Improving Chinese Named Entity Recognition by Large-Scale Syntactic Dependency Graph

Peng Zhu, Dawei Cheng, Fangzhou Yang, Yifeng Luo, Dingjiang Huang, Weining Qian, *Member, IEEE*, and Aoying Zhou

***Abstract*—Named entity recognition (NER) is a preliminary task in natural language processing (NLP). Recognizing Chinese named entities from unstructured texts is challenging due to the lack of word boundaries. Even if performing Chinese Word Segmentation (CWS) could help to determine word boundaries, it is still difficult to determine which words should be clustered together for entity identification, since entities are often composed of multiple-segmented words. As dependency relationships between segmented words could help to determine entity boundaries, it is crucial to employ information related to syntactic dependency relationships to improve NER performance. In this paper, we propose a novel NER model to learn information about syntactic dependency graphs with graph neural networks, and merge learned information into the classic Bidirectional Long Short-Term Memory (BiLSTM) - Conditional Random Field (CRF) NER scheme. In addition, we extract various kinds of task-specific hidden information from multiple CWS and part-of-speech (POS) tagging tasks, to further improve the NER model. We finally leverage multiple self-attention components to integrate multiple kinds of extracted information for named entity identification. Experimental results on three public benchmark datasets show that our model outperforms the state-of-the-art baselines in most scenarios.**

***Index Terms*—Graph neural network, multi-task learning, name entity recognition, self-attention, syntactic dependency graph.**

I. Introduction

NAMED entity recognition (NER) is an important preliminary information extraction task in natural language processing (NLP). It involves determining the boundaries of named entities, related to person names, locations, organizations, etc. Also, it identifies their entity categories from unstructured texts, which could then be provided to various downstream NLP tasks for information acquisition, such as information retrieval [1], [2], question answering [3], [4], relation extraction [5]–[7], event extraction [8]–[10], entity linking [11], [12], etc. Compared with Indo-European languages, Chinese NER is especially challenging due to the lack of word boundaries. Ignoring word-level information and directly using character-level information to recognize Chinese entities usually results in worse performance, whereas performing Chinese word segmentation (CWS) to leverage word-level information could help determine word boundaries.

However, it is still challenging to determine the entity boundaries concerning which segmented words should be clustered together as entities, since a named entity usually contains multiple-segmented words. In addition to word segmentation, determining entity boundaries needs to consider the meanings and syntactic dependencies of words, as the same set of characters may denote various meanings with dissimilar CWS results and syntactic dependency relationships, resulting in various NER results. For example, the character sequence "中国建设银行 (China Construction Bank)" contained in the sentence "中国建设银行拥有一万三千多家分支机构 (China Construction Bank has more than 13,000 branches)" is identified as an entity, while the character sequence "中国 (China)" contained in sentence "在中国建设银行分支机构需要得到审批 (it needs to get approval to build bank branch institutions in China)," rather than "中国建设银行 (build bank branch institutions in China)," is identified as an entity, just as shown in Fig. 1. We can see that leveraging syntactic dependencies [13] could help identify named entities. However, auxiliary information, such as CWS information, that helps to parse the syntactic dependencies of sentences, is concealed from the final syntactic dependency parsing results, and is not explicitly available for NER, while just leveraging syntactic dependency parsing results may bring in incorrect entity boundaries, and thus achieves suboptimal objectives.

In this paper, we propose an NER model, namely syntactic dependency information-named entity recognition (SDI-NER), to improve NER performance via leveraging syntactic dependency information with graph neural networks. Even though syntactic dependency parsing has been used for relation extraction [14], [15], we are the first to learn from the large-scale syntactic dependency graph for named entity recognition. Our model is implemented via multi-task learning, where a first

Manuscript received May 6, 2021; revised October 21, 2021 and January 17, 2022; accepted February 6, 2022. Date of publication February 28, 2022; date of current version March 5, 2022. This work was supported in part by the National Key R&D Program of China under Grant 2018YFC0831904, in part by the National Natural Science Foundation of China under Grants U1711262 and 62072185, and in part by the Joint Research Program of SeekData Inc. and ECNU. The work of Weining Qian was supported by the Research Funds of Happiness Flower ECNU under grant number 2019ECNU-XFZH006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Eric Fosler-Lussier. *(Corresponding authors: Dawei Cheng; Yifeng Luo.)*

Peng Zhu, Yifeng Luo, Dingjiang Huang, Weining Qian, and Aoying Zhou are with the School of Data Science and Engineering, East China Normal University, Shanghai 200062, China (e-mail: pzhu@stu.ecnu.edu.cn; yfluo@dase.ecnu.edu.cn; djhuang@dase.ecnu.edu.cn; wnqian@dase.ecnu.edu.cn; ayzhou@dase.ecnu.edu.cn).

Dawei Cheng is with the Department of Computer Science and Technology, Tongji University, Shanghai 200070, China (e-mail: dcheng@tongji.edu.cn).

Fangzhou Yang is with SeekData Inc., Shanghai 200070, China (e-mail: fangzhou.yang@seek-data.com).

Digital Object Identifier 10.1109/TASLP.2022.3153261

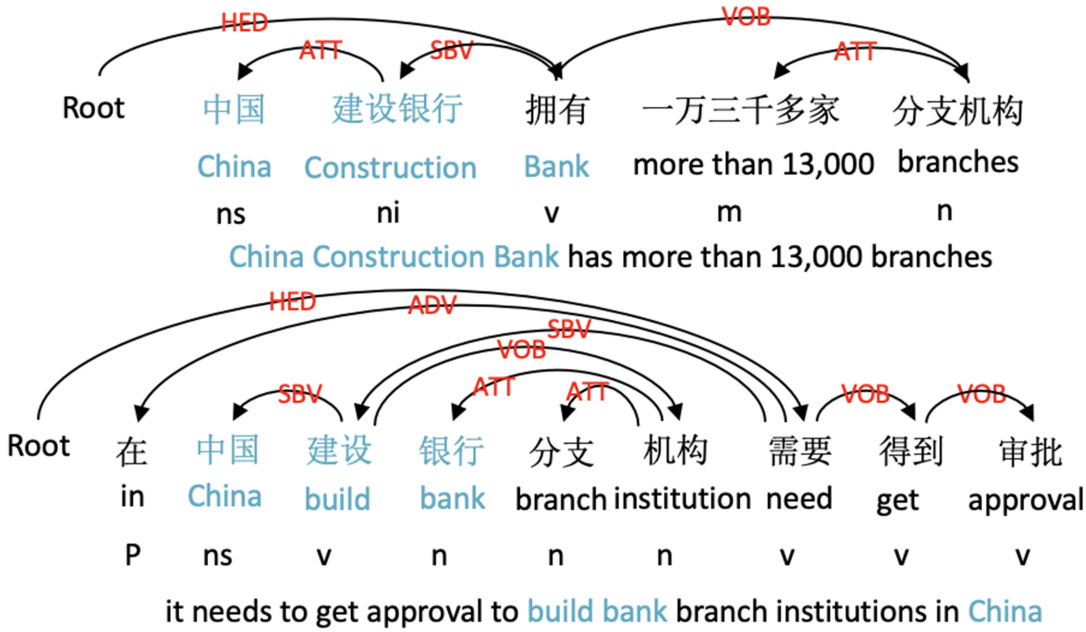


Fig. 1. Syntactic dependency graph of two sentences. In the POS tagging of the figure, "ns" is the geographical name, "ni" is the organization name, "v" is the verb, "m" is the number, "n" is the general noun and "p" is the preposition.

task learns word embeddings from syntactic dependencies with graph neural network, a second task learns multiple kinds of contextualized character embeddings as auxiliary information, and a third task merges the learned information into the classic BiLSTM-CRF to improve NER performance with multiple self-attention components. We transform the dependency parsing trees of sentences into uni-directed graphs, and learn word embeddings with syntactic dependency information via training a graph neural network (GNN) on these uni-directed graphs to approach the ground-truth NER labels. Multiple kinds of contextualized character embeddings are learned via training multiple deep neural networks to approach the CWS and POS tagging labels, generated with corresponding open-source tools. In the third task, character embeddings learned from the first task are fed to a BiLSTM network to learn contextualized character embeddings. Finally, the learned contextualized character and word embeddings are used to train corresponding self-attention components to extract key features for NER predictions via approaching the ground-truth labels.

In summary, we make the following contributions in this paper: 1) To the best of our knowledge, this is the first work that learns from the large-scale syntactic dependency graph for improving Chinese named entity recognition, based on graph neural network. 2) We learn multiple kinds of contextualized character and word embeddings from multiple tasks to help identify entity boundaries. 3) We leverage multiple self-attention components to extract key features contained within various kinds of contextualized characters and word embeddings to predict NER results. 4) We perform extensive experiments on three public Chinese NER datasets, and the results show that our model consistently achieves comparable results to the compared baseline NER models. In the rest of this paper, we introduce the related work in Section II, and illustrate the methodologies and techniques implemented in our model in Section III The experimental evaluations and case study are presented in Section IV and Section V respectively, and we finally conclude the paper in Section VI.

II. Related Work

Research on named entity recognition can mainly be categorized into rule and dictionary-based methods, statistical machine learning-based methods, and deep learning-based methods.

Rule and dictionary-based methods are the earliest methods with named entity libraries and manual rules. Methods based on statistical machine learning mainly include models based on hidden Markov model (HMM) [16], maximum entropy model (MEM), support vector machine (SVM) [17], and conditional random fields (CRF) [18], where CRF is currently the most widely used statistical NER method.

Most deep learning-based methods [19]–[21] inherit the classical LSTM-CRF or CNN-CRF architecture. [22] employs BiLSTM to extract hidden features, and then feeds them into the CRF decoder to decode label predictions. [23] and [24] use convolutional neural network (CNN) to learn word representations from characters for English NER. [25] proposes a semi-supervised learning model based on a BiLSTM neural network, with massive unlabeled texts and quite limited labeled texts, and [26] proposes a gated convolutional neural network (GCNN) model for Chinese NER. [27] proposes a novel word-character LSTM (WC-LSTM) model for Chinese NER, adding word information into the start or the end character of a word. [28] investigates a convolutional attention network (CAN) for Chinese NER, combining a character-based CNN with a local-attention layer and a gated recurrent unit (GRU) with a global self-attention layer. [29] presents an approach for pretraining a

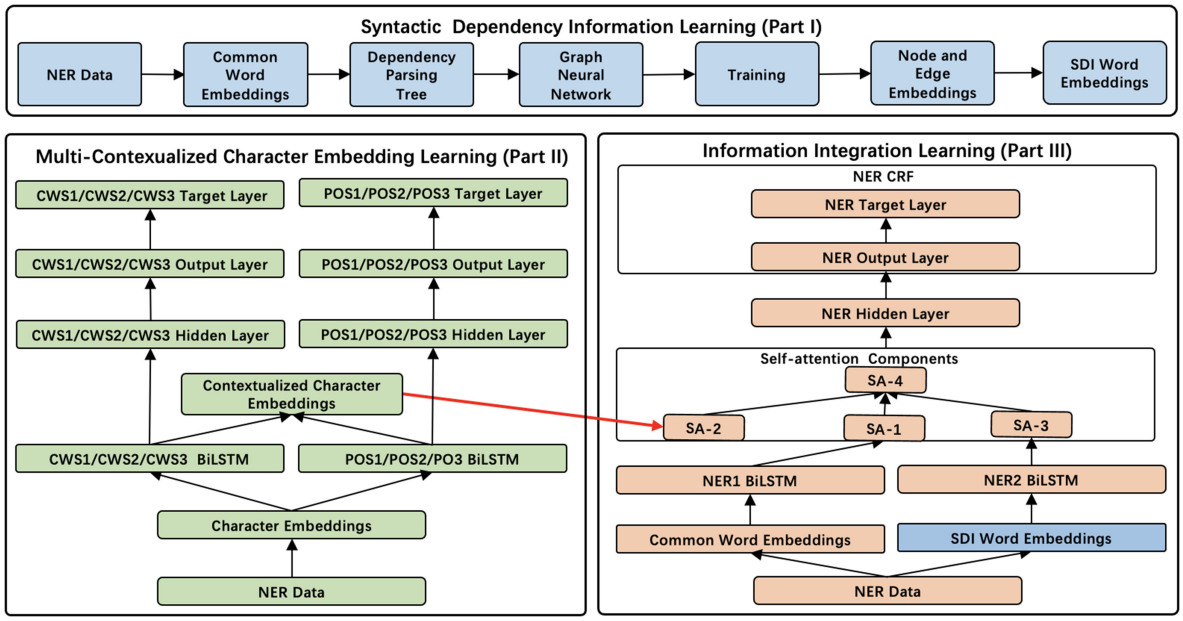


Fig. 2. Architecture of SDI-NER, where CWS and POS labels for task training are generated by Thulac, Pyltp and Jieba.

bidirectional transformer model, and achieves significant performance gains across various language understanding problems in English. [30] adds a module into the hybrid semi-Markov CRF architecture, and shows that utilizing external gazetteers benefits segmental neural NER models. [31] uses multi-metadata embedding that fuses the information of radicals, characters and words through a Cross-Transformer network. [32] proposes DC-SAN, a Dynamic Cross and Self-lattice Attention Network that models dense interactions over word-character lattice structure for Chinese NER.

Recently, graph neural networks and lexicon information have been applied to recognition of named entities [33]–[35]. [33] investigates that dependency trees play a positive role for entity recognition by using graph convolutional network (GCN) to boost the results of a bidirectional LSTM. [34] uses a lexicon to construct a graph neural network for Chinese NER. [35] directly leverages lexical knowledge with a collaborative graph network to recognize Chinese named entities efficiently. [36] firstly propose a Star-transformer based NER system. Then both explicit head and tail boundary information and Dependency GAT-based implicit boundary information are combined to improve Chinese NER. However, these methods usually use dictionary information and graph networks to identify entities, while ignoring syntactic dependencies between words in sentences. Besides, these methods do not use CWS and POS tagging information, which could help identify entities better. In addition, [13] uses different types of syntactic information to improve NER through focused combinations. [37] uses the idea of graph-based dependency analysis to provide a global view of the inputs to the model through a biaffine [38] model. [39] proposes a neural-based approach to social media text NER, where both local and enhanced syntactic are considered. [40] exploits NTP as an auxiliary task to enhance NER. [41] proposes an end-to-end model to learn optimal assignments of latent NER tags using observed tokens and weak labels provided by labeling functions. In comparison, our approach makes full use of the information obtained from multiple tagging tools, and learns the large-scale syntactic dependency graph based on the graph neural network to improve NER performance.

III. The SDI-NER Model

Our NER model consists of the syntactic dependency information learning module (Part I), the multi-contextualized character embedding learning module (Part II), and the information integration learning module (Part III), as illustrated in Fig. 2. Part I learns word embeddings with syntactic dependency information via ground-truth NER labels, based on the graph neural network. Part II learns multiple kinds of contextualized character embeddings, via approaching CWS and POS tagging labels generated with multiple open-source tools. As each of these tools implements CWS and POS tagging in its own right and has its own merits, we intend to combine their merits to improve NER performance with the learned multiple kinds of contextualized character embeddings. Part III integrates all learned information with self attentions via approaching ground-truth NER labels. Part I is trained separately in advance on all NER training datasets in order to obtain "SDI Word Embeddings," which are used in the next. Then Part II and Part III are trained together with multi-task learning, where the NER datasets are used for model training in turn during various rounds of multi-task learning. We will introduce our model in detail in the rest of the section.

A. Data Encoding and Labels

The NER datasets contain NER labels manually annotated, CWS, and POS labels generated with tagging tools. All these labels are deemed as ground-truth labels for sample loss computations during model training. The CWS and POS labels are used for training the model's left module, while NER labels for the model's middle and right module. The left module

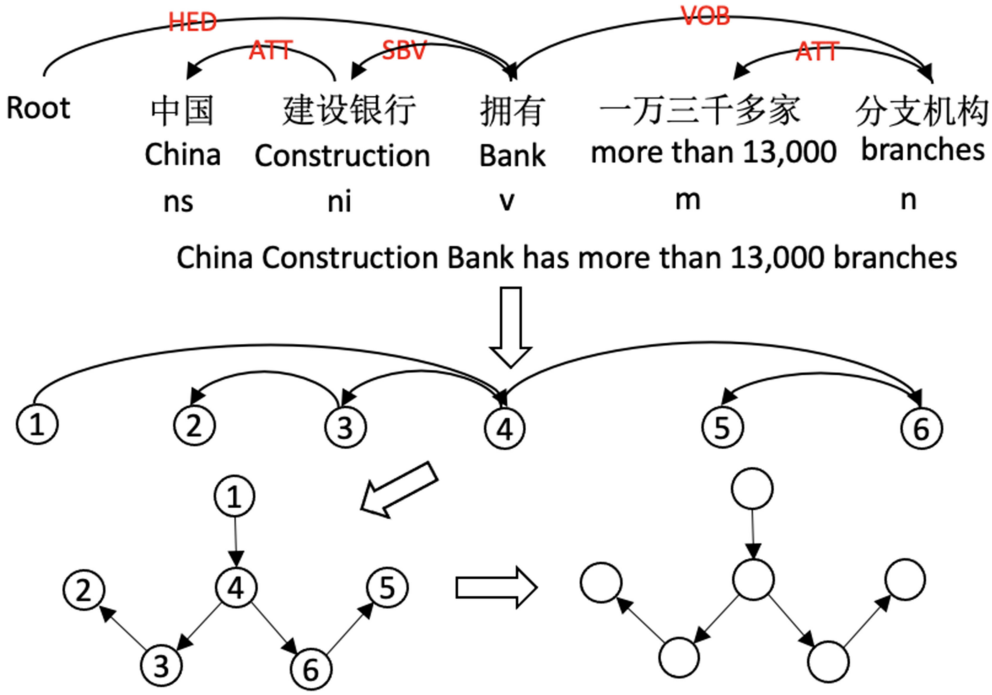


Fig. 3. Example of converting syntactic dependency graphs to uni-directed graphs.

encodes sentences with character embeddings, and the middle and right modules encode sentences with word embeddings. Besides, word embeddings with syntactic dependency information learned from the right module, together with the multiple kinds of contextualized character embeddings learned from the left module, are fed to the middle module for model training. The characters and words of sentences contained within the training corpora are initially mapped to their pre-trained embeddings.

