Learning With Annotation of Various Degrees

Joey Tianyi Zhou, Meng Fang, Hao Zhang, Chen Gong, Xi Peng,
Zhiguo Cao, and Rick Siow Mong Goh

***Abstract*—In this paper, we study a new problem in the scenario of sequences labeling. To be exact, we consider that the training data are with annotation of various degrees, namely, fully labeled, unlabeled, and partially labeled sequences. The learning with fully un/labeled sequence refers to the standard setting in traditional un/supervised learning, and the *proposed partially labeling* specifies the subject that the element does not belong to. The partially labeled data are cheaper to obtain compared with the fully labeled data though it is less informative, especially when the tasks require a lot of domain knowledge. To solve such a practical challenge, we propose a novel deep conditional random field (CRF) model which utilizes an end-to-end learning manner to smoothly handle fully/un/partially labeled sequences within a unified framework. To the best of our knowledge, this could be one of the first works to utilize the partially labeled instance for sequence labeling, and the proposed algorithm unifies the deep learning and CRF in an end-to-end framework. Extensive experiments show that our method achieves state-of-the-art performance in two sequence labeling tasks on some popular data sets.**

***Index Terms*—Deep conditional random field (CRF), incomplete annotation, partially labeled data, sequence labeling.**

I. Introduction

EXISTING machine learning methods could be roughly grouped into three categories, i.e., supervised learning [1]–[8], semisupervised learning [9]–[13], and unsupervised learning [14]–[16], [16]–[22]. The major difference among them lies on the annotation degrees of training data. To be specific, supervised learning requires that all training data are annotated with labels. In contrast, unsupervised learning relaxes such a limitation and handles training data without the help of annotation. Different from un/full supervised learning, semisupervised learning assumes that only a part of data is fully labeled, which has played a key role in the machine learning community, thanks to less annotation efforts.

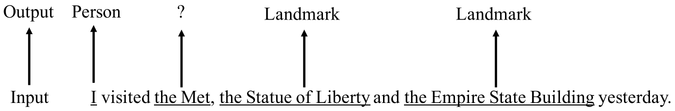

Output
Person
?
Landmark
Landmark
Input
I visited the Met, the Statue of Liberty and the Empire State Building yesterday.

Fig. 1. Example of partial labels: for the people who are not familiar with New York may be unable to annotate "the Met" as the landmark, however, they are able to annotate "the Statue of Liberty" and "the Empire State Building" as landmarks. In consequence, such a common seeing phenomenon will lead to the proposed partial annotations.

Different from the aforementioned learning schemes, this paper relaxes the annotation assumptions adopted in existing works and allows some data to receive partial annotations. Our motivation comes from a lot of practical applications. To be specific, sequence labeling aims at building a system to predict the structured output for a given input, which has been applied to numerous real-world applications including part-of-speech tagging (POS) [23], named entity recognition (NER) [24], speech recognition [25], and so on. Most existing sequence labeling methods such as BiLSTM-CRF [23] follow fully supervised learning setting, which usually use fully labeled sequences to model the transition between different positions in the sequence. Apart from labeled sequences, however, we observe there exists another case in the sequence labeling, termed *partially labeled sequences*. As shown in Fig. 1, "I visited the Met, the Statue of Liberty, and the Empire State Building yesterday." For those people who are not familiar with New York may be unable to annotate "the Met" as the landmark while they are able to annotate "the Statue of Liberty" and "the Empire State Building" as landmarks. In this case, the annotated entity ("the Met") might be missing in a sentence. Clearly, the above-mentioned example can be considered as neither a labeled sequence nor an unlabeled sequence. To our surprise, this problem has been largely ignored in existing literature.

Based on the above-mentioned observation, we propose a novel learning setting, termed as learning with annotation of various degrees (LAVD), which is remarkably different from existing settings such as supervised, unsupervised, or semi-supervised learning. Fig. 2 illustrates the difference between

Manuscript received April 18, 2018; revised September 2, 2018; accepted November 28, 2018. This work was supported in part by Singapore Government's Research, Innovation and Enterprise 2020 Plan (Advanced Manufacturing and Engineering Domain) through programmatic under Grant A1687b0033, in part by the National Natural Science Foundation of China under Grant 61806135, Grant 61432012, Grant U1435213, Grant 61602246, and Grant 61876211, in part by the Fundamental Research Funds for the Central Universities under Grant YJ201748 and Grant 30918011319, in part by the NSF of Jiangsu Province under Grant BK20171430, in part by the Summit of the Six Top Talents Program under Grant DZXX-027, in part by the Lift Program for Young Talents of Jiangsu Province, and in part by the CAST Lift Program for Young Talents. *(J. T. Zhou and M. Fang contributed equally to this work.) (Corresponding author: Xi Peng.)*

J. T. Zhou and R. S. M. Goh are with the Institute of High Performance Computing, A*STAR, Singapore 138632 (e-mail: zhouty@ihpc.a-star.edu.sg; gohsm@ihpc.a-star.edu.sg).

M. Fang is with the AI Lab, Shenzhen 518057, China (e-mail: mfang@tencent.com).

H. Zhang is with Artificial Intelligence Initiative, A*STAR, Singapore 138632 (e-mail: zhang_hao@scei.a-star.edu.sg).

C. Gong is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chen.gong@njust.edu.cn).

X. Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: pengx.gm@gmail.com).

Z. G. Cao is with the School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: zgcao@hust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2885854

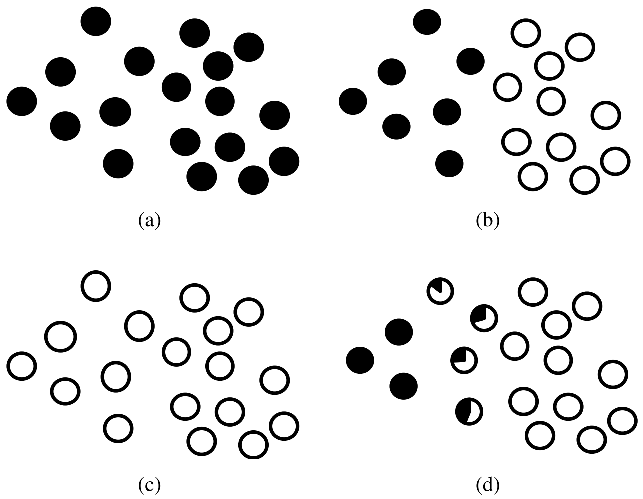


Fig. 2. Visual illustration of different settings. (a) Supervised learning. (b) Semisupervised learning. (c) Unsupervised learning. (d) LAVD.

proposed LAVD and these learning paradigms. To be exact, we aim at solving a sequence labeling task which assumes that the training data simultaneously contain fully labeled, partially labeled, and unlabeled data. To achieve LAVD, we propose a new deep conditional random field, termed dCRF3 which is a marriage of CRFs and neural networks with a novel objective function. The proposed objective function simultaneously considers labeled, partially labeled and unlabeled sequences, which is optimized in an end-to-end manner. The contribution of this paper could be summarized as follows.

1) To the best of our knowledge, this could be one of first works to define a more generalized problem of LAVD in the scenario of sequence labeling. The hidden structures can be learned through exploiting information shared from unlabeled, partially labeled, and fully labeled data in the proposed LAVD.
2) We propose a new model, termed as dCRF3, to achieve the LAVD. Specifically, dCRF3 employs recurrent neural networks (RNNs) to learn relevant high-level features in an end-to-end manner. Furthermore, a new objective function is specifically designed for LAVD. Extensive experimental results show the remarkable improvements over the state-of-the-art (SoA) sequence labeling methods.

II. Related Work

Our work is related to the following topics, i.e., weakly supervised learning and sequence labeling which are briefly introduced in this section.

A. Weakly Supervised Learning

Most SoA techniques such as convolutional neural network [26]–[28] require using a large number of fully labeled data to train model. In practice, however, it is a daunting task to attain strong supervision information due to the high cost of data labeling process. Therefore, more and more attention shifts from fully supervised learning to weakly supervised learning [11], [12]. According to the difference in annotation, weakly supervised learning could be further divided into the following three groups [29] (see Table I), i.e., incomplete supervision, inexact supervision, and inaccurate supervision. To be specific, incomplete supervision requires a subset of training data is fully annotated and the rests are without labels. To address this issue, semisupervised learning is proposed. Inexact supervision is given with coarse-level or ambiguous annotations [30]. Namely, instead of receiving a set of instances which are individually labeled, the learner receives a set of labeled bags, each containing many instances. For example, for the gene-based disease diagnosis [31], the expert requires judging whether some combinations of single-nucleotide polymorphisms (SNPs) lead to a disease, instead of knowing which one SNP is the disease factor. To address this issue, multiinstance learning is proposed. Inaccurate supervision means that some labels are incorrect, which is common seeing in crowdsourcing [32], [33].

The proposed LAVD is remarkably different from these existing weakly supervised learning paradigms in the following two aspects. First, the annotation of data is different. In detail, the partially annotation defined in our LAVD specifies that *a part of* ground truth of *some data points* is missing or undetermined as shown in Fig. 2. In contrast, existing weakly supervised methods assume that all the data points are either fully labeled or unlabeled, which consider the annotation of the whole data set instead of each single data point. Second, LAVD considers all kinds of labeled data in a unified framework, namely, fully labeled, partially labeled, and unlabeled. To solve such a complex and challenging issue, we specifically design a deep learning-based method (dCRF3) in sequence labeling. Note that some existing works on learning with partially labeled sequence are quite highly related to our setting. Specifically, Tsuboi *et al.* [34] proposed training CRFs using incomplete annotations and [35] extended it to alleviate the annotation efforts by utilizing the active learning. Fernandes and Brefeld [36] devised a simple transductive loss-augmented perceptron to learn from inexpensive partially annotated sequences. Different from these works, in our LAVD setting, we consider three types of annotations in a unified framework. In this way, these works can be considered as a special case in LAVD. In addition, different from the mentioned works with hand-craft features, the proposed dCRF3 are a deep learning-based method and enables the end-to-end training, thus leading to the significant improvement in performance.

B. Sequence Labeling

Sequence labeling including POS tagging and NER has been studied for many years in NLP. Most early studies were based on hand-crafted rules which have suffered from degraded performance in practice. In recent, more and more works have devoted to developing learning-based methods to automatically induce sequence labeling, especially supervised methods. Supervised learning algorithms, e.g., hidden Markov model (HMM) [37], support vector machine [38], CRF [39], and neural networks [23], typically employ a system to read a large-scale annotated training data, memorize a

TABLE I
WEAK SUPERVISION SETTING COMPARISON

Setting	Supervision Level	Supervision Accuracy	Unlabeled data	Partially labeled data
Semi-supervised Learning	instance	accurate	yes	no
Multi-instance Learning	bag	accurate	no	no
Crowdsourcing	instance	inaccurate	no	no
LAVD (Our)	instance	accurate	yes	yes

list of entities, and create disambiguation rules based on discriminative features. Among them, CRFs and HMMs are very popular and have achieved SoA performance in handling sequential data [23], [39]. Comparing with generative models (e.g., HMMs), discriminative models (e.g., CRFs) are usually more powerful, which focus on maximizing the conditional probability of the true label rather than modeling the joint distribution. Thanks to impressive results achieved by CRFs, it has been widely used for handling sequential data in the scenario of natural language processing and biological sequence analysis.

More recently, the huge success achieved by deep learning has inspired increasing interest in combining CRFs and neural networks [24], [40]–[42]. For example, Liu *et al.* [43] proposed a deep neural network which learns the unary and pairwise potentials of continuous CRF in a unified deep CNN framework for the task of depth estimation. Lin *et al.* [44] combined the strengths of both CNNs and CRF-based graphical models within a unified framework for the semantic image segmentation problem. In natural language processing, Do and Artieres [45] proposed an approach to combine deep neural networks and Markov networks for solving the sequence labeling problem. Yao *et al.* [46] showed that the performance of an RNN-based word tagger can be significantly improved by incorporating the elements of CRFs in language understanding. Different from our method, most of these approaches treat learning with CRF and feature extraction with neural networks as two separate steps of supervised learning scheme.

Similar to most classification methods [26], [47], sequence labeling [23] usually requires to use a large number of fully labeled sequences for training, where all elements in a sequence have been annotated in advance. As discussed in Section I, such fully labeled sequences are hard to be obtained, especially in the scenario of NLP. Different from computer vision tasks, sequence labeling often involves expert knowledge from the NLP domain, which is much more expensive than image annotation.

Unlabeled data, in contrast, are abundant and contain latent sequential patterns. Unlabeled data for sequence labeling have been extensively studied in unsupervised learning, such as unsupervised POS tagging [48]–[50]. For example, Ammar *et al.* [49] introduced the CRF autoencoder to represent data. Lin *et al.* [50] considered replacing words by word embeddings in HMMs or CRFs models. When the amount of labeled data is fixed, a better performance could always be achieved if unlabeled data are utilized during learning [51], [52].

To reduce the high cost for annotation, semisupervised learning algorithms were developed for sequential labeling such that both unlabeled and labeled data are utilized. For example, Täckström *et al.* [52] and Tsuboi *et al.* [34] introduced a weakly supervised constrained lattice training method with outside resource, such as dictionaries. Zhang *et al.* [53] used the CRF autoencoder and provided an electromagnetic-based training method. Marinho *et al.* [51] introduced moment-based semisupervised learning method for HMMs. Täckström *et al.* [52] and Tsuboi *et al.* [34] introduce a constrained lattice training method with partially labeled data. Verbeek and Triggs [54] and Marinho *et al.* [51] extended traditional CRFs/HMMs in a semisupervised manner to utilize unlabeled data. Moreover, Marcheggiani and Artieres [35] proposed an active learning method to reduce annotation efforts for sequence labeling. Clearly, these semisupervised learning methods are different from the proposed one since most of them assume each sentence are fully labeled or unlabeled. The partially labeled data source in sequence labeling is experimentally shown effectiveness in bridging labeled and unlabeled data. As a consequence, such a limitation may hinder their performance in boosting SoA.

Based on the works described earlier, we demonstrate that it is possible to learn from different types of data sources within a unified framework, ranging from fully labeled data, to the partially labeled, and even to unlabeled data, for sequence labeling. To the end, we provide an end-to-end trainable system for all data sources with no manual feature engineering.

III. dCRF3 FOR SEQUENCE LABELING

In this section, we introduce a new deep conditional random fields for sequence labeling, termed as dCRF3. Our model mainly contains two components: one is for learning feature representation and the other is for labeling sequence. Fig. 3 has shown the architecture of our model.

A. Conditional Random Fields

Sequence classification (or labeling) problems aim at modeling the dependencies among labels, while predicting a structured output sequence for the input sequence. In such tasks, dependencies among neighboring labels are crucial, it is conducive to consider the correlations among labels in neighborhoods and jointly decode a chain of labels so that the resulting label sequence could be meaningful. For example, in NER task with beginning, inside, outside, end and single annotation, it is meaningless and illegal to annotate *I-LOC* after *B-ORG* (i.e., mixing the different annotation types). CRF is widely used to make joint labeling of the tokens in a sequence [39], which is capable of capturing the dependence information and further avoiding generating illegal annotations. Therefore, we model the label sequence jointly using the CRF, instead of predicting each label independently.

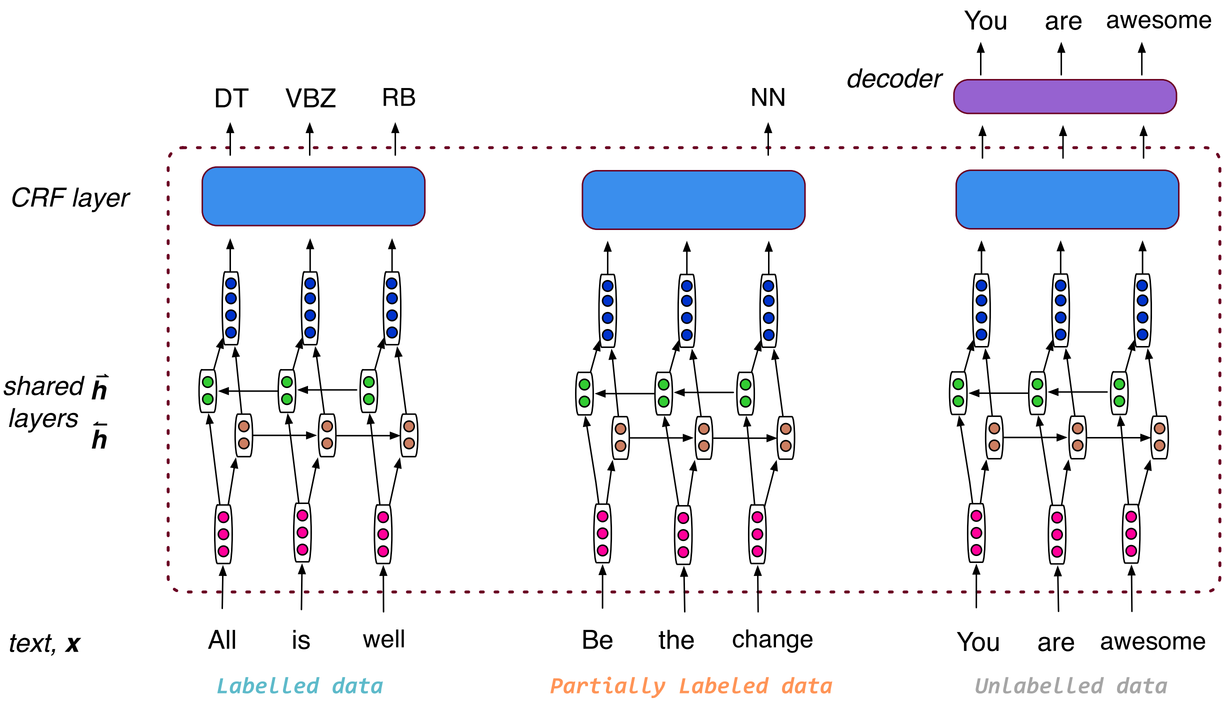


Fig. 3. Structure of dCRF3.

Formally, let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ represent a sequence of input instances and $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ be a list of predicted random variables whose components belong to a set of labels $\mathbf{Y}$. Here, $\mathbf{y}$ is linked by conditional dependencies which are encoded by an undirected graph or chain $G = (V, E)$ with cliques $c \in C$. For a given input sequence $\mathbf{x}$, inference is performed by finding the output which maximizes the conditional probability $p(\mathbf{y}|\mathbf{x})$. Based on the above-mentioned notations and the Hammersley–Clifford theorem, the conditional probability in CRF is defined by

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Psi(\mathbf{x}, \mathbf{y}_c) \tag{1}$$

where $Z(\mathbf{x})$ is a global normalization term with the definition of $Z(\mathbf{x}) = \sum_y \prod_{c \in C} \Psi_c(\mathbf{x}, \mathbf{y}_c)$, and $\Psi(\mathbf{x}, \mathbf{y}_c)$ denotes a potential function parameterized by features.

In general, the features are constructed by human experts or preprocessing steps, e.g., the popular log-linear model [39]. Different from these traditional approaches, we propose to achieve the features by a neural network. In other words, our method learns features from data in a data-driven way, thus enjoying better representative capacity.

B. Feature Representation Based on Neural Networks

In this section, we elaborate the feature learning module adopted in our dCRF3. More specifically, dCRF3 passes the input data points through a neural network consisting of several hidden layers so that the latent representation is learned as the features. Different from popular neural networks, our network consists of two different connections, namely, feed-forward and lateral connection (see Fig. 3). The lateral connection is used to capture the context information, which is implemented by long short-term memory (LSTM) units. The feed-forward connection is designed to learn feature representation which simultaneously incorporates the temporal information given by the lateral connection. In short, with the corporation of these two connections, the neural network works as an automatic feature generator and could learn high-level representation. One unique characteristic of sequence labeling is that the historical and future input for a given time step could be accessed. To exploit such a characteristic, we use a bidirectional LSTM architecture [55] to extract features. LSTM is a variant of RNN, capable of learning long-term dependencies and coping with the gradient vanishing/exploding problems. Basically, LSTM unit is similar to RNN unit, except that the hidden layer updates are replaced by purpose-built memory cells, where those cells are three multiplicative gates which control the proportions of information to forget and to pass to the next time step. Fig. 4 has shown the schematic of a LSTM unit [56].

Formally, the formulas to update an LSTM unit at time t are

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i h_{t-1} + \mathbf{U}_i x_t + \mathbf{b}_i)\\
\mathbf{f}_t &= \sigma(\mathbf{W}_f h_{t-1} + \mathbf{U}_f x_t + \mathbf{b}_f)\\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c h_{t-1} + \mathbf{U}_c x_t + \mathbf{b}_c)\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t\\
\mathbf{o}_t &= \sigma(\mathbf{W}_o h_{t-1} + \mathbf{U}_o x_t + \mathbf{b}_o)\\
h_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}$$

where σ is the sigmoid activation function, $\odot$ represents elementwise product, $\mathbf{i}$, $\mathbf{f}$, and $\mathbf{o}$ are the input gate, forget gate, and output gate, respectively. x_t represents the input instance at the time stamp t, and h_t is the corresponding hidden state

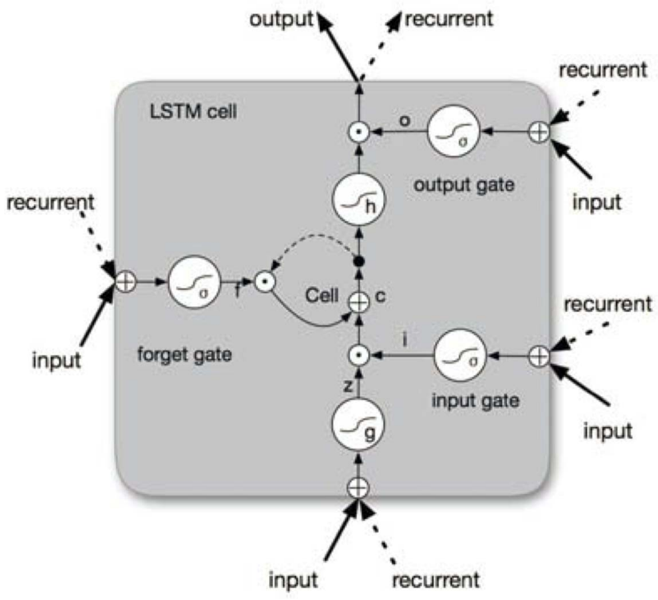


Fig. 4. Schematic of LSTM unit.

(also called output) at time of t. $\mathbf{W}_*$ denotes the weights for hidden state h_t, $\mathbf{U}_*$ are the weights of different gates for input x_t, and $\mathbf{b}_*$ denotes the bias.

The hidden state h_t of LSTM is capable of taking information from left (past) contexts but cannot take information from right (future) contexts. However, it is beneficial to have access to both past and future contexts for sequence labeling tasks. Therefore, we introduce bidirectional LSTM (Bi-LSTM) which presents the input sequence forward and backward to two separate hidden states to learn the past and future information, respectively. Specifically, let $h_t = \texttt{lstm}(h_{t-1}, x_t)$ denote the hidden state update process of LSTM, for Bi-LSTM, we have

$$\overrightarrow{h}_t = \texttt{lstm}(\overrightarrow{h}_{t-1}, x_t)$$
$$\overleftarrow{h}_t = \texttt{lstm}(\overleftarrow{h}_{t+1}, x_t)$$

where $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ denote forward hidden state and backward hidden state, respectively. To enjoy nonlinearity, we add a nonlinear layer after the LSTM layer, namely

$$h_t = \texttt{tanh}(W_r \overrightarrow{h}_t + W_l \overleftarrow{h}_t + b_t). \tag{2}$$

This representation includes both local and global information, which could capture contextually sensitive signals across the sequence.

C. CRF Layer for Sequence Labeling

With the features generated by the aforementioned neural network, we design a general model to capture the relationships of labels in a sequence. In this paper, for sequence classification, we employ a linear chain model based on the first-order Markov chain structure because it allows investigating the potential power on standard sequence labeling tasks. In the chain model, there are two kinds of cliques, namely, local cliques and transition cliques. More specifically, local cliques correspond to the elements in a sequence, whose representation is denoted by h_t as defined in (2). Alternatively, transition cliques, on the other hand, reflect the evolution of states between two neighboring elements at the timestamp $t-1$ and t, where the transition distribution is defined by θ.

Formally, a linear-chain CRF can be written as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{\sum_{i=1}^{|\mathbf{x}|} \theta_{y_{i-1},y_i} + W_{y_i} h_i\right\} \tag{3}$$

where $Z(\mathbf{x})$ is an instance-specific normalization function and θ indicates a transition matrix that contains transition probabilities, i.e., $\theta_{i,j}$ is the probability of transition (y_i, y_j).

Unlike most CRF models and their variants, we do not assume every element to be labeled. In other words, in our setting, there may be a fully labeled sequence $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$, a partially labeled sequences, $\{(x_4, y_4), x_5, (x_6, y_6)\}$, and an unlabeled sequence $\{x_7, x_8, x_9\}$. Clearly, traditional CRFs cannot solve such a complex problem. In the following, we will introduce how to solve this problem one by one.

D. Fully Labeled Data

The scenario of fully labeled data is a standard case in [39]. For a given independent identically distributed training data set $\mathcal{D}_l$, $\mathbf{x}$ denotes the input sequence and $\mathbf{y}$ denotes the corresponding label. The conditional probability of $\mathbf{y}$ with respect to $\mathbf{x}$ is calculated using (3), and feature learning is achieved by maximizing the conditional log-likelihood as follows:

$$\sum_{\mathbf{x}\in\mathcal{D}_l} \log p(\mathbf{y}|\mathbf{x}). \tag{4}$$

The model parameters could be optimized by solving the above-mentioned problem. In this paper, we further reformulate the optimization as a layer and add it at the top of the feature learning neural network. More specifically, we define the objective function with the conditional likelihood as follows:

$$\ell_l = -\sum_{\mathbf{x}\in\mathcal{D}_l} \log p(\mathbf{y}|\mathbf{x}). \tag{5}$$

The loss is then collected through the forward pass of the whole neural network, which is further used to compute the gradient with respect to each parameter. With the gradient, our model could be optimized with backpropagation. Moreover, we also enforce ℓ_2 regularization to the parameters of the neural network to avoid overfitting.

E. Partially Labeled Data

The scenario with partially labeled data is different from that with fully labeled data. Therefore, we cannot use the standard conditional probability of output $\mathbf{y}$ straightforwardly. Instead, we propose a new way to address labeled and unlabeled elements in a sequence. In detail, for labeled elements, we employ the same procedure described previously to process them.

For unlabeled elements, we consider all possible labels by defining a new conditional probability of $\mathbf{y}$

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \Bigg\{ & \sum_{i \in \{\mathbb{1}_{y_{i-1}}=1, \mathbb{1}_{y_i}=1\}} \theta_{y_{i-1},y_i} + W_{y_i} h_i \\ & + \sum_{i \in \{\mathbb{1}_{y_{i-1}}=0, \mathbb{1}_{y_i}=1\}} \sum_{y_{i-1}} \theta_{y_{i-1},y_i} + W_{y_i} h_i \\ & + \sum_{i \in \{\mathbb{1}_{y_{i-1}}=1, \mathbb{1}_{y_i}=0\}} \sum_{y_i} \theta_{y_{i-1},y_i} + W_{y_i} h_i \Bigg\} \end{aligned} \tag{6}$$

where $\mathbb{1}_{y_i} = 1$ indicates that y_i is known at the ith clique, $\mathbb{1}_{y_i} = 0$ indicates that y_i is unknown at the ith clique, and $Z(\mathbf{x})$ indicates the normalization factor. Comparing with the standard CRF model [34], $Z(\mathbf{x})$ keeps unchanged in our method.

Similar to (4), the estimation of parameters for partially labeled data is based on the conditional log-likelihood. Thus, we design our neural network by using the conditional log-likelihood as another objective function with the following definition:

$$\ell_p = - \sum_{\mathbf{x} \in \mathcal{D}_p} \log p(\mathbf{y}|\mathbf{x}). \tag{7}$$

From the above-mentioned equation, our method models the labeled and partially labeled data in the similar way. Thus, we use the same parameters for these two scenarios.

F. Unlabeled Data

Unlabeled case is an extremely special case of our model, i.e., there is no label available for any sequence. Thus, we cannot construct conditional likelihood like what we did for labeled instances. To address this issue, we assume there are hidden sequential patterns behind the observed data and the unlabeled data actually carry useful information for recognizing these patterns. To explore and exploit the useful information hidden into unlabeled data, we introduce an encoder–decoder network structure [49], [57] for handling unlabeled data.

To be specific, we build a model $p(\hat{\mathbf{x}}|\mathbf{x})$ by encoding the input into a latent state sequence which is further decoded to reconstruct the input sequence.[1] The idea behind our model is utilizing latent states to represent the input sequence as accurate as possible. In our setting, the deep CRF is regarded as an encoder, which exploits the global features of the input observation $\mathbf{x}$. For the decoder, we introduce a simple decoding layer to reconstruct the sequence according to independent categorical distributions, that is, conditioned on the hidden label

$$p(\hat{\mathbf{x}}|\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) p(\hat{\mathbf{x}}|\mathbf{y}) \tag{8}$$

where $p(\mathbf{y}|\mathbf{x})$ is the encoder, and each $\mathbf{x}_i$ is generated from $p_\alpha(x_i|y_i)$. This further leads to

$$\begin{aligned} p(\hat{\mathbf{x}}|\mathbf{x}) &= \sum_{\mathbf{y}} \frac{p(\mathbf{y}|\mathbf{x})}{Z(\mathbf{x})} \prod_{i=1}^{|\mathbf{x}|} p_\alpha(\hat{x}_i|y_i) \\ &= \sum_{\mathbf{y}} \frac{\exp\left(\sum_i^{|\mathbf{x}|} \log \alpha_{\hat{x}_i|y_i} + \theta_{y_{i-1},y_i} + W_{y_i} h_i\right)}{Z(\mathbf{x})} \end{aligned} \tag{9}$$

where $\mathbf{y}$ ranges over all label sequences and α is a matrix of conditional multinominals, i.e., a word $\hat{x}_i$ conditioned on a label y_i. α is defined by a softmax function, such that $\alpha_{\cdot|y_i}$ are constrained to lie on the simplex.

We also add this part into the proposed model and treat it as the top layer. We use the conditional likelihood as the objective function to train the model. Note that unlabeled case is different from the fully/partially labeled case thanks to the existence of the decoder. Equation (9) has defined the conditional likelihood of the reconstructed observations $\hat{\mathbf{x}}$ given the observation $\mathbf{x}$, which is used to define the loss function as follows:

$$\ell_u = - \sum_{\mathbf{x} \in \mathcal{D}_u} \log p(\hat{\mathbf{x}}|\mathbf{x}). \tag{10}$$

G. Joint Training

Thus far, we have presented our neural network-based CRFs model for different cases, ranging from fully labeled, to partially labeled, to unlabeled data. For each case, we build different losses that are integrated together for jointly training our model. Note that despite the differences in the degree of annotation, we let them share the same neural network architecture. The motivation is that unlabeled data also contain sequential patterns similar to those found in labeled data, such as similar grammatical properties. Combining (5), (7), and (10), we define the joint objective function as follows:

$$\sum_{\mathcal{D}_l} \ell_l(\mathcal{W}) + \sum_{\mathcal{D}_p} \ell_p(\mathcal{W}) + \sum_{\mathcal{D}_u} \ell_u(\mathcal{W}, \alpha) + ||\mathcal{W}||_2 + ||\alpha||_2 \tag{11}$$

where $\mathcal{W}$ indicates the parameters of our neural network-based CRFs. Note that ℓ_2 regularization is adopted to avoid overfitting.

Our model enjoys the advantages of both neural networks and CRFs, which is trainable end-to-end model and could be optimized by the stochastic gradient descent. During training, sequences from labeled, partially labeled, and unlabeled data sources are collected to form respective minibatches and the error is computed for the purpose of optimizing our model.

Once the parameters are learned, predictions are made by maximizing the posteriori estimation of $\mathbf{y}$ with respect to $\mathbf{x}$, that is,

$$\mathbf{y}^* = arg \max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}). \tag{12}$$

Here, the above-mentioned problem is solved by adopting the Viterbi algorithm.

[1] We use $\hat{\mathbf{x}}$ to denote the output of the autoencoder. Note that $\hat{\mathbf{x}}$ is the approximation to the input $\mathbf{x}$.

TABLE II

USED DATA

data splits	sentences	tokens
Penn Treebank		
Training set	38,219	912,344
Develop set	5,527	131,768
Test set	5,462	129,654
CoNLL2003		
Training set	14,987	204,567
Develop set	3,466	51,578
Test set	3,684	46,666
New York Times		
Training set	3,514,623	80,836,329

IV. EXPERIMENTS

We verify the effectiveness of our method for two widely used sequential labeling tasks, namely, POS tagging and NER on two different data sets.

A. Data Sets

We use two labeled data sets for these two different tasks, i.e., Penn Treebank [58] for POS tagging and CoNLL2003 [59] for NER. For all the evaluated data sets, we use the default data partitions and investigate the performance of the testing partition. By following the popular testing protocol [23], we adopt accuracy and F-score as the performance metric for POS tagging and NER, respectively. Furthermore, we also use the New York Times corpus from July 1994 to June 1995 as unlabeled data. Table II summaries the statistical characteristics of these data sets.

B. Word Embedding

It has been shown in [42] that word embedding is beneficial to NLP tasks and plays a vital role to improve sequence labeling performance. In our experiments, we use Stanford's publicly available GloVe 200-dimensional word embeddings[2] trained on six billion words from Wikipedia and web text [60].

C. Experimental Settings

We compare dCRF3 with the following five SoA baselines.

1) *CRF:* A standard linear-chain CRF model.
2) *CRFCC:* A semi-CRF-based model to address unlabeled data using projected labels [52]. For comparisons, we follow its unsupervised setting and use all possible labels instead.
3) *RNN:* A basic RNN model [61].
4) *BiLSTM:* A bidirectional LSTM network [55].
5) *BiLSTM-CRF:* The SoA in sequence labeling tasks including POS tagging and NER. In short, the method adds a CRF layer on a bidirectional LSTM [23].

Among the evaluated methods, CRF and BiLSTM-CRF only use the labeled data for training. RNN and BiLSTM ignore the loss of unlabeled elements in partially labeled sequences and cannot utilize unlabeled data either. In contrast, our proposed method is capable of exploiting labeled, partially labeled and unlabeled data if provided. For CRF, we use standard feature functions.[3] For our dCRF3 model, the word embedding size is 200 and LSTM hidden dimension is 128. Similarly, we set the same for BiLSTM and BiLSTM-CRF. For the Penn Treebank data set, the size of the label set is 45. For the CoNLL2003 data set, we use the inside and outside format and the size of the label set is 4.

TABLE III

PERFORMANCE ON POS TAGGING (ACCURACY) AND NER (F-SCORE) FOR DIFFERENT MODELS WITH DIFFERENT TYPES OF DATA. ALL MODELS HAVE ONE FOLD (10% OF THE DATA SET) OF FULLY LABELED TRAINING DATA, AND THE COLUMNS SHOW THE VARYING AMOUNTS OF PARTIALLY LABELED TRAINING DATA (1 FOLD = 10%)

Partially labelled		10%	20%	30%	40%	50%
POS	CRF	94.5	94.5	94.5	94.5	94.5
	RNN	92.2	93.0	93.2	93.6	94.1
	BiLSTM	95.2	95.2	95.3	95.5	96.4
	BiLSTM-CRF	95.1	95.3	95.1	95.1	95.4
	CRFCC	94.8	94.8	95.6	95.2	96.0
	dCRF3	95.6	95.9	96.2	96.4	96.8
NER	CRF	71.1	71.1	71.1	71.1	71.1
	RNN	70.3	70.6	70.7	70.9	71.1
	BiLSTM	73.4	75.4	79.1	80.6	83.8
	BiLSTM-CRF	73.9	74.0	74.1	74.2	74.2
	CRFCC	72.3	72.6	73.5	74.6	75.8
	dCRF3	74.6	78.2	82.2	85.5	86.5

D. Experiments

In our experiments, we have three different settings: the partially labeled data setting, the unlabeled data setting, and the joint partially and unlabeled data setting. We only use Penn Treebank and CoNLL2003 data sets. In the beginning, we split the training set into 10 equally sized folds for each data set. As the used data set is quite large, even 10% of the data sets is very challenging. For example, 10% of Penn Treebank data set includes 3821 sentences, which is sufficient to train a good model as shown in our experiments. We simulate labeled, unlabeled, and partially labeled data based on these folds and report the averaged results over 10 repeated tests. For partially labeled data, we randomly removed 50% label information in each sentence to create partially labeled data.

1) Partially Labeled Data: The upper half of Table III shows the performance of different models with varying number of folds of partially labeled data. It can be observed that our model outperforms other models, rather significantly in the case of NER. Note that dCRF3, CRFCC, BiLSTM, and RNN all learn information from the partially labeled data whereas BiLSTM-CRF and CRF cannot. Both dCRF3 and BiLSTM generate superior results to RNN, likely due to the fact that LSTMs are better at memorizing sequential information.

[2] https://nlp.stanford.edu/projects/glove/

[3] http://www.nltk.org/_modules/nltk/tag/crf.html

TABLE IV

PERFORMANCE ON POS TAGGING (ACCURACY) AND NER (F-SCORE) FOR DIFFERENT MODELS WITH DIFFERENT TYPES OF DATA. ALL MODELS HAVE ONE FOLD (10% OF THE DATA SET) OF FULLY LABELED TRAINING DATA, AND THE COLUMNS SHOW THE VARYING AMOUNTS OF UNLABELED TRAINING DATA (1 FOLD = 10%)

Unlabelled		10%	20%	30%	40%	50%
POS	CRF	94.5	94.5	94.5	94.5	94.5
	RNN	90.3	90.3	90.3	90.3	90.3
	BiLSTM	95.0	95.0	95.0	95.0	95.0
	BiLSTM-CRF	95.4	95.4	95.4	95.4	95.4
	CRFCC	94.5	94.9	95.1	95.4	95.7
	dCRF3	95.4	95.8	96.0	96.3	96.5
NER	CRF	70.9	70.9	70.9	70.9	70.9
	RNN	70.0	70.0	70.0	70.0	70.0
	BiLSTM	71.5	71.5	71.5	71.5	71.5
	BiLSTM-CRF	74.0	74.0	74.0	74.0	74.0
	CRFCC	71.1	71.8	72.3	73.0	73.5
	dCRF3	74.3	75.5	76.5	77.0	79.6

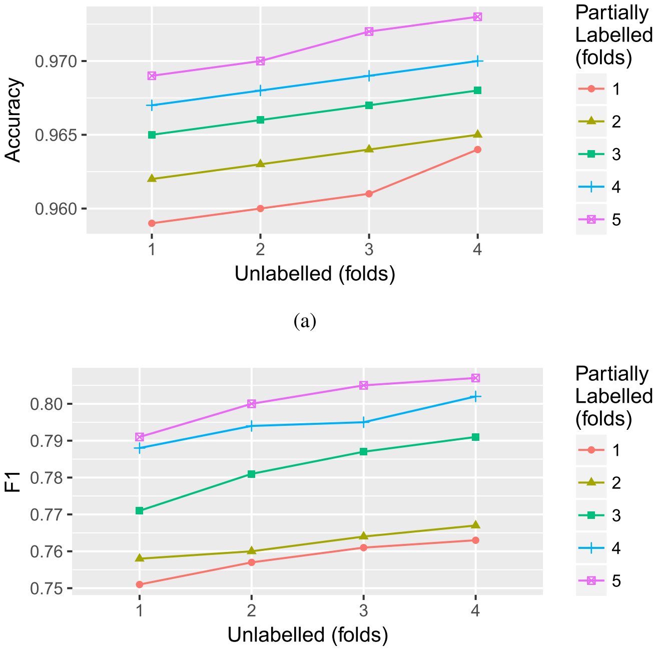
CRFCC does not work well because it assigns all possible labels to unlabeled words and some labels are fake. As the volume of partially labeled data increases, the performance of dCRF3 and RNN and BiLSTM, as they receive more supervision, also improves.

2) Unlabeled Data: The lower half of Table IV presents the performance of different models with varying number of folds of unlabeled data. Again, our model is superior to the other models. This can be attributed to dCRF3's ability to learn from unlabeled data while others cannot. It is also shown that the performance of dCRF3 is boosted as more unlabeled data comes in, demonstrating that the unlabeled data are useful for supervised learning. CRFCC uses the unlabeled data and projected labels to make a supervised model possible. However, this supervision is not accurate. Unlike dCRF3 based on neural networks, CRFCC uses a standard linear combination of features and weights, which is insufficient for modeling unlabeled sequences.

3) Joint Partially Labeled and Unlabeled Data: In Fig. 5, we show the performance on the two tasks using both the partially labeled and unlabeled data. As the amount of the data grows, our model becomes more competent, suggesting that useful information can be drawn and learned from such data sources. Looking further into the correlation between model performance and the size of data from each source, we discover that, while including more data from either source generally helps, the contribution of adding partially labeled data is more evident than that of unlabeled data.

E. Real-World Scenario

We further investigate how well our method works on real-world data sets with different real-world unlabeled data. We use original Penn Treebank and CoNLL2003 data sets for POS tagging and NER tasks, respectively. We use full-sized training set as labeled data. The unlabeled data is from the New York Times corpus. We consider different sizes of the unlabeled data. We show our results in Fig. 6. It shows that with the help of unlabeled data, our method becomes slightly better than the SoA results (based on the BiLSTM-CRF model [23]). Because unlabeled data help to learn better representation. However, after a portion of the unlabeled data, the performance does not increase. This illustrates the limitation of the unlabeled data.



(a)

(b)

Fig. 5. Performance of the proposed model with different sizes of joint partially labeled and unlabeled data (1 fold = 10%). (a) POS tagging. (b) NER.

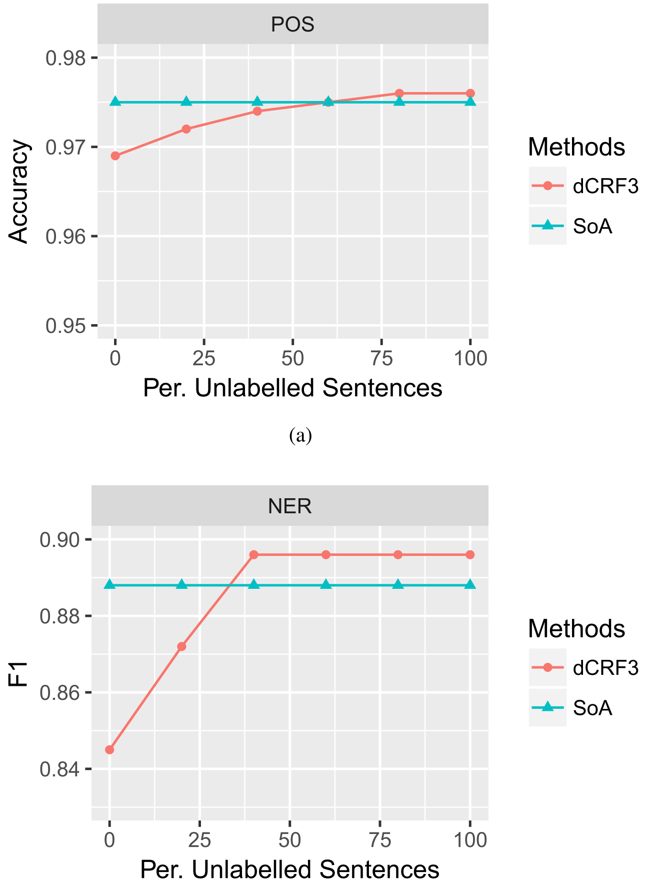
F. Parameter Analysis

In real cases, there are always different numbers of labeled, partial labeled, and unlabeled samples. Thus, it should be critical to set parameters before three loss functions imposed on these three types of samples in (11), respectively. We further conduct the joint training with different weights for different parts in loss function to consider the data imbalance, which is written as follows:

$$\ell = \sum_{\mathcal{D}_l} \ell_l(\mathcal{W}) + \lambda_p \sum_{\mathcal{D}_p} \ell_p(\mathcal{W}) + \lambda_u \sum_{\mathcal{D}_u} \ell_u(\mathcal{W}, \alpha) + ||\mathcal{W}||_2 + ||\alpha||_2. \quad (13)$$

In the real-word applications, the number of labeled data n_l is usually smaller than the number of partially labeled data n_p and the number of unlabeled data n_{n_u}, since the fully labeled data are more expensive, i.e., $(n_l \leq \min\{n_p, n_u\})$. Therefore, the equal weight for different types of data may be suboptimal solution for LAVD in this situation. The weights for partially labeled and unlabeled data should decay according to the ratio.

Based on this observation, we further experimentally investigate five heuristic strategies for determining the



(a)

(b)

Fig. 6. Performance of the proposed model with different sizes of unlabeled data comparing with the SoA results. SoA results of the two tasks. (a) POS tagging. (b) NER.

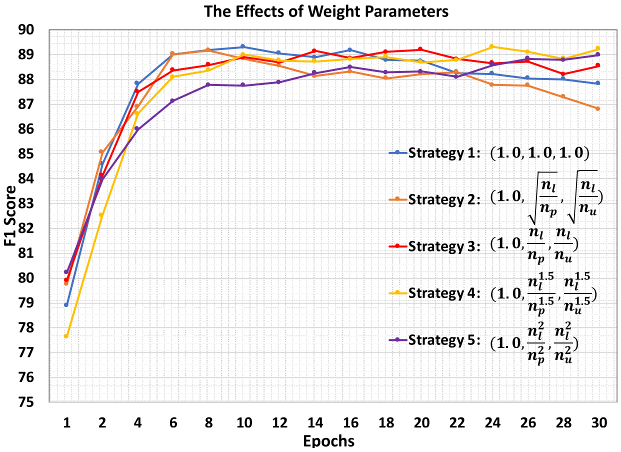


Fig. 7. Effects of weight parameters.

weights $(\lambda_l, \lambda_p, \lambda_u)$. The first one is the equal weight $(1, 1, 1)$. The second one is $(1, (n_l/n_p), (n_l/(n_u))$. The third one is $(1, (n_l/n_p)^{1/2}, (n_l/n_u)^{1/2})$. The fourth is $(1, (n_l^{1.5}/n_p^{1.5}), (n_l^{1.5}/n_u^{1.5}))$. The fifth is $(1, (n_l^2/n_p^2), (n_l^2/n_u^2))$. In the experiment, we set 10%, 40%, and 50% data as the fully labeled data, partially labeled, and data and unlabeled data, respectively. The results for these five strategies are shown in the Fig. 7 for NER on CoNLL2003 data set. From the results, we empirically find that the fourth strategy $(1, (n_l^{1.5}/n_p^{1.5}), (n_l^{1.5}/n_u^{1.5}))$ performs the best.

V. Conclusion

In this paper, we proposed a new deep CRF to learn hidden structured patterns from the data collected in various scenarios. In brief, the data are with the annotation of various degrees, ranging from labeled data, to partially labeled, to even unlabeled data. Experimental results suggest that partially labeled and unlabeled data can improve the performance of sequence labeling.

Acknowledgment

The authors would like to thank the anonymous editors and reviewers for their valuable comments and constructive suggestions to improve the quality of this paper.

References

[1] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised machine learning: A review of classification techniques,” in *Proc. Conf. Emerg. Artif. Intell. Appl. Comput. Eng.*, The Netherlands, Jun. 2007, pp. 3–24.

[2] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[3] C. Deng, X. Liu, C. Li, and D. Tao, “Active multi-kernel domain adaptation for hyperspectral image classification,” *Pattern Recognit.*, vol. 77, pp. 306–315, May 2018.

[4] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa, “Learning common and feature-specific patterns: A novel multiple-sparse-representation-based tracker,” *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2022–2037, Apr. 2018.

[5] X. Lan, A. J. Ma, P. C. Yuen, and R. Chellappa, “Joint sparse representation and robust feature-level fusion for multi-cue visual tracking,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5826–5841, Dec. 2015.

[6] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, “Unsupervised deep hashing with similarity-adaptive and discrete optimization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3034–3044, Dec. 2018.

[7] T. Zhang, C. L. P. Chen, L. Chen, X. Xu, and B. Hu, “Design of highly nonlinear substitution boxes based on I-Ching operators,” *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3349–3358, Dec. 2018.

[8] Z. Huang, H. Zhu, J. T. Zhou, and X. Peng, “Multiple marginal Fisher analysis,” *IEEE Trans. Ind. Electron.*, to be published, doi: 10.1109/TIE.2018.2870413.

[9] X. Zhu, “Semi-supervised learning literature survey,” Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, 2006, p. 4, vol. 2, no. 3.

[10] S. Li and Y. Fu, “Low-rank coding with b-matching constraint for semi-supervised classification,” in *Proc. IJCAI*, Beijing, China, Aug. 2013, pp. 1472–1478.

[11] B. Zhuang, L. Liu, Y. Li, C. Shen, and I. Reid, “Attend in groups: A weakly-supervised deep learning framework for learning from Web data,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2915–2924.

[12] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, “Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.

[13] X. Lu, W. Zhang, and X. Li, “A coarse-to-fine semi-supervised change detection for multispectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3587–3599, Jun. 2018.

[14] M. Khanum, T. Mahboob, W. Imtiaz, H. A. Ghafoor, and R. Sehar, “A survey on unsupervised machine learning algorithms for automation, classification and maintenance,” *Int. J. Comput. Appl.*, vol. 119, no. 13, pp. 1–6, 2015.

[15] Y. Yang, F. Liang, N. Jojic, S. Yan, J. Feng, and T. S. Huang. (Aug. 2017). “Discriminative similarity for clustering and semi-supervised learning.” [Online]. Available: https://arxiv.org/abs/1709.01231

[16] J. T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, and R. S. M. Goh, "Transfer hashing: From shallow to deep," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6191–6201, Dec. 2018, doi: 10.1109/TNNLS.2018.2827036.

[17] H. Yong, D. Meng, W. Zuo, and L. Zhang, "Robust online matrix factorization for dynamic background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1726–1740, Jul. 2018.

[18] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1888–1902, Aