to state-of-the-art results for the constituency parsing task. Along with this strand of work, we study the influence of global and local attention for CWS. We build a SAN-CRF word segmentor, which gives competitive results compared with BiLSTMs. To our knowledge, we are the first to investigate SAN for Chinese word segmentation.[2]

C. Contextualized Word Representation

Context-dependent word representations pre-trained from large-scale corpora have received much recent attention. For example, ELMo [19] is based on recurrent neural networks language models. OpenAI GPT [21] builds a left-to-right language model with a multi-layer multi-head self-attention networks, which can handle long-range dependencies better compared to recurrent networks. Different from OpenAI GPT, BERT [6] is trained from large scale raw texts using masked language model task with a deep bidirectional Transformer. [40] propose a multi-criteria method of CWS based on BERT, which uses a private projection layer to learn segmentation criteria of each dataset, and a shared projection layer to learn their common basic knowledge. However, they did not verify their method on cross-domain datasets. Our work investigates the effect of contextualized character representation on both in-domain and cross-domain CWS under a unified SAN framework.

III. Baseline

We take BiLSTM-CRF as our baseline, which has been shown giving the state-of-the-art results [5], [33]. Formally, given an input sentence with m characters $s = c_1, c_2, \ldots, c_m$, where c_i denotes the ith character, the task of character-based CWS is to assign each character c_i with a label y_i, where $y_i \in \{B, M, E, S\}$ [31]. The labels B, M, E and S represent the beginning, middle and end of a word, and single character word, respectively.

For each character c_i, its input representation is the concatenation of unigram character embedding $\mathbf{e}_{c_i}$ and bigram character embedding $\mathbf{e}_{c_i c_{i+1}}$ as follows:

$$\mathbf{x}_i^c = \mathbf{e}_{c_i} \oplus \mathbf{e}_{c_i c_{i+1}}, \tag{1}$$

where $\oplus$ represents concatenation operation.

Following [5], we feed the character representations $\mathbf{x}_i^c$ into BiLSTM layers to capture forward and backward hidden states. The final hidden representation of character c_i is the concatenation of forward hidden state $\overleftarrow{\mathbf{h}}_i$ and backward hidden state $\overrightarrow{\mathbf{h}}_i$:

$$\mathbf{h}_i = \overleftarrow{\mathbf{h}}_i \oplus \overrightarrow{\mathbf{h}}_i. \tag{2}$$

On top of the hidden representations, a CRF layer is used to consider the dependencies of adjacent labels. Formally, the probability of a label sequence $y = y_1, y_2, \ldots, y_n$ of sentence s is:

$$P(y|s) = \frac{\exp\left(\sum_{i=1}^{n}(F(y_i) + L(y_{i-1}, y_i))\right)}{\sum_{y' \in \mathbb{C}(s)} \exp\left(\sum_{i=1}^{n}(F(y'_i) + L(y'_{i-1}, y'_i))\right)}, \tag{3}$$

where $\mathbb{C}(s)$ is the set of all possible label sequences of sentence s. $F(y_i) = \mathbf{W}^{l_i}\mathbf{h}_i + b^{l_i}$ is the emission score from hidden representation $\mathbf{h}_i$ to label y_i. $\mathbf{W}^{l_i}$ and b^{l_i} are label-specific parameters. $L(y_{i-1}, y_i)$ is the transition score from y_{i-1} to y_i, which is a scalar model parameter specific to a pair of labels.

IV. Model

Fig. 1 shows our segmentor framework on an input character sequence "中国科学院院士 (Fellow of the Chinese Academy of Sciences)". The model takes character representation and positional embeddings as input. By matching the input to a word-POS lexicon, word information is investigated by using attention for each character. Multiple layers of self-attention

[1][Online]. Available: https://github.com/leileigan/SAN-CWS.

[2]There has been recent unpublished work on this subject also [41] which is done contemporarily.

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.

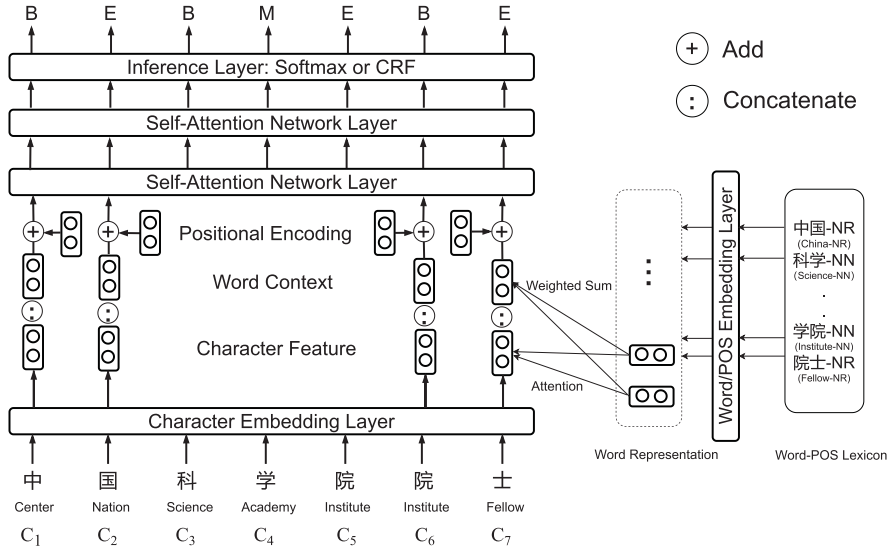


Fig. 1. Model Overview. The character embedding layer maps characters to character embeddings. Each character representation is concatenated with candidate words matched from a word-POS lexicon. Multiple layers of self-attention network are used as feature extractor. A CRF layer on top of SAN layers is used to model the dependencies of adjacent labels.

network [28] are used as feature extractor to replace BiLSTM in the baseline. Similar to the baseline, we also use a CRF layer on top of the self-attention network to model the dependencies of adjacent labels.

A. Embedding Layer

As shown in Fig. 1, the embedding layer consists of character embeddings and positional embeddings. The character representation of c_i is the concatenation of unigram character embedding $\mathbf{e}_{c_i}$ and bigram character embedding $\mathbf{e}_{c_i c_{i+1}}$:

$$\widetilde{\mathbf{x}}_i^c = \mathbf{e}_{c_i} \oplus \mathbf{e}_{c_i c_{i+1}}. \tag{4}$$

Because self-attention network does not explicitly consider sequence information, positional encoding is added to the input embeddings $\widetilde{\mathbf{x}}_i^c$ of self-attention network as follows:

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/d_{model}}) \\ \mathbf{x}_i^c &= \widetilde{\mathbf{x}}_i^c + PE, \end{aligned} \tag{5}$$

where *pos* is the position, i is the dimension, d_{model} is the dimension of output, respectively, and $+$ denotes vector addition. $\mathbf{x}_i^c$ is the final representation of the character c_i.

B. Self-Attention Network

We extend the model of [28] for the SAN segmentor, which has multiple identical layers, each being composed of a multi-head self-attention sub-layer and a position-wise fully connected feed-forward network.

Multi-head self-attention is used to exchange information directly between positions in the sequence. First, for single-head self-attention, the hidden representations of the input representation X is computed by scaled dot-product attention as follows:

$$\begin{aligned} \text{Attention}(X) &= \text{Attention}(Q, K, V) \\ &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \end{aligned} \tag{6}$$

where $Q = W_Q^T X, K = W_K^T X, V = W_V^T X$ are query, key and value vectors, respectively, d_k is the dimension of key and value vectors, and $W_Q^T,\ W_K^T,\ W_V^T$ are parameters.

Local Self-Attention In order to investigate the effect of long-term dependencies on CWS task, we propose a *local self-attention*, which only considers the surrounding positions for each character instead of all positions in the sequence. The intuition is that long-term dependencies may bring more noise than useful information for a sequence labeling task [15]. The *local self-attention* is denoted as:

$$\begin{aligned} \text{L-Attention}(X) &= \text{Attention}(Q, K, V) \\ &= \left(\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \odot W\right)V, \end{aligned} \tag{7}$$

where $\odot$ means element-wise product, W is a mask matrix to control the self-attention inner a window and its element W_{ij} is denoted as:

$$W_{ij} = \begin{cases} 1 & j - i <= WS \\ -\infty & \text{otherwise} \end{cases}. \tag{8}$$

Here WS is the window size.

Multi-Head Self-Attention is used, which linearly maps Q, K and V into multiple versions Q_i, K_i and V_i and then concatenates the outputs of different head_i as follows:

$$\text{MH}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_n)W^o, \tag{9}$$

where

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{10}$$

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.

W^o, W_i^Q, W_i^K and W_i^V are parameters.

On top of the multi-head attention sub-layer, a fully connected feed-forward network (FFN) is applied to each position. FFN is composed of two linear transformations with a ReLU activation:

$$\text{FFN}(x) = W_2\text{ReLU}(0, xW_1 + b_1) + b_2, \tag{11}$$

where W_1, W_2, b_1 and b_2 are parameters.

V. Investigating Rich Character and Word Features for SAN CWS

We incorporate rich character and word features into the SAN model. In particular, pre-trained contextualized character representation is introduced as well as a word-based neural type-supervised domain adaptation method.

A. BERT Character Representation

BERT [6] is trained from a large-scale corpora by using a deep bidirectional Transformer for masked LM tasks. Usages of BERT can be divided into feature-based and fine-tuning methods. The former fixes all model parameters and directly extracts character features from the pre-trained model, while the latter jointly fine-tunes all parameters on downstream tasks. We take the latter method, feeding the input sequence of characters into BERT and use the top layer output as character representation. Development experiments show that fine-tuning BERT embeddings give better results than the feature-based method.

Formally, the unigram character embedding of character c_i is replaced by using pre-trained BERT embedding according to the whole sentence.

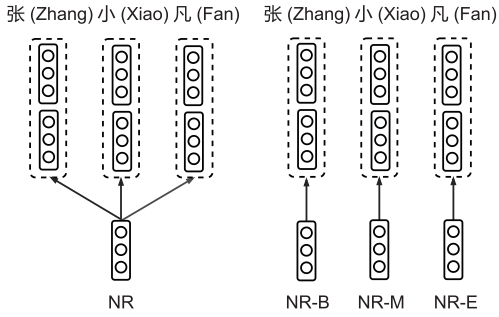
$$\mathbf{e}_{c_i} = \mathbf{e}^c_{bert}(c_i), \tag{12}$$

where $\mathbf{e}^c_{bert}$ denotes a pre-trained BERT character embedding.

B. Integrating Word-POS Lexicon for Type-Supervision

We integrate word information into SAN to handle rare words in cross-domain settings. Following the definition by [35], we describe this model in a cross-domain setting only, where C_s denotes a set of annotated source-domain sentences, and ξ_t denotes an annotated target-domain lexicon, in which each word is associated with one POS tag. The domain adaptation model is first trained on C_s, and makes use of ξ_t when performing target domain segmentation.

As shown in Fig. 1, for each character c_i in the input sentence, the set of all character subsequences that match words in the external lexicon $\mathbb{D}$ is denoted as $\mathbf{w}^i = \{w_{b_1,e_1}, w_{b_2,e_2}, \ldots, w_{b_m,e_m}\}$. Here b_k and e_k are the start and end index of the matched words in the sentence, where $e_k >= i$ and $b_k <= i$. Word embeddings should intuitively be used for encoding w_{b_k,e_k}. However, for characters forming domain specific words, there may not be readily available embeddings. POS embeddings can be used as alternative unlexicalized features of words embeddings. We introduce how to integrate POS embeddings to enrich word information from both prediction and training.



(a) integral POS embedding (P_t) (b) individual POS embedding (P_{t_b})

Fig. 2. Two methods to learn POS embeddings. In the integral POS embedding method, characters in "张小凡(Person Name)" attend to the same POS *NR*. In the individual POS embedding method, different characters attend to different POS tags with in-word positional information (B, M and E).

Prediction During testing, we additionally match character subsequences in a given input sentence to a word-POS lexicon ξ_t. For each matching subspan, we find a vector representation by first performing a lookup action to a word embedding table, and then using the corresponding POS embedding to represent the word if no word embedding is available for the subspan as follows:

$$\mathbf{x}^w_{b_k,e_k} = \begin{cases} \mathbf{e}^w(w_{b_k,e_k}) & \text{if } w_{b_k,e_k} \in \mathbb{D} \\ \mathbf{p}^w(p_k) & \text{otherwise} \end{cases}, \tag{13}$$

where p_k is the corresponding POS tag of w_{b_k,e_k} provided by ξ_t, $\mathbf{e}^w$ and $\mathbf{p}^w$ are word and POS embedding lookup tables, respectively.

Training Training is performed on a source domain corpus only. We do not fine-tune word embeddings. The key task for knowledge transfer is the learning of POS embeddings, which offers a generalized representation for words not in the embedding lexicon. To this end, we randomly replace words in the training data with their gold-standard POS tags as follows:

$$P(w_{b_k,e_k}) = \min\left(1, \sqrt{\frac{t}{f(w_{b_k,e_k})}}\right), \tag{14}$$

where $f(w_{b_k,e_k})$ is the frequency of w_{b_k,e_k} in the training data and t is a chosen threshold.

The representation of w_{b_k,e_k} is:

$$\mathbf{x}^w_{b_k,e_k} = \begin{cases} \mathbf{p}^w(p_i) & 0 \le rb < P(w_{b_k,e_k}) \le 1 \\ \mathbf{e}^w(w_{b_k,e_k}) & \text{otherwise} \end{cases}, \tag{15}$$

where rb is a random number and p_i is the gold-standard POS tag of w_{b_k,e_k}. We name this method as integral POS embedding method (P_t).

Considering the positional information of characters in the word, the set of POS tags can be denoted in combination with segmentation labels: $L = \{p_{1b}, p_{1\text{-}m}, p_{1e}, p_{1\text{-}s}, \ldots, p_{ne}\}$. We name this method as individual POS embedding method (P_{t_b}). The difference between P_t and P_{t_b} is that for c_j and matched word w_{b_i,e_i}, if c_j is the first, middle or last character of w_{b_i,e_i}, the corresponding POS tag of w_{b_i,e_i} is p_{ib}, p_{im} and p_{ie},

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.

respectively. Fig. 2 shows the difference between P_t and P_{t_b} through an example.

For each character c_i, we integrate dictionary word information by augmenting its embedding with a word context vector h_i, which is the weighted sum over $\mathbf{x}^w_{b_k,e_k}$ for all spans (b_k, e_k) that contain c_i. In particular,

$$\mathbf{h}_i = \sum \alpha_{ik} \mathbf{x}^w_{b_k,e_k}, \tag{16}$$

where the weight for each context word is:

$$\begin{aligned}\boldsymbol{\alpha}_{ik} &= \text{attention}(\mathbf{x}^c_i, \mathbf{x}^w_{b_k,e_k}) \\ &= \frac{\exp(\text{score}(\mathbf{x}^c_i, \mathbf{x}^w_{b_k,e_k}))}{\sum_{k=1}^{m}(\exp(\text{score}(\mathbf{x}^c_i, \mathbf{x}^w_{b_k,e_k})))}.\end{aligned} \tag{17}$$

Considering computation efficiency, the score function is:

$$\text{score}(\mathbf{x}^c_i, \mathbf{x}^w_{b_k,e_k}) = \mathbf{x}^c_i W \mathbf{x}^w_{b_k,e_k}, \tag{18}$$

where W denotes a model parameter. The output of the attention layer is the concatenation of the character embedding $\mathbf{x}^c_i$ and the context vector $\mathbf{h}_i$:

$$\mathbf{x}_i = \mathbf{x}^c_i \oplus \mathbf{h}_i. \tag{19}$$

C. Decoding and Training

For decoding, the Viterbi algorithm [29] is used to find the highest scored label sequence y^* over a input sentence.

For training a set with N gold-standard samples, the loss function is negative log-likelihood of sentence-level with L_2 regularization:

$$\text{Loss} = -\sum_{i=1}^{N} \log(P(y_i|s_i)) + \frac{\lambda}{2}||\Theta||^2, \tag{20}$$

where λ and Θ represent L_2 regularization parameter and the model parameter set, respectively.

VI. Experiments

We carry out an extensive set of experiments to investigate the effectiveness of SAN-CRF and the proposed neural type-supervised domain adaptation method across different domains under different settings. F1-value is taken as our main metric.

A. Datasets

We separately evaluate the proposed model in in-domain and cross-domain settings. For in-domain evaluation, CTB6 (Chinese Tree Bank 6.0), PKU and MSR are taken as the datasets. The train/dev/test split of CTB6 follows [36], while the split of PKU and MSR are taken from the SIGHAN Bakeoff 2005 [8]. For cross-domain evaluation, PKU is used as the source domain, and three Chinese novel datasets including DL (DouLuoDaLu), FR (FanRenXiuXianZhuan) and ZX (ZhuXian) [20] are used as the target domains. Following [35], we collect target-domain lexicons from Internet Encyclopedia[3,4,5] and annotate every word in the lexicons with one POS tag. Table I shows the statistics of the datasets.

[3][Online]. Available: https://baike.baidu.com/item/诛仙/12418

[4][Online]. Available: https://baike.baidu.com/item/凡人修仙传/54139

[5][Online]. Available: https://baike.baidu.com/item/斗罗大陆/5313

TABLE I
Statistics of Datasets

Datasets		PKU	MSR	CTB6	ZX	FR	DL
Training set	#sent	19.1k	86.9k	23.4k	PKU		
	#word	1.11m	2.37m	641k			
	#char	1.83m	4.05m	1.06m			
Testing set	#sent	1.9k	4.0k	2.8k	0.7k	1.3k	1.3k
	#word	0.10m	0.11m	0.70m	35.2k	35.3k	31.5k
	#char	0.17m	0.18m	1.16m	50.3k	64.2k	52.2k

TABLE II
Hyper-Parameter Values

Parameter	Value	Parameter	Value
Char emb size	50	SAN layer num	2
Word emb size	200	SAN head num	2
Bigram emb size	50	SAN hidden size	100
BERT emb size	768	SAN Inner size	100
LSTM layer	3	SAN Relu dropout	0.1
LSTM hidden	100	Attention dropout	0.1
LSTM input dropout	0.1	Resiual dropout	0.1
Batch size	32	Window size	5

TABLE III
Effect of Numbers of BiLSTM Layer and Hidden Dimension on the CTB6 Development Dataset

#Layer	#Hidden Dimension	F1-Value	Model Size
1	100	95.48	11.94M
1	200	95.54	12.04M
1	300	95.42	12.19M
1	400	95.34	12.37M
2	100	95.57	12.00M
2	200	95.56	12.29M
2	300	95.49	12.73M
2	400	95.44	13.33M
3	100	95.58	12.06M
3	200	95.56	12.53M
3	300	95.46	13.27M
3	400	95.44	14.29M
4	100	95.41	12.13M
4	200	95.51	12.77M
4	300	95.40	13.81M
4	400	95.34	15.26M

B. Experimental Settings

Training Details Table II shows the values of model hyper-parameters. For the SAN CWS model, we use the Adam [11] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$. Following [28], we increase the learning rate linearly for the first *warmup_steps* steps, and then decrease it proportionally. The value of *warmup_steps* is set to 1000. When BERT is used for character embeddings, the learning rate is set to 5e-6. For the baseline model, we use **s**tochastic **g**radient **d**escent (SGD) follwing [33], and the initial learning rate is set to 0.001, which gives better development results.

Baseline Settings For the BiLSTM baseline, we ran a grid search on the CTB6 dataset to find the best values of BiLSTM layer and hidden dimension. The results are listed in Table III, which shows that the model with 3 layers and 100 hidden dimension achieves the best development result. We use this setting as our BiLSTM baseline model.

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.

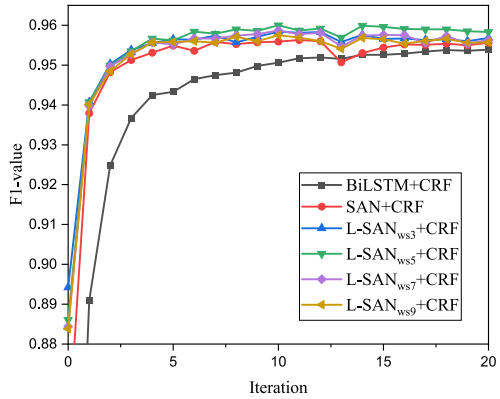


Fig. 3. Comparisons of BiLSTM, self-attention network and local self-attention network with different window sizes on the CTB6 development dataset. L-SAN$_{wsN}$ means local self-attention network with window size N.

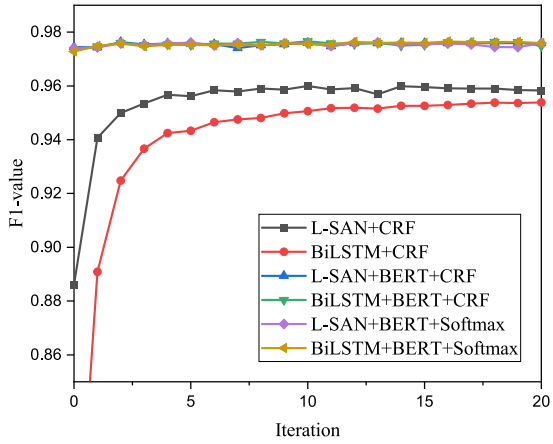


Fig. 4. Effectiveness of using BERT for in-domain CWS. "+BERT" means the model replaces the word2vec character unigram representation with BERT. "+Softmax" means the CRF layer is replaced with a softmax classification layer.

Character and Word Embedding The pre-trained word embedding size is 200, which is based on word co-occurrence and the directions of word pairs [23], and the word length is restricted to 4. we use the topmost layer output as character embedding of the pre-trained Chinese Simplified BERT model with 12 layers, 768 hidden units and 12 heads.[6] Besides that, the bigram embeddings and character unigram embeddings used for attending words are pre-trained using word2vec [1], which is the same as following [36].

C. Development Experiments

We perform development experiments on the CTB6 development dataset to investigate the influence of hyper-parameters of self-attention network for CWS, and compare the performance of SAN, especially local self-attention, with BiLSTM. In addition, we evaluate the effect of utilizing of BERT for CWS models. We denote the original self-attention network as "SAN," and *local self-attention network* as "L-SAN". "BiLSTM" is our baseline model, which uses a bidirectional LSTM as feature extractor.

Effect of Local Attention As Fig. 3 shows, the performance of "L-SAN+CRF" models with different window sizes give better results compared to "SAN+CRF," which suggests that long-range global context can bring more noise and harm segmentation. In addition, the proposed local self-attention network model achieves competitive results compared with the baseline BiLSTM model. Fig. 3 also shows the impact of different window sizes on the performance. First, we find that "L-SAN" with window size 5 achieves the best result. Second, when we increase the window size larger than 5, the performance decreases. We hypothesize that a larger window size brings more noise than information. However, if we set the window size smaller than 5, the performance also decreases, which is likely because a small window size does not capture sufficient context. Thus, we choose 5 as the window size for the local self-attention network.

Effect of BERT As shown in Fig. 4, by replacing word2vec character embeddings with BERT, both BiLSTM and L-SAN models reach the best F1-value within several epochs, with a significant improvement. We further demonstrates the effect of BERT by replacing the CRF layer with a softmax classification layer, which still achieves strong performance, proving that context-dependent word representation can benefit CWS even without a CRF layer.

[6]https://github.com/huggingface/pytorch-pretrained-BERT

TABLE IV
EFFECT OF NUMBERS OF HEADS AND LAYERS OF L-SAN ON THE CTB6 DEVELOPMENT DATASET

#Layer	#Head	F1-Value	Model Size
2	2	96.0	12.01M
2	4	96.0	12.31M
2	6	95.9	12.76M
2	8	95.8	13.37M
4	2	96.0	12.14M
4	4	95.8	12.71M
4	6	95.8	13.61M
4	8	95.8	14.82M
6	2	96.0	12.26M
6	4	95.9	13.11M
6	6	95.9	14.45M
6	8	95.3	16.27M

Width and Depth [28] show that increasing the model size can improve the performance of English-to-German translation. We investigate the effect of number of layers and heads on CWS by varying the number of layers between 2 and 6, and varying the number of heads from 2 to 8. The dimension of head is fixed to 50. The results and total numbers of parameters are listed in Table IV. We found that the model can achieve the best F1-value 96.0 with only 2 layers and 2 heads, yet the F1-value does not increase simultaneously with the increase of model size. According to this result, we fix the number of layers and heads to 2 and 2 for the remaining experiments, respectively.

D. Final Results

In-Domain Results We evaluate our model on three news datasets, including CTB, PKU and MSR. The main results and the results of recent state-of-the-art models are listed in Table V. Compared with the baseline "BiLSTM+CRF" model, the proposed "L-SAN+CRF" model achieves similar results, which

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.

TABLE V
IN DOMAIN RESULTS

Models	CTB6	PKU	MSR
Zhang et al. [36]	96.0	95.7	97.7
Cai et al. [3]	-	95.8	97.1
Yang et al. [32]	96.2	96.3	97.5
Zhou et al. [39]	96.2	96.0	97.8
Zhang et al. [37]	96.4	96.5	97.8
Ma et al. [16]	96.7	96.1	98.1
BiLSTM + CRF	95.3	95.0	97.3
L-SAN + CRF	95.5	95.1	97.1
BERT + Softmax	97.4	96.6	98.4
BERT + CRF	97.4	96.6	98.4
BiLSTM + BERT + Softmax	97.4	96.6	98.4
BiLSTM + BERT + CRF	**97.5**	96.7	98.3
L-SAN + BERT + Softmax	97.3	96.6	98.3
L-SAN + BERT + CRF	97.4	**96.7**	**98.4**

TABLE VI
CROSS DOMAIN RESULTS

Model	ZX	FR	DL
Liu and Zhang [14]	87.2	87.5	91.4
Qiu and Zhang [20]	87.4	86.7	91.9
Ye et al. [34]	89.6	89.6	93.5
BiLSTM + BERT + CRF	90.6	90.7	93.1
L-SAN + BERT + CRF	90.5	91.1	93.0
BiLSTM + BERT + CRF + *t*	91.5	92.0	93.6
L-SAN + BERT + CRF + *t*	91.8	92.3	94.3
BiLSTM + BERT + CRF + *t_b*	93.1	92.4	**95.6**
L-SAN + BERT + CRF + *t_b*	**93.1**	**93.0**	95.1

proves that self-attention network can be a competitive feature extractor for CWS. When replacing word2vec character embedding with BERT, the "BiLSTM+BERT+CRF" model gives 46.8%, 34.0% and 37.0% error reduction on CTB6/PKU/MSR, respectively, and the "L-SAN+BERT+CRF" model has 42.2%, 32.7% and 44.8% error reductions on three in-domain datasets, respectively. Fig. 5 also shows that replacing the CRF layer with a softmax classification layer can still achieve competitive results, which demonstrates the strong effect of BERT. We also try to remove BiLSTM or L-SAN module and directly inference based on the output of BERT. The results again show the strong effect of BERT on in-domain CWS.

Cross-Domain Results We evaluate our model on the three cross-domain datasets, including ZX, FR and DL. The main results and results of three state-of-the-art models are listed in Table VI. "*t*" means that a neural type-supervised method is used to learn POS embeddings and domain-specific words are generalized to corresponding tags. In "*t_b*," we learn different POS embeddings for different positions in a word, as mentioned in Section V. As shown in Table VI, the F1-values of "L-SAN+BERT+CRFT" have an average 0.7 improvement compared with the state-of-the-art results [34] on ZX and FR without using [34]'s domain adaptation techniques, which show that BERT has relatively less effect on cross-domain CWS compared with strong domain adaptation methods. This may be because ZX, FR and DL are all Chinese novels which contain a large number of noun entities, and their writing styles are different from the news domain. "L-SAN+BERT+CRF+*t*" model has 21.15%, 25.96% and 1.54% error reduction on the ZX/FR/DL datasets, respectively, which shows that the proposed

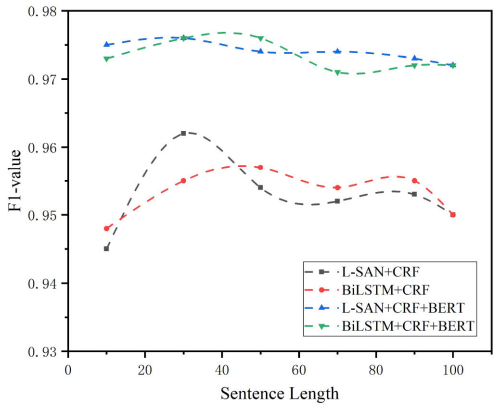


Fig. 5. F1-value against the sentence lengths.

TABLE VII
COMPARISON OF THE TRAINING TIME FOR ONE EPOCH AND TEST SPEED. ST MEANS SENTENCES

Model	Model Size	Training (s)	Test (st/s)
BiLSTM+CRF	12.06M	185.35	642.15st/s
L-SAN + CRF	12.01M	180.29	704.34st/s

neural type-supervised method can handle out of vocabulary words more effectively. For characters within a word, instead of sharing the same POS embedding of the word, we further distinguish POS embeddings of characters according to their position in a word. "L-SAN+BERT+CRF+*t_b*" gives 33.65%, 32.69% and 24.62% on the three datasets, respectively, which shows the advantage of more supervision information. Fig. 6 also shows that our domain adaptation method based on BiLSTM can still achieve the state-of-the-art results on the three cross-domain datasets.

E. Analysis

Sentence Length As shown in Fig. 5, we compare the baseline model and local self-attention network model, as well as the two models with BERT input representation against different sentence lengths on the CTB6 test dataset. The two models without using BERT show a similar performance-length curves, which reach a peak at around 30-character sentences and decrease when sentence length over 90. One possible reason is that very short sentences are rare while long sentences are semantically more challenging. However, the two models using BERT show more stable performance-length curves, which shows that contextualized BERT representation can stabilize the performance against the sentence length.

Comparison of Training and Test Speed We compare the training and test speeds of the L-SAN model with the BiLSTM baseline on the CTB6 dataset. Both models are evaluated on the same server with a GeForce GTX 1080Ti GPU. The batch size is set to 32 for both training and test. The results are listed in Table VII. We can find that the "L-SAN+CRF" model outperforms "BiLSTM+CRF" in both training and test speeds with almost the same model size, which demonstrates the advantage of parallelization by SAN.

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII
CASE STUDIES, X REPRESENTS UNGRAMMATICAL WORD

#Example 1: 韩立也在光罩边缘处止住了下落的身影 Han Li also stopped the falling figure at the edge of the mask	
Gold Segmentation	韩立/也/在/光罩/边缘/处/止住/了/下落/的/身形 Han Li/also/at/the mask/the edge/of/stopped/x/the falling/figure
L-SAN+CRF+BERT	韩/立/也/在/光罩/边缘/处/止住/了/下落/的/身形 Han/Li/also/at/the mask/the edge/of/stopped/x/the falling/figure
L-SAN+CRF+BERT+*t*	韩立/也/在/光罩/边缘/处/止住/了/下落/的/身形 Han Li/also/at/the mask/the edge/of/stopped/x/the falling/figure
L-SAN+CRF+BERT+*t_b*	韩立/也/在/光罩/边缘/处/止住/了/下落/的/身形 Han Li/also/at/the mask/the edge/of/stopped/x/the falling/figure
#Example 2: 戴沐白虎掌上利刃弹开 Dai Mubai pops up the blade on the palm	
Gold Segmentation	戴沐白/虎掌/上/利刃/弹开 Dai Mubai/palm/on/blade/pops up
L-SAN+CRF+BERT	戴/沐/白虎/掌/上/利刃/弹开 Dai/Mu/white tiger/palm/on/blade/pops up
L-SAN+CRF+BERT+*t*	戴沐/白虎/掌上/利刃/弹开 Dai Mu/white tiger/palm/on/blade/pops up
L-SAN+CRF+BERT+*t_b*	戴沐白/虎掌/上/利刃/弹开 Dai Mubai/palm/on/blade/pops up

TABLE IX
SEGMENTATION PRECISION OF NOUN ENTITIES WITH THE HIGHEST FREQUENCY

Word	Count	M1	M2	M3
唐三(Person Name)	273	0.98	0.99	1.00
韩立(Person Name)	185	0.07	0.67	1.00
戴沐白(Person Name)	159	0.01	0.31	1.00
小舞(Person Name)	153	0.90	0.98	1.00
张小凡(Person Name)	142	0.00	0.06	1.00
玄骨(Person Name)	114	0.96	0.97	0.98
魂狮(Proper Name)	90	1.00	1.00	1.00
宁荣荣(Person Name)	86	0.01	0.57	1.00
朱竹清(Person Name)	81	0.03	0.76	1.00
魂环(Proper Name)	72	1.00	1.00	1.00
魂力(Proper Name)	71	0.97	0.99	1.00
魂兽(Proper Name)	53	1.00	1.00	1.00
斗魂(Proper Name)	51	0.71	0.73	0.76
叶知秋(Person Name)	51	0.00	0.00	0.00
乌丑(Person Name)	45	0.97	1.00	1.00
Average Precision	108	0.55	0.73	0.96

Noun Entity Segmentation Noun entities raise a key problem for cross-domain CWS. Table IX shows segmentation results of the three models on the 15 most frequent noun entities on the three datasets. M1 and M2 represent "L-SAN+CRF+BERT" and "L-SAN+CRF+BERT+*t*," respectively, while M3 represents "L-SAN+CRF+BERT+*t_b*". As the table shows, the average precision of MI is 0.55. By using neural type-supervised domain adaptation method, the average precision of M2 has a improvement of 0.18 in absolute value. Some person names are incorrectly segmented by M2, such as "戴 沐 白(Person Name)" and "张 小 凡(Person Name)". When incorporating the character positional information in the word, the average segmentation precision improves further and most noun entities can be correctly segmented (except for the word "叶 知 秋(Person Name)," the main reason is that the domain lexicon does not contain "叶 知 秋(Person Name)"). This shows that our method makes effective use of domain lexicons.

Case Study As shown in Table VIII, we use two examples of neural type-supervised domain adaptation for discussion. In example 1, "L-SAN+CRF+BERT" fails to handle the domain entity noun "韩 立(Person Name)" while the two neural type-supervised domain adaptation method segment it correctly, which shows the effect of our domain adaptation methods for noun entities. However, for example 2, only "L-SAN+CRF+BERT+*t_b*" segments it correctly. One possible reason is that it maybe difficult to distinguish between "韩 立白 (Person Name)" and "白虎 (White Tiger)," which shows the importance of character positional information.

VII. CONCLUSION

We investigated self-attention network (SAN) for Chinese word segmentation, demonstrating that it can achieve comparable results with recurrent network methods. We found that local attention gives better results compared to standard SAN. Under SAN, we also investigate the influence of rich character and word features, including BERT character embeddings and a neural attention method to integrate word information into character based CWS. Extensive in-domain and cross-domain experiments show that the proposed SAN method achieves the state-of-the-art performance on both in-domain and cross-domain Chinese word segmentation datasets.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations (Workshop Poster)*, 2013.
[2] D. Cai and H. Zhao, "Neural word segmentation learning for Chinese," in *Pro. 54th Annu. Meeting Assoc. Comput. Linguist. (Volume 1: Long Papers)*, 2016, vol. 1, pp. 409–420.
[3] D. Cai, H. Zhao, Z. Zhang, Y. Xin, Y. Wu, and F. Huang, "Fast and accurate neural word segmentation for Chinese," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist. (Volume 2: Short Papers)*, 2017, vol. 2, pp. 608–615.
[4] X. Chen, X. Qiu, C. Zhu, and X. Huang, "Gated recursive neural network for Chinese word segmentation," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguist. 7th Int. Joint Conf. Natural Lang. Process. (Volume 1: Long Papers)*, 2015, vol. 1, pp. 1744–1753.
[5] X. Chen, X. Qiu, C. Zhu, P. Liu, and X. Huang, "Long short-term memory neural networks for Chinese word segmentation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1197–1206.
[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. Human Lang. Technol.*, 2019, vol. 1, pp. 4171–l4186.
[7] J. L Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.
[8] T. Emerson, "The second international Chinese word segmentation bakeoff," in *Proc. 4th SIGHAN Workshop Chin. Lang. Process.*, 2005. [Online]. Available: https://www.aclweb.org/anthology/I05-3017
[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[10] W. Jiang, M. Sun, Y. Lü, Y. Yang, and Q. Liu, "Discriminative learning with natural annotations: Word segmentation as a case study," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguist. (Volume 1: Long Papers)*, 2013, vol. 1, pp. 761–769.
[11] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[12] N. Kitaev and D. Klein, "Constituency parsing with a self-attentive encoder," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist. (Volume 1: Long Papers)*, 2018, vol. 1, pp. 2676–2686.
[13] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 282–289.
[14] Y. Liu and Y. Zhang, "Unsupervised domain adaptation for joint segmentation and POS-tagging," *Proc. COLING 2012: Posters*, 2012, pp. 745–754.

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.

[15] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
[16] J. Ma, K. Ganchev, and D. Weiss, "State-Of-The-Art Chinese word segmentation with Bi-LSTMs," in *Proc. Conf. Empirical Methods Natural Lang. Proc.*, 2018, pp. 4902–4908.
[17] 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. Proc. Syst.*, 2013, pp. 3111–3119.
[18] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[19] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proc. 16th Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist. Human Lang. Techno.*, 2018, vol. 1, pp. 2227–2237.
[20] L. Qiu and Y. Zhang, "Word segmentation for Chinese novels," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2440–2446.
[21] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018. [Online]. Available: URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf
[22] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "DiSAN: Directional self-attention network for RNN/CNN-free language understanding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5446–5455.
[23] 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. Linguist. Human Lang. Technol., Volume 2 (Short Papers)*, 2018, vol. 2, pp. 175–180.
[24] E. Strubell, P. Verga, D. Andor, D. Weiss, and A. McCallum, "Linguistically-informed self-attention for semantic role labeling," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 5027–5038.
[25] W. Sun and J. Xu, "Enhancing Chinese word segmentation using unlabeled data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 970–979.
[26] Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi, "Deep semantic role labeling," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4929–4936.
[27] G. Tang, M. Müller, A. Rios, and R. Sennrich, "Why self-attention? A targeted evaluation of neural machine translation architectures," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4263–4272.
[28] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[29] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Informat. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
[30] J. Xu and X. Sun, "Dependency-based gated recursive neural network for Chinese word segmentation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist. (Volume 2: Short Papers)*, 2016, vol. 2, pp. 567–572.
[31] N. Xue, "Chinese word segmentation as character tagging," *Int. J. Comput. Linguist. Chi. Lang. Process., vol. 8, no. 1, Feb.: Special Issue Word Formation Chi. Lang. Process.*, vol. 8, no. 1, pp. 29–48, Feb. 2003.
[32] J. Yang, Y. Zhang, and F. Dong, "Neural word segmentation with rich pretraining," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.*, 2017, vol. 1, pp. 839–849.
[33] J. Yang, Y. Zhang, and S. Liang, "Subword encoding in lattice LSTM for Chinese word segmentation," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 2720–2725.
[34] Y. Ye, W. Li, Y. Zhang, L. Qiu, and J. Sun, "Improving cross-domain Chinese word segmentation with word embeddings," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 2726–2735.
[35] M. Zhang, Y. Zhang, W. Che, and T. Liu, "Type-supervised domain adaptation for joint segmentation and pos-tagging," in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguist.*, 2014, pp. 588–597.
[36] M. Zhang, Y. Zhang, and G. Fu, "Transition-based neural word segmentation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist. (Volume 1: Long Papers)*, 2016, vol. 1, pp. 421–431.
[37] Q. Zhang, X. Liu, and J. Fu, "Neural networks incorporating dictionaries for Chinese word segmentation," in *Proc. AAAI-18 AAAI Conf. Artif. Intell.*, 2018, pp. 5682–5689.
[38] Y. Zhang and S. Clark, "Chinese segmentation with a word-based perceptron algorithm," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguist.*, 2007, pp. 840–847.
[39] H. Zhou, Z. Yu, Y. Zhang, S. Huang, X.-Y. DAI, and J. Chen, "Word-context character embeddings for Chinese word segmentation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 760–766.
[40] W. Huang, X. Cheng, K. Chen, T. Wang, and W. Chu, "Toward fast and accurate neural Chinese word segmentation with multi-criteria learning," 2019, *arXiv:1903.04190*.
[41] X. Qiu, H. Pei, H. Yan, and X. Huang, "A concise modelfor multi-criteria chinese word segmentation with transformer encoder," 2019, *arXiv:1906.12035*.
[42] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones, "Character-level language modeling with deeper self-attention," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3159–3166.
[43] B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, and T. Zhang, "Modeling localness for self-attention networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4449–4458.
[44] M. Xu, D. F Wong, B. Yang, Y. Zhang, and L. S. Chao, "Leveraging local and global patterns for self-attention networks," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguist.*, 2019, pp. 3069–3075.
[45] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol., Volume 2 (Short Papers)*, 2018, pp. 464–468.
[46] C. Wang and B. Xu, "Convolutional neural network with word embeddings for Chinese word segmentation," in *Proc. 8th Int. Joint Conf. Natural Lang. Process. (Volume 1: Long Papers)*, 2017, pp. 163–172.
[47] H. Yan, X. Qiu, and X. Huang, "A graph-based model for joint Chinese word segmentation and dependency parsing," *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 78–92, 2020.

Leilei Gan received the B.S. degree from the School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing, China, and the M.S. degree from the Department of Computer Science and Technology, Nanjing University, Nanjing, China. He is currently working toward the Ph.D. degree with the School of Computer Science and Technology, Zhejiang University, Hangzhou, China His research interest is mainly focuses on natural language processing.

Yue Zhang is currently an Associate Professor with Westlake University, Hangzhou, China. His research interests include natural language processing and computational finance. He has been working on statistical parsing, text synthesis, natural language synthesis, machine translation, information extraction, sentiment analysis, and stock market analysis intensively. He won the best paper awards of IALP 2017 and COLING 2018. He serves the editorial board for the *Transactions of the Association of Computational Linguistics* (Action Editor), *ACM Transactions on Asian and Low-Resource Language Information Processing* (Associate Editor), and IEEE TRANSACTIONS ON BIG DATA (Associate Editor), and as area chairs of COLING 2014/18, NAACL 2015/19, EMNLP 2015/17/19, and ACL 2017/18/19.

Authorized licensed use limited to: Westlake University. Downloaded on January 03,2023 at 05:57:32 UTC from IEEE Xplore. Restrictions apply.