B. Contained Modules in SDI-NER

1) Syntactic Dependency Information Learning: We generate dependency parsing trees for sentences contained in the NER training datasets with Pyltp toolkit [42] and then transform them into uni-directed graphs, where words in a sentence are deemed as nodes, and dependency relationships between words are deemed as edges. As shown in Fig. 3, first, each word in a sentence is represented as a node, and the edge between nodes is the connection between words in the syntactic dependency tree. Then these syntactic dependency elements could be formulated as undirected graphs. These uni-directed graphs are merged into a large graph, where two uni-directed graphs containing the same words are connected through nodes corresponding to overlapped words. We then feed the large graph into the GNN to predict words' NER categories via approaching ground-truth NER labels. The embeddings of nodes and edges are adjusted during network training, and syntactic dependency information for NER is learned when the network training finishes. The syntactic dependence relationship here refers to the dependence relationship between words, such as attributive intermediate relationship (ATT), verb-object relationship (VOB), and subject-predicate relationship (SBV), etc.[1] The syntactic dependence information refers to the information of the dependence relationship between words. When the graph network is trained, the representation of each node in the graph is the representation of each word in the sentence. When the graph network training is completed, the representation of these words contains information about the relationship with other words, and the information is syntactic dependent information. We take the weighted embedding of a node and its edges as the embedding of the corresponding word to make NER predictions in information integration learning.

2) Multi-Contextualized Character Embedding Learning: We use multiple open source CWS and POS tagging tools (Thulac, Jieba and Pyltp) to generate CWS and POS labels for each sentence contained in the NER datasets. Each kind of these labels is taken as ground-truth labels to train the model, to yield corresponding approximate label prediction results. The encoded characters contained in a sentence of the NER datasets are input into the CWS and POS BiLSTM networks to generate their contextualized character embeddings, which are fed to corresponding hidden layers and output layers to generate CWS and POS prediction results. When the model training finishes, the predicted CWS or POS labels should approximate the ground-truth labels generated with the used tools, we splice the inputs from six different BiLSTM networks to get the outputs of the contextualized character embeddings. The contextualized character embeddings output by the BiLSTM networks should contain useful task-specific information learned from each task, and they are fed to a self-attention neural network, called SA-2, in the information integration learning.

3) Information Integration Learning: In information integration learning, we extract key features from multiple kinds of character and word embeddings to predict NER results. We feed sentences encoded with common word embeddings

[1] http://www.ltp-cloud.com/intro

to a BiLSTM network, called NER1, for contextualized word embedding learning. The resultant embeddings are further fed to a self-attention neural network SA-1 for feature extraction. Similarly, the word embeddings learned from syntactic dependency information learning are fed to a BiLSTM network, called NER2, for contextualized word embedding learning, which are then fed to a self-attention neural network SA-3 to capture the key features concerned with syntactic dependency information. The output contextualized word embeddings of SA-1 and SA-3, combined with the contextualized character embeddings (SA-2) learned from the multi-contextualized character embedding learning, are fed to the fourth self-attention neural network, SA-4, for information integration. The output embeddings of SA-4 are fed to an NER hidden layer and a CRF-based output layer for NER label prediction. Finally, the predicted labels are used to compute sample losses in the target layer, compared with the ground-truth NER labels.

C. Neural Networks in SDI-NER

1) Graph Neural Network: The merged graph, consisting of all uni-directed graphs transformed from all sentences' dependency parsing trees, is fed to the graph neural network for model training. We denote the graph network as $G = (V, E)$, where V denotes the node set and E denotes the edge set. We denote node v's neighbor node set as $ne[v]$, and edge set as $co[v]$. As nodes and edges are represented as embedding vectors in graph neural networks, we denote the embedding of node v as $l_c \in \mathbb{R}^{d_V}$ and that of edge (v_1, v_2) as $l_{(v1,v2)} \in \mathbb{R}^{d_E}$, where d_V and d_E respectively denote the dimensional sizes of node embeddings and edge embeddings [43]. Let $\mathcal{G}$ denote the set of sub-graphs of the graph network G, and $\mathcal{V}$ denote the set of node subsets. Named entity recognition is to determine whether each word in a sentence is an entity, and what kind of entity it is. This can be regarded as a multi-classification problem, and all non-entities and different types of entities are regarded as different categories. Consider each word in the sentence as a node in the graph, and connect all the nodes according to the syntactic dependency tree to get a graph. To determine whether each word in the original sentence is an entity is to classify all nodes in the graph.

Training the graph neural network for NER prediction is defined as a supervised learning problem: $\mathcal{L} = \{(G_i, v_{i,j}, t_{i,j}) \mid G_i = (V_i, E_i) \in \mathcal{G}; v_{i,j} \in V_i; t_{i,j} \in \mathbb{R}^m,\ 1 \le i \le p, 1 \le j \le q_i\}$, where $v_{i,j} \in V_i$ denotes the j-th node in the node set $V_i \in \mathcal{V}$, $t_{i,j}$ denotes the one-hot vector of the target NER label for node $v_{i,j}$, m is the dimensionality of $t_{i,j}$, $p \le |\mathcal{G}|$ and $q_i \le |V_i|$.

We denote the parametric local transition function f_w to define the dependency of a node's state, x_v, on their neighborhoods, and denote the parametric local output function g_w to define a node's output o_v, just as follows:

$$\begin{aligned} x_v &= f_w(l_v, l_{co[v]}, x_{ne[v]}, l_{ne[v]}) \\ o_v &= g_w(x_v, l_v) \end{aligned} \tag{1}$$

where $l_v, l_{co[v]}, x_{ne[v]}, l_{ne[v]}$ respectively denote the node embedding of node n, the edge embeddings of its edges, the state embeddings and the node embeddings of its neighbor nodes, x_v and $o_v \in \mathbb{R}^s$, s is the dimensionality of the state embedding vector x_v and the output vector o_v. One thing worth noting is that the local transition function f_w and local output function g_w share the same parameters w.

With function f_w and function g_w, a mapping φ_w from $\mathcal{G} \times \mathcal{V}$ to $\mathbb{R}^m$ is defined. For the uni-directed graph $G_i = (V_i, E_i) \in \mathcal{G}$ transformed from a sentence's dependency parsing tree, a node $v_{i,j} \in V_i$ denoting a word in the sentence can be mapped to an output vector $o_{v_{i,j}}$, as $o_{v_{i,j}} = \varphi_w(G_i, v_{i,j})$, where a component in the output vector $o_{v_{i,j}}$ denotes the probability of the word belonging to an NER category, corresponding to its position. Thus training the graph neural network for NER label prediction is to estimate the parameter w, so that the mapping function φ_w can approximately approach the NER labels associated with the words contained in the NER training datasets, denoted as $L = \{(G_i,\ v_{i,j}, t_{i,j}) \mid G_i = (V_i, E_i) \in \mathcal{G}; v_{i,j} \in V_i;\ t_{i,j} \in \mathbb{R}^m, 1 \le i \le p, 1 \le j \le q_i\}$, where q_i is the number of nodes with NER labels in G_i, and we define the minimal optimization objective function of training the graph neural network as:

$$L_{gnn} = \sum_{i=1}^{p} \sum_{j=1}^{q_i} (t_{i,j} - \varphi_w(G_i, n_{i,j}))^2 \tag{2}$$

2) BiLSTMs: Two BiLSTM networks, called NER1 and NER2, are employed in information integration learning, and six BiLSTMS networks, called CWS and POS BiLSTMs, are employed in multi-contextualized character embedding learning. All these BiLSTM neural networks have similar structures. We define $k \in \{NER1, NER2\}$, $k' \in \{CWS1, CWS2, CWS3, POS1, POS2, POS3\}$. We denote the output of a BiLSTM from information integration learning as $\mathrm{P} = (p_1, p_2, \ldots, p_N)$, and the output of a BiLSTM from the multi-contextualized character embedding learning as $\mathrm{O} = (o_1, o_2, \ldots, o_N)$. Then for a sentence $\mathrm{x} = (c_1, c_2, \ldots, c_N)$ in the NER training corpus, with N denoting its length and $c_i \in \mathbb{R}^{d_h}$, we can compute its contextualized states from the BiLSTM neural network as:

$$\begin{aligned} \rightarrow \mathrm{h}_i &= \rightarrow LSTM\ (\rightarrow h_{i-1}, c_i) \\ \leftarrow \mathrm{h}_i &= \leftarrow LSTM\ (\leftarrow h_{i+1}, c_i) \\ \mathrm{h}_i &= \rightarrow h_i \oplus \leftarrow h_i \\ \mathrm{p}_i^k &= BiLSTM(c_i^k, p_{i-1}^k; \theta_k) \\ \mathrm{o}_i^{k'} &= BiLSTM(c_i^{k'}, o_{i-1}^{k'}; \theta_{k'}) \end{aligned} \tag{3}$$

where $\rightarrow h_i \in \mathbb{R}^{d_h}$ and $\leftarrow h_i \in \mathbb{R}^{d_h}$ denote the contextualized states of the forward and backward LSTM at position i, $\oplus$ denotes the concatenation operation, d_h is the dimensionality of h_i, θ_k and $\theta_{k'}$ respectively denote the parameters of the two different BiLSTMs from information integration learning and multi-contextualized character embedding learning.

3) Self-Attentions: We employ multiple multi-head self-attention mechanisms, namely SA-1, SA-2, SA-3, and SA-4, for feature extraction and information integration. All these self-attention mechanisms have similar structures, only that their inputs and outputs vary, where vectors from multiple sources are concatenated as an entire input vector. Without loss of generality, we take SA-1 as an example to illustrate how these self-attentions

work. The scaled dot-product attention [44] can be described as follows:

$$\mathrm{Attention}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{softmax}(\frac{\mathrm{QK}^T}{\sqrt{\mathrm{d}}})\mathrm{V} \tag{4}$$

where $\mathrm{Q} \in \mathbb{R}^{N\times 2d_h}$, $\mathrm{K} \in \mathbb{R}^{N\times 2d_h}$, and $\mathrm{V} \in \mathbb{R}^{N\times 2d_h}$ denote the query matrix, key matrix, and value matrix respectively. In line with the settings of [45], we set $\mathrm{Q} = \mathrm{K} = \mathrm{V} = \mathrm{P}$, and set the dimensionality of the hidden units contained in a BiLSTM neural network d equal to $2d_h$. The multi-head attention [44] first linearly projects the queries, keys, and values h times via using different linear projections, and the scaled dot-product attention is performed on h attention layers in parallel. Finally, results of scaled dot-product attention are concatenated and once again projected to get a new representation. Formally, multi-head attention can be defined as follows:

$$\begin{aligned} head_i &= \mathrm{Attention}(\mathrm{QW}_i^Q, \mathrm{KW}_i^K, \mathrm{VW}_i^V) \\ \mathrm{P}'_{(\mathrm{SA}-1)} &= \mathrm{Concat}(head_1, \ldots, head_h)\mathrm{W}_o \end{aligned} \tag{5}$$

where $\mathrm{W}_i^Q \in \mathbb{R}^{2d_h\times d_k}$, $\mathrm{W}_i^K \in \mathbb{R}^{2d_h\times d_k}$, $\mathrm{W}_i^V \in \mathbb{R}^{2d_h\times d_k}$ and $\mathrm{W_o} \in \mathbb{R}^{2d_h\times 2d_h}$ are trainable projection parameters, d_h is the dimensionality of values and d_k is the dimensionality of queries or keys in attention, and $d_k = d_h$.

For a sentence in the NER dataset, we compute its final representation via SA-4 as:

$$\begin{aligned} head_i &= \mathrm{Attention}(\mathrm{QW}_i^Q, \mathrm{KW}_i^K, \mathrm{VW}_i^V) \\ \mathrm{P}'_{(\mathrm{SA}-4)} &= \mathrm{Concat}(head_1, \ldots, head_h)\mathrm{W}_o \end{aligned} \tag{6}$$

where $\mathrm{W}_i^{'Q} \in \mathbb{R}^{6d_h\times d_k}$, $\mathrm{W}_i^{'K} \in \mathbb{R}^{6d_h\times d_k}$, $\mathrm{W}_i^{'V} \in \mathbb{R}^{6d_h\times d_k}$ and $\mathrm{W}'_{\mathrm{o}} \in \mathbb{R}^{6d_h\times 6d_h}$ are trainable projection parameters.

4) CRF Layers: Considering that NER, CWS, and POS tagging are sequence labeling tasks, we introduce a specific CRF layer for each task to mediate label predictions for different parts contained in a sentence instead of making label predictions independently. We take the CRF layer introduced for the NER prediction task in information integration learning as an example to illustrate how the CRF layers work. In addition, the output layer is the output of the model and the target layer is the label of the model in Fig. 2. Given a sentence $\mathrm{x} = (c_1, c_2, \ldots, c_L)$ with a predicted label sequence $\mathrm{y} = (y_1, y_2, \ldots, y_L)$, the CRF labeling process can be formalized as:

$$\begin{aligned} \mathrm{output}_i &= W_p p'_{i(SA-4)} + b_p \\ score(\mathrm{x},\mathrm{y}) &= \sum_{i=1}^{L}(\mathrm{output}_{i,y_i} + \mathrm{T}_{y_{i-1},y_i}) \\ \bar{\mathrm{y}} &= \arg\max_{\mathrm{y}\in\mathrm{Y_x}} score(\mathrm{x},\mathrm{y}) \end{aligned} \tag{7}$$

where $\mathrm{W}_p \in \mathbb{R}^{T\times 6d_h}$ and $\mathrm{b}_p \in \mathbb{R}^T$ are trainable parameters, T is the number of output labels, and output_{i,y_i} is the score of the y_i-th label of character c_i. Here M is a transition matrix that defines the scores of two successive labels, and $\mathrm{Y_x}$ denotes all candidate label sequences for a given sentence x. For decoding, we generate the predicted label sequence $\bar{\mathrm{y}}$ with the Viterbi algorithm [46].

D. Model Training

For model training, we exploit the negative log-likelihood objective as the loss function. Given S training samples $[(\mathrm{x}^{(i)};\hat{\mathrm{y}}^{(i)}), \ldots, (\mathrm{x}^{(S)};\hat{\mathrm{y}}^{(S)})]$ for a training task, we use gradient back-propagation to minimize the task's loss function, which is defined as follows:

$$\begin{aligned} p(\mathrm{x}|\hat{\mathrm{y}},\theta) &= \frac{e^{score(\mathrm{x},\hat{\mathrm{y}})}}{\sum_{\tilde{\mathrm{y}}\in\mathrm{Y_x}} e^{score(\mathrm{x}\tilde{\mathrm{y}})}} \\ L_{Task} &= -\sum_{i=1}^{S} \log p(\mathrm{x}^{(i)}|\hat{\mathrm{y}}^{(i)},\theta) \end{aligned} \tag{8}$$

where $p(\mathrm{x}|\hat{\mathrm{y}},\theta)$ is the probability of the ground-truth label sequence $\hat{\mathrm{y}}$ and θ denotes the CRF parameters. The final objective function of SDI-NER can be formulated as:

$$\begin{aligned} L = L_{NER} + L_{CWS1} + L_{CWS2} + L_{CWS3} + L_{POS1} \\ + L_{POS2} + L_{POS3} \end{aligned} \tag{9}$$

where L_{NER}, L_{CWS1}, L_{CWS2}, L_{CWS3}, L_{POS1}, L_{POS2}, and L_{POS3} can be computed via Eq.8.

At each iteration of model training, we sample a batch of training instances from the NER dataset for loss evaluations and update all trainable parameters. We use the Adam algorithm [60] to optimize the final objective function.

IV. Experimental Evaluations

We perform extensive experiments to evaluate the Chinese NER performance of SDI-NER, and compare its performance with the mainstream baseline NER models. We present the experimental settings and findings in the section.

A. Evaluation Datasets

We evaluate all models' NER performances on three public datasets, including the Weibo NER, OntoNotes4, and Chinese Resume datasets. The Weibo NER dataset [47] consists of annotated NER messages collected from Sina Weibo and is annotated with four kinds of entity labels: PER, LOC, ORG and GEO (Geopolitical), including named entities and nominal entities. OntoNotes4 [61] is a manually annotated multilingual corpus in the news domain containing various text annotations, including four kinds of named entities: PER, ORG, LOC, and GPE(Geographical Administrative Entities). The Chinese Resume dataset [50] is composed of resumes collected from Sina Finance, and it is annotated with eight kinds of named entities: CONT (Country), EDU (Educational Institution), LOC, PER, ORG, PRO (Profession), RACE (Ethnicity/Background) and TITLE (Job Title). Statistics about these datasets are presented in Table I.

We employ multiple open-source toolkits to generate CWS and POS tagging labels for the NER datasets, to learn multiple kinds of contextualized character embeddings from CWS and POS tagging tasks. For each NER dataset, we have "Thulac CWS1" and "Thulac POS1" labels, "Pyltp CWS1" and "Pyltp POS1" labels, and "Jieba CWS1" and "Jieba POS1" labels, and we illustrate how sentences are annotated in Fig. 4. When

TABLE I
STATISTICS OF THE DATASETS

Dataset	Train sent	Dev sent	Test sent
Weibo NER	1350	270	270
OntoNotes4	15724	4301	4346
Chinese Resume	3821	463	477

NER Data	Sentence	新华社			北京		二月		十一日			电	(	记者		唐虹		)	
	(word)	Xinhua	News	Agency,	Beijing,		February					11th		(Reporter		Tang		Hong)	
	Sentence	新	华	社	北	京	二	月	十	一	日	电	(	记	者	唐	虹	)	
	(character)	Xinhua	News	Agency,	Beijing,		February					11th		(Reporter		Tang		Hong)	
	NER Label	B-ORG	M-ORG	E-ORG	B-GPE	E-GPE	O	O	O	O	O	O	O	O	O	B-PER	E-PER	O	
	DP Label	1			2		0		0			0	0	0		3		0	
	CWS	1	2	3	1	3	1	3	1	2	3	0	0	1	3	1	3	0	
	POS	B-ni	I-ni	E-ni	B-ns	E-ns	B-t	E-t	B-t	I-t	E-t	S-v	S-wp	B-n	E-n	B-np	E-np	S-wp	

Fig. 4. Example labels of the selected sentence for various tasks, where "DP Label" denotes the NER category labels that a node could be classified with in graph neural network, "0" means a character is a word, "1" means the beginning of a word, "3" means the end of a word, "2" means the middle part of a word. "BIOES" annotation method is used for NER and POS tagging labels. In the POS tagging, "ni" is the organization name, "ns" is the geographical name, "t" is the temporal word, "v" is the verb, "wp" is the punctuation, "n" is the general noun, and "np" is the person name.

using the CWS information training model, we need to know how the word is segmented, "0" means a character is a word, "1" means the beginning of a word, "3" means the end of a word, "2" means the middle part of a word. The word segmentation information can be represented very well by using "0123," so there is no such complicated representation method as "BIOES".

B. Parameter Settings

We adjust hyperparameters according to the NER performance achieved in information integration learning on the development set. In the information integration learning and the syntactic dependency information learning, the initial learning rates are set to 0.001 and 0.01 respectively. In all datasets, the dimensionality of BiLSTM contextualized states d_h is set to 120, the number of self-attention units and heads are set to 240 and 10. To avoid overfitting, we set the dropout rate to 0.3, and the training batch size to 80 on three evaluation datasets. The embeddings that we used in the experiments are pre-trained. For the three public datasets, the same embeddings as [50] are used in this paper, training word2vec [62] embeddings over the Chinese Giga-Word dataset [63], with about 704,400 words in the final lexicon. Character and character bigram embeddings are also pre-trained on the Chinese Giga-Word dataset, using word2vec. We use precision(P), recall(R) and F1-score as the performance evaluation metrics.

C. Baseline Models

We compare our model with the following 20 open-source NER models on the three public evaluation datasets. As these baseline models were proposed at different times, different models are applicable on different datasets. The Chinese Resume dataset released in 2018 is only applicable for nine baseline models, rather than all the 20 baseline models. So we compare our NER model with various numbers of baseline models on various datasets in our experiments.

- [57]: which formulates the problem of exploring signals on annotated bilingual texts as a simple ILP (Integer Linear Program) problem and encourages entity labels to agree with bilingual constraints.
- [58]: which encourages both cross-language and intra-document consistency with a factored probabilistic sequence model.
- [47]: which uses both labeled and unlabeled raw text with a joint NER training objective for word embeddings.
- [48]: which incorporates word boundary information into an NER system for Chinese social media.
- [25]: which directly trains on F1-Score rather than label accuracy, and trains on both F1-Score and label accuracy in an integrated fashion.
- [49]: which learns in-domain un-annotated texts by self-training with a confidence-based semi-supervised function.
- [59]: which investigates the effect of discrete and neural feature combinations for a range of fundamental NLP tasks based on sequence labeling.
- [45]: which makes full use of task-shared boundary information, and prevents the task-specific features of CWS in a novel adversarial transfer learning framework.
- Lattice [50]: which encodes a sequence of input characters, as well as all potential words that match a lexicon in a lattice-structured LSTM model, for Chinese NER.
- CAN-NER [28]: which first combines CNN with the local-attention mechanism to enhance the ability of the NER model via implicitly capturing local context relationships existing among character sequences.
- WC-LSTM [27]: which adds word information into the starting or the ending characters of words, and alleviates the influence of word segmentation errors while obtaining the word boundary information in a word-character LSTM model.

TABLE II
MODEL PERFORMANCE ON THE WEIBO NER DATASET

Models	Named Entity			Named Mention			Overall
	P(%)	R(%)	F1 (%)	P(%)	R (%)	F1 (%)	F1 (%)
[47]	74.78	39.81	51.96	71.92	53.03	61.05	56.05
[48]	66.67	47.22	55.28	74.48	54.55	62.97	58.99
[25]	66.93	40.67	50.60	66.46	53.57	59.32	54.82
[49]	61.68	48.82	54.50	74.13	53.54	62.17	58.23
[45]	59.51	50.00	54.34	71.43	47.90	57.35	58.70
Lattice [50]	-	-	53.04	-	-	62.25	58.79
CAN-NER [28]	-	-	55.38	-	-	62.98	59.31
WC-LSTM [27]	-	-	52.55	-	-	67.41	59.84
LGN [34]	-	-	55.34	-	-	64.98	60.21
Lexicon [35]	67.31	48.61	56.45	75.15	62.63	68.32	63.09
[51]	-	-	63.10	-	-	56.30	59.50
TENER [52]	-	-	-	-	-	-	58.17
LR-CNN [53]	-	-	57.14	-	-	66.67	59.92
PLTE [54]	-	-	62.21	-	-	49.54	55.15
FLAT [55]	-	-	-	-	-	-	63.42
SoftLexicon (LSTM) [56]	-	-	59.08	-	-	62.22	61.42
MECT [31]	-	-	61.91	-	-	62.51	63.30
SDI-NER	73.61	54.08	62.94	77.87	54.91	64.40	**63.65**

- LGN [34]: which uses a lexicon to construct a graph neural network and implements Chinese NER as a graph node classification task.
- Lexicon [35]: which directly integrates lexical knowledge efficiently for Chinese NER in a Collaborative Graph Network.
- [51]: which proposes a novel approach based on graph neural networks with a multidigraph structure that captures the information that the gazetteers offer.
- TENER [52]: which adopts adapted Transformer Encoder to model the character-level features and word-level features.
- LR-CNN [53]: which proposes a novel CNN structure for incorporating lexicons into Chinese NER and effectively accelerates the model training by its use of parallelism.
- PLTE [54]: which models all the characters and matched lexical words in parallel with batch processing and augments self-attention with positional relation representations to incorporate lattice structure.
- FLAT [55]: which converts the lattice structure into a flat structure consisting of spans and each span corresponds to a character or latent word and its position in the original lattice.
- SoftLexicon (LSTM) [56]: which proposes a simple but effective method for incorporating word lexicons into the character representations for Chinese NER.
- MECT [31]: which improves the performance of Chinese NER by fusing the structural information of Chinese characters.

D. Overall Performance

Table II presents the experimental results for NER, achieved by our model and other baseline models on the Weibo NER dataset, where "NE," "NM" and "Overall" respectively indicate F1-scores for identifying named entities, named mention (excluding named entities) and both. Named entity refers to which type of entity. Named mention is a rough classification, we know that the word is an entity, but we don't know what kind of entity the word is. We compare our models with three multi-task learning models [45], [47], [48], and two semi-supervised learning models [25], [49], on the Weibo NER dataset. Besides, eleven recent models [27], [28], [34], [35], [50]–[56] are also compared with our model. We can see that the FLAT model [55] and the Lexicon model [31] achieve the highest and second-highest F1-scores in all baseline models, which are 63.42% and 63.30% respectively, while SDI-NER improves the overall F1-score to 63.65%.

Table III presents the experimental results for NER, achieved by various models on the OntoNotes4 dataset. [57] and [58] reach F1-scores of 75.02% and 74.32%, and [31] achieves an F1-score of 76.92%, which is the highest F1-score in the baseline models. [50] proposes a lattice LSTM to exploit word information in character sequence, achieving an F1-score of 73.88%. [28] achieves an F1-score of 73.64% among character-based models without using external data. [27] achieves an F1-score of 74.43% because of using the self-attention strategy. Two recent NER models [31], [56] are also compared with SDI-NER, and we can see that our model achieves the highest F1-score of 76.95%.

Table III presents the results on the Chinese Resume dataset. [50] releases this dataset and achieves an F1-score of 94.46%. The F1-score of CAN-NER [28] is 94.94% and that of WC-LSTM [27] is 95.21%. Moreover, unlike previous models, LGN model [34] integrates lexicon information into the graph neural network more efficiently, and achieves an F1-score of 95.37%. The model MECT [31] achieve the highest F1-scores in all baseline models because of fusing the structural information of Chinese characters. Consistently with observations on the Weibo NER dataset and OntoNotes4 dataset, SDI-NER achieves the highest F1-score of 95.95%.

The performance that SDI-NER achieves outperforming all baselines in most scenarios validates the effectiveness of three basic assumptions contained in our model: learning information about syntactic dependencies between words could help recognize named entities; employing information learned from CWS

TABLE III
MODEL PERFORMANCE ON THE ONTONOTES DATASET AND CHINESE RESUME DATASET

dataset	OntoNotes			Chinese Resume		
Models	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
[57]	77.71	72.51	75.02	-	-	-
[58]	76.43	72.32	74.32	-	-	-
[59]	72.98	80.15	76.40	-	-	-
Lattice [50]	76.35	71.56	73.88	94.81	94.1	94.46
CAN-NER [28]	75.05	72.29	73.64	95.05	94.82	94.94
WC-LSTM [27]	76.09	72.85	74.43	95.27	95.15	95.21
LGN [34]	76.13	73.68	74.89	95.28	95.46	95.37
Lexicon [35]	75.06	74.52	74.79	-	-	-
[51]	75.40	76.60	76.00	-	-	-
TENER [52]	-	-	72.43	-	-	95.00
LR-CNN [53]	76.40	72.60	74.45	95.37	94.84	95.11
PLTE [54]	76.78	72.54	74.60	95.34	95.46	95.40
FLAT [55]	-	-	76.45	-	-	95.45
SoftLexicon (LSTM) [56]	77.28	74.07	75.64	95.30	95.77	95.53
MECT [31]	77.57	76.27	76.92	96.50	95.39	95.89
SDI-NER	76.96	76.94	**76.95**	96.04	95.86	**95.95**

TABLE IV
MODEL PERFORMANCES OF USING BERT EMBEDDINGS ON THE WEIBO NER DATASET, ONTONOTES4 DATASET AND CHINESE RESUME DATASET

Dataset	Models	Embeddings	NE	NM	Overall
Weibo NER	Lattice [50]	BERT	53.63	58.86	56.21
	SoftLexicon (LSTM) [56]	BERT	65.77	62.05	63.80
	MECT [31]	BERT	-	-	68.20
	SDI-NER	BERT	66.78	69.85	**68.28**
	SDI-NER	word2vec	62.94	64.40	**63.65**
Dataset	Models	Embeddings	P	R	F1
OntoNotes	Lattice [50]	BERT	75.26	70.71	72.92
	SoftLexicon (LSTM) [56]	BERT	76.01	79.96	77.93
	MECT [31]	BERT	-	-	80.14
	SDI-NER	BERT	82.14	78.39	**80.22**
	SDI-NER	word2vec	76.96	76.94	**76.95**
Resume	Lattice [50]	BERT	94.47	94.36	94.41
	SoftLexicon (LSTM) [56]	BERT	94.87	96.50	95.68
	MECT [31]	BERT	-	-	95.53
	SDI-NER	BERT	96.61	95.54	**96.07**
	SDI-NER	word2vec	96.04	95.86	**95.95**

and POS tagging could help identify word and entity boundaries; employing self-attention mechanisms could help capture key features from various kinds of contextualized character and word embeddings, and thus identify entity boundaries and improve the overall NER performance. Compared with other baseline models, we find that SDI-NER achieved a minor performance improvement on the Chinese Resume dataset. The possible reason may be that the features extracted from CWS and POS tagging tasks exert a more positive influence on this dataset. What's more, the above experimental results strongly verify that utilizing word and character information in our model is more effective than other models.

E. Model Performances With BERT Embeddings

This experiment verifies the performance our model achieves with the BERT embeddings, as BERT has been widely used for various NLP tasks. As Lattice [50] is the Chinese NER method based on character embedding proposed in 2018, and it has been widely used and improved in the past four years. Model SoftLexicon (LSTM) [56] and model MECT [31] are the models that have achieved the best results recently. We use these as the baseline models for evaluating the NER performance of our model achieved with the BERT embeddings. In addition, Table IV presents the experimental results performed by the four models with BERT embeddings on three evaluation datasets. We can see that our model outperforms Lattice on all datasets, especially on the Weibo NER dataset, where the F1-score is improved by over 8%. Compared with the latest two models SoftLexicon (LSTM) [56] and MECT [31], when both BERT word embeddings are used, our model is still better than these two models. The results indicates that our model is effective, whether it is using BERT word embeddings or not.

F. Ablation Experiments

To illustrate how the performance of our model is improved with various components contained in SDI-NER, we divide the whole model into the following four parts: syntactic dependency information learning part (Part I), the multi-contextualized character embedding learning part (Part II), information integration learning part (Part III) and the multiple self-attention parts in information integration learning (Part IV), and perform ablation experiments to evaluate the performance achieved by the model without some parts of Parts I, III and IV, on the three datasets. The experimental results achieved on different datasets are presented in Table V. We can see that removing different parts from our proposed model will induce performance decreases

TABLE V
MODEL PERFORMANCE ACHIEVED ON SDI-NER WITHOUT CORRESPONDING PARTS ON THE WEIBO NER DATASET, ONTONOTES4 DATASET AND CHINESE RESUME DATASET

Weibo NER Dataset			
Models	NE	NM	Overall
SDI-NER	62.94(0.00)	64.40(0.00)	63.65(0.00)
w/o Part IV	62.31(-0.63)	63.71(-0.69)	63.26(-0.39)
w/o Part II + IV	62.66(-0.26)	63.08(-1.32)	62.85(-0.70)
w/o Part III + IV	62.83(-0.11)	63.79(-0.61)	63.36(-0.29)
w/o Part I + IV	62.53(-0.41)	63.25(-1.15)	63.07(-0.58)
w/o Part II + III + IV	62.09(-0.85)	63.64(-0.76)	63.32(-0.33)
w/o Part I + II + IV	62.21(-0.73)	63.49(-0.91)	63.21(-0.44)
OntoNotes Dataset			
Models	P	R	F1
SDI-NER	76.96(0.00)	76.94(0.00)	76.96(0.00)
w/o Part IV	74.43(-2.53)	73.29(-3.65)	73.86(-3.10)
w/o Part II + IV	74.53(-2.43)	73.23(-3.61)	73.87(-3.09)
w/o Part III + IV	73.22(-3.74)	73.37(-3.57)	73.29(-3.67)
w/o Part I + IV	73.51(-3.45)	72.53(-4.41)	73.02(-3.94)
w/o Part II + III + IV	72.38(-4.57)	72.29(-4.65)	72.33(-4.63)
w/o Part I + II + IV	73.97(-2.99)	73.43(-3.51)	73.70(-2.26)
Resume Dataset			
Models	P	R	F1
SDI-NER	96.04(0.00)	95.86(0.00)	95.95(0.00)
w/o Part IV	94.75(-1.29)	94.61(-1.25)	94.68(-1.27)
w/o Part II + IV	94.59(-1.45)	94.74(-1.12)	94.66(-1.29)
w/o Part III + IV	94.89(-1.15)	94.39(-1.47)	94.64(-1.31)
w/o Part I + IV	94.61(-1.43)	94.36(-1.50)	94.48(-1.47)
w/o Part II + III + IV	94.21(-1.83)	94.23(-1.63)	94.22(-1.73)
w/o Part I + II + IV	95.16(-0.88)	94.19(-1.67)	94.67(-1.28)

TABLE VI
MODEL PERFORMANCE ACHIEVED ON SDI-NER WITHOUT USING THULAC TOOLKIT, PYLTP TOOLKIT AND JIEBA TOOLKIT ON THE WEIBO NER DATASET, ONTONOTES4 DATASET AND CHINESE RESUME DATASET

Weibo NER Dataset			
Models	NE	NM	Overall
SDI-NER	62.94(0.00)	64.40(0.00)	63.65(0.00)
w/o Part I	62.52(-0.42)	63.82(-0.58)	63.49(-0.16)
w/o Part II	62.56(-0.38)	63.81(-0.59)	63.51(-0.14)
w/o Part III	62.53(-0.41)	63.90(-0.50)	63.47(-0.18)
w/o Part I + II	62.33(-0.61)	63.77(-0.63)	63.39(-0.26)
w/o Part I + III	62.39(-0.55)	63.72(-0.68)	63.35(-0.30)
w/o Part II + III	62.36(-0.58)	63.74(-0.66)	63.36(-0.29)
OntoNotes Dataset			
Models	P	R	F1
SDI-NER	76.96(0.00)	76.94(0.00)	76.96(0.00)
w/o Part I	76.01(-0.95)	74.50(-2.44)	75.25(-1.71)
w/o Part II	75.93(-1.03)	74.53(-2.41)	75.22(-1.74)
w/o Part III	75.98(-0.98)	74.46(-2.48)	75.21(-1.75)
w/o Part I + II	75.14(-1.82)	73.62(-3.32)	74.37(-2.59)
w/o Part I + III	75.11(-1.85)	73.67(-3.27)	74.38(-2.58)
w/o Part II + III	75.07(-1.89)	73.56(-3.38)	74.31(-2.65)
Resume Dataset			
Models	P	R	F1
SDI-NER	96.04(0.00)	95.86(0.00)	95.95(0.00)
w/o Part I	95.07(-0.97)	95.17(-0.69)	95.12(-0.83)
w/o Part II	95.04(-1.00)	95.19(-0.67)	95.11(-0.84)
w/o Part III	95.06(-0.98)	95.15(-0.71)	95.10(-0.85)
w/o Part I + II	94.89(-1.15)	94.90(-0.96)	94.89(-1.06)
w/o Part I + III	94.91(-1.13)	94.93(-0.93)	94.92(-1.03)
w/o Part II + III	94.88(-1.16)	94.92(-0.94)	94.90(-1.05)

with different extents, and this indicates that each of the parts contained in our proposed model contributes to improving the overall performance for NER identification.

In the part of Multi-contextualized Character Embedding Learning, when we do not use a toolkit or only use one toolkit and two toolkits, the extracted features do not contain enough information. If we use too many, the model is too complex, and the training time is too long. To illustrate how the performance of our model is improved with three POS taggers and three word segmentation toolkits in SDI-NER, we divide the three POS taggers and three word segmentation toolkits into the following three parts: "Thulac CWS1" and "ThulacPOS1" (Part I), "Pyltp CWS" and "Pyltp POS1" (Part II) and "Jieba CWS1" and "Jieba POS1" (Part III), and perform ablation experiments to evaluate the performance achieved by the model without some parts of Parts I, II and III on the three datasets. The experimental results are shown in Table VI. We can see that when using one POS tagger and one word segmentation toolkit respectively, the results are not much different. When using two POS taggers and two word segmentation toolkits respectively, the results are not much different. Still, both are better than using one POS tagger and one word segmentation toolkit. When using

TABLE VII
MODEL PERFORMANCE ACHIEVED ON SDI-NER WITHOUT CWS AND POS ON THE WEIBO NER DATASET, ONTONOTES4 DATASET AND CHINESE RESUME DATASET

Weibo NER Dataset			
Models	NE	NM	Overall
SDI-NER	62.94(0.00)	64.40(0.00)	63.65(0.00)
w/o Part I	62.25(-0.69)	63.56(-0.84)	62.87(-0.78)
w/o Part II	62.31(-0.63)	63.52(-0.88)	62.89(-0.76)
w/o Part I + II	62.05(-0.89)	63.21(-1.19)	62.61(-1.04)
OntoNotes Dataset			
Models	P	R	F1
SDI-NER	76.96(0.00)	76.94(0.00)	76.95(0.00)
w/o Part I	75.53(-1.43)	74.32(-2.62)	74.92(-2.03)
w/o Part II	75.42(-1.54)	74.28(-2.66)	74.85(-2.10)
w/o Part I + II	75.07(-1.89)	73.85(-3.09)	74.46(-2.49)
Resume Dataset			
Models	P	R	F1
SDI-NER	96.04(0.00)	95.86(0.00)	95.95(0.00)
w/o Part I	94.97(-1.07)	95.06(-0.80)	95.01(-0.94)
w/o Part II	94.92(-1.12)	95.08(-0.78)	95.00(-0.95)
w/o Part I + II	94.76(-1.28)	94.97(-0.89)	94.86(-1.09)

Sentence	无	锡	威	孚	高	科	技	集	团	股	份	有	限	公
	司	第	四	届	、	第	五	届	监	事	会	主	席	,
	Chairman	of	the	Fourth	and	Fifth	Supervisory	Board	of	Wuxi	Weifu	High-Tech	Group	Co,
Labels/SDI-NER	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG
	E-ORG	B-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	E-TITLE	O
w/o Part IV	B-LOC	E-LOC	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG
	E-ORG	O	O	O	O	O	O	O	B-TITLE	M-TITLE	M-TITLE	M-TITLE	E-TITLE	O
w/o Part I+IV	B-LOC	E-LOC	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG
	E-ORG	B-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	E-TITLE	O
w/o Part II+IV	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	E-ORG	O	O	O	O	O
	O	B-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	E-TITLE	O
w/o Part III+IV	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	E-ORG	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG
	E-ORG	O	O	O	O	B-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	E-TITLE	O
w/o Part I+II+IV	B-LOC	E-LOC	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG
	E-ORG	O	O	O	O	O	O	O	B-TITLE	M-TITLE	M-TITLE	M-TITLE	E-TITLE	O
w/o Part I+III+IV	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	M-ORG	E-ORG	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG
	E-ORG	B-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	M-TITLE	E-TITLE	O

Fig. 5. NER labels predicted by our model with various compositions.

three POS taggers and three word segmentation toolkits, the results are the best. It can be seen from the experimental results that it is necessary to adopt three POS taggers and three word segmentation toolkits, because it can improve the accuracy of the experiment.

To illustrate the effect of CWS and POS tagging information on our model, we divide CWS and POS into Part I and Part II, and remove one or both parts for the experiments. The experimental results are shown in Table VII. The result shows that the model accuracy decreases when removing CWS or POS for experiments, which means both parts are practical. The accuracy decreases more when both parts are removed, which shows that adding CWS information and POS information is helpful to train the model.

G. Case Study

Finally, we illustrate how the final predicted NER results are influenced by each part of SDI-NER with a case study. We perform NER identification on a sentence chosen from the Chinese Resume dataset with SDI-NER containing various components. The predicted NER labels are presented in Fig. 5, where wrong predictions are marked in yellow. The entities that need to be correctly identified are "无锡威孚高科技集团股份有限公司" and "第四届、第五届监事会主席". The "无锡威孚高科技集团股份有限公司" is correctly identified only by SDI-NER, and the second "第四届、第五届监事会主席" is correctly identified by SDI-NER and SDI-NER without Part I+IV, Part II+IV, and Part I+III+IV. We can see that each part plays its role for NER, and removing any part from the model would eventually influence the overall NER performance, resulting in incorrect NER label predictions.

V. CONCLUSION

In this paper, we propose a novel framework to learn from the large-scale syntactic dependency graph for improving Chinese named entity recognition based on graph neural network. In particular, we learn multiple kinds of contextualized character embeddings from multiple CWS and POS tagging tasks to help identify entity boundaries. We design multiple self-attention components to extract key features from the learned information for NER prediction. Extensive experiments show that our proposed method achieves at least as well as the state-of-the-art baselines. The result proves the effectiveness of SDI-NER in learning good representations from informative syntactic graphs for named entity recognition.

In addition, because many datasets are used to construct a large-scale syntactic dependency graph, the training speed of the model is slow. In the future, we will study how to reduce the model parameters and increase the training speed while keeping the experimental results basically unchanged.

REFERENCES

[1] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proc. Annu. Conf. Assoc. Comput. Linguistics*, 2015, pp. 167–176.

[2] E. M. Voorhees *et al.*, "TREC: Experiment and evaluation in information retrieval," *J. Assoc. Informat. Sci. Technol.*, vol. 32, no. 4, pp. 563–567, 2006.

[3] A. Fader, L. Zettlemoyer, and O. Etzioni, "Paraphrase-driven learning for open question answering," in *Proc. Assoc. Comput. Linguistics*, 2013, pp. 1608–1618.

[4] X. Yao and B. Van Durme, "Information extraction over structured data: Question answering with freebase," in *Proc. Assoc. Comput. Linguistics*, 2014, pp. 956–966.

[5] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2005, pp. 724–731.

[6] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *Proc. Assoc. Comput. Linguistics*, 2016, pp. 1105–1116.

[7] S. Singh, S. Riedel, B. Martin, J. Zheng, and A. McCallum, "Joint inference of entities, relations, and coreference," in *Proc. Workshop Automated Knowl. Base Construction*, 2013, pp. 1–6.

[8] S. Han, X. Hao, and H. Huang, "An event-extraction approach for business analysis from online chinese news," *Electron. Commerce Res. Appl.*, vol. 28, pp. 244–260, 2018.

[9] D. Cheng, Z. Niu, and L. Zhang, "Delinquent events prediction in temporal networked-guarantee loans," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2020.3027346.

[10] C. Yubo *et al.*, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proc. Annu. Conf. Assoc. Comput. Linguistics*, 2015, pp. 167–176.

[11] S. Upadhyay, N. Gupta, and D. Roth, "Joint multilingual supervision for cross-lingual entity linking," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 2486–2495.

[12] J. Xin, Y. Lin, Z. Liu, and M. Sun, "Improving neural fine-grained entity typing with knowledge attention," in *Proc. Conf. Assoc. Adv. Artif. Intell.*, 2018, pp. 1–8.

[13] Y. Nie, Y. Tian, Y. Song, X. Ao, and X. Wan, "Improving named entity recognition with attentive ensemble of syntactic information," in *Proc. Assoc. Comput. Linguistics*, 2020, pp. 1–15.

[14] M. Zhang, J. Zhang, and J. Su, "Exploring syntactic features for relation extraction using a convolution tree kernel," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2006, pp. 288–295.

[15] G. Zhou, L. Qian, and J. Fan, "Tree kernel-based semantic relation extraction with rich syntactic and semantic information," *Inf. Sci.*, vol. 180, pp. 1313–1325, 2010.

[16] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: A High-Performance Learning Name-Finder," in *Proc. 5th Conf. Appl. Natural Lang. Process.*, 1998, pp. 194–201.

[17] H. Isozaki and H. Kazawa, "Efficient support vector classifiers for named entity recognition," in *Proc. Int. Conf. Comput. Linguistics*, 2002, pp. 1–7.

[18] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Mach. Learn.*, 2001, pp. 282–289.

[19] 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*, 2016, pp. 260–270.

[20] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar, "Deep active learning for named entity recognition," in *Proc. 2nd Workshop Representation Learn. NLP*, 2017, pp. 252–256.

[21] P. Zhu, D. Cheng, F. Yang, Y. Luo, W. Qian, and A. Zhou, "ZH-NER: Chinese named entity recognition with adversarial multi-task learning and self-attentions," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2021, pp. 603–611.

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

[23] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," in *Proc. Trans. Assoc. Comput. Linguistics*, 2016, pp. 357–370.

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

[25] H. He and X. Sun, "F-score driven max margin neural network for named entity recognition in chinese social media," in *Proc. Eur. Chapter Assoc. Comput. Linguistics*, 2016, pp. 713–718.

[26] C. Wang, W. Chen, and B. Xu, "Named entity recognition with gated convolutional neural networks," in *Proc. China Nat. Conf. Chin. Comput. Linguistics*, 2017, pp. 110–121.

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

[28] Y. Zhu and G. Wang, "CAN-NER: Convolutional attention network forchinese named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 3384–3393.

[29] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli, "Cloze-driven pretraining of self-attention networks," in *Proc. Conf. Assoc. Adv. Artif. Intell.*, 2019, pp. 3216–3222.

[30] T. Liu, J.-G. Yao, and C.-Y. Lin, "Towards improving neural named entity recognition with gazetteers," in *Proc. Conf. Empir. Methods Natural Lang. Joint Conf. Natural Lang. Process.*, 2019, pp. 5360–5369.

[31] S. Wu, X. Song, and Z. Feng, "Mect: Multi-metadata embedding based cross-transformer for chinese named entity recognition," in *Proc. Assoc. Comput. Linguistics*, 2021, pp. 1–11.

[32] S. Zhao, M. Hu, Z. Cai, H. Chen, and F. Liu, "Dynamic modeling cross- and self-lattice attention network for chinese NER," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2021, pp. 14 515–14 523.

[33] A. Cetoli, S. Bragaglia, A. D. O'Harney, and M. Sloan, "Graph convolutional networks for named entity recognition," in *Proc. Int. Conf. Web Stud.*, 2017, pp. 37–45.

[34] T. Gui *et al.*, "A lexicon-based graph neural network for chinese NER," in *Proc. Conf. Empir. Methods Natural Lang. Joint Conf. Natural Lang. Process.*, 2019, pp. 1039–1049.

[35] D. Sui, Y. Chen, K. Liu, J. Zhao, and S. Liu, "Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network," in *Proc. Conf. Empir. Methods Natural Lang. Joint Conf. Natural Lang. Process.*, 2019, pp. 3821–3831.

[36] C. Chen and F. Kong, "Enhancing entity boundary detection for better chinese named entity recognition," in *Proc. Assoc. Comput. Linguistics*, 2021, pp. 20–25.

[37] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," in *Proc. Assoc. Comput. Linguistics*, 2020, pp. 1–7.

[38] T. Dozat and C. D. Manning, "Deep biaffine attention for neural dependency parsing," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–8.

[39] Y. Nie, Y. Tian, X. Wan, Y. Song, and B. Dai, "Named entity recognition for social media texts with semantic augmentation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2020, pp. 1–9.

[40] T. Qian, M. Zhang, Y. Lou, and D. Hua, "A joint model for named entity recognition with sentence-level entity type attentions," *IEEE/ACM Trans. Audio Speech Lang.*, pp. 1438–1448, 2021.

[41] J. Parker and S. Yu, "Named entity recognition through deep representation learning and weak supervision," in *Proc. Assoc. Comput. Linguistics*, 2021, pp. 3828–3839.

[42] W. Che, Z. Li, and T. Liu, "LTP: A chinese language technology platform," in *Proc. Int. Conf. Comput. Linguistics*, 2010, pp. 13–16.

[43] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, 2008.

[44] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[45] P. Cao, Y. Chen, K. Liu, J. Zhao, and S. Liu, "Adversarial transfer learning for chinese named entity recognition with self-attention mechanism," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 182–192.

[46] G. D. Forney, "The viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

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

[48] N. Peng and D. Mark, "Improving named entity recognition for Chinese social media with word segmentation representation learning," in *Proc. Assoc. Comput. Linguistics*, 2016, pp. 149–155.

[49] H. He and X. Sun, "A unified model for cross-domain and semi-supervised named entity recognition in Chinese social media," in *Proc. Conf. Assoc. Adv. Artif. Intell.*, 2017, pp. 3216–3222.
[50] Y. Zhang and J. Yang, "Chinese NER using lattice LSTM," in *Proc. Assoc. Comput. Linguistics*, 2018, pp. 1554–1564.
[51] R. Ding, P. Xie, X. Zhang, W. Lu, L. Li, and L. Si, "A neural multi-digraph model for chinese NER with gazetteers," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 1462–1467.
[52] H. Yan, B. Deng, X. Li, and X. Qiu, "Tener: Adapting transformer encoder for name entity recognition," 2019. [Online]. Available: https://arxiv.org/abs/1911.04474
[53] T. Gui, R. Ma, Q. Zhang, L. Zhao, Y.-G. Jiang, and X. Huang, "CNN-based Chinese NER with lexicon rethinking," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4982–4988.
[54] X. Mengge, Y. Bowen, L. Tingwen, W. Bin, M. Erli, and L. Quangang, "Porous lattice-based transformer encoder for chinese NER," in *Proc. Int. Conf. Comput. Linguistics*, 2019, pp. 3831–3841.
[55] X. Li, H. Yan, X. Qiu, and X. Huang, "Flat: Chinese ner using flat-lattice transformer," in *Proc. Assoc. Comput. Linguistics*, 2020, pp. 6836–6842.
[56] M. Peng, R. Ma, Q. Zhang, and X. Huang, "Simplify the usage of lexicon in chinese NER," in *Proc. Assoc. Comput. Linguistics*, 2020, pp. 5951–5960.
[57] W. Che, M. Wang, C. D. Manning, and T. Liu, "Named entity recognition with bilingual constraints," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2013, pp. 52–62.
[58] M. Wang, W. Che, and C. D. Manning, "Effective bilingual constraints for semi-supervised learning of named entity recognizers," in *Proc. Conf. Assoc. Adv. Artif. Intell.*, 2013, pp. 919–925.
[59] J. Yang, Z. Teng, M. Zhang, and Y. Zhang, "Combining discrete and neural features for sequence labeling," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*, 2016, pp. 140–154.
[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[61] R. Weischedel *et al.*, "Ontonotes release 4.0," Feb. 2011. [Online]. Available: https://doi.org/10.35111/gfjf-7r50
[62] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[63] R. Parker, D. Graff, K. Chen, J. Kong, and K. Maeda, "Chinese gigaword fifth edition (LDC2011t13)," Nov. 2011. [Online]. Available: https://doi.org/10.35111/102m-dr17

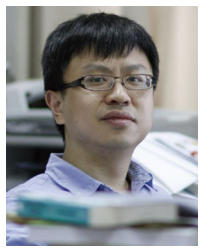
Peng Zhu received the bachelor's degree from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2014. He is currently a working toward the Ph.D. degree with the School of Data Science and Engineering, East China Normal University, Shanghai, China. His research interests include knowledge engineering, natural language processing, and deep learning.

Dawei Cheng received the Ph.D. Degree in computer science from Shanghai Jiao Tong University, Shanghai, China. He was a Postdoctoral Associate with the MoE Key Lab of Artificial Intelligence, Department of Computer Science, Shanghai Jiao Tong University. He is currently an Assistant Professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. His research interests include graph learning, data mining, and machine learning.

Fangzhou Yang received the B.Sc. degree in computer science from Shanghai Jiao Tong University, Shanghai, China and the master's degree from the Technical University of Berlin, Berlin, Germany, and Shanghai Jiao Tong University. He is currently a Senior Research Scientist with the Laboratory of Artificial Intelligence (Seek-Data Ltd.), Emoney Inc., Shanghai, China. His research interests include pattern recognition and data mining in finance, portfolio management, and quantitative investment.

Yifeng Luo received the bachelor's degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2004, the master's degree from Tsinghua University, Beijing, China, in 2007, and the Ph.D. degree from the School of Computer Science, Fudan University, Shanghai, China, in 2015. He is currently an Associate Professor with the School of Data Science and Engineering, East China Normal University, Shanghai, China. His research interests include knowledge engineering, natural language processing, and deep learning.

Dingjiang Huang received the Ph.D. degree in computation mathematics from the Dalian University of Technology, Dalian, China, in 2007. He is currently a Professor with the School of Data Science and Engineering, East China Normal University, Shanghai, China. From 2008 to 2011, he was a Postdoctoral Research Fellow with the School of Computer Science, Fudan University, Shanghai, China, and with the School of Computer Engineering, Nanyang Technological University, Singapore.



Weining Qian (Member, IEEE) received the bachelor's degree, master's degree, and Ph.D. degrees in computer science and technology from Fudan University, Shanghai, China, in 1998, 2001, and 2004, respectively. He is currently a Professor with East China Normal University, Shanghai, China, and the Dean of the School of Data Science and Engineering. His research interests include data management systems for internet-level applications, scalable transaction processing, benchmarking of big data management systems, and massive data analysis and processing and their applications.

Aoying Zhou received the Ph.D. degree from the Department of Computer Science and Technology, Fudan University, Shanghai, China, in 1993. He is currently the Vice President of East China Normal University, Shanghai, China, the Dean of the founding school of the School of Data Science and Engineering, a Professor, and Doctoral Tutor. His research interests include web data management, data-intensive computing, memory cluster computing, distributed transaction processing, Big Data benchmarking, and performance optimization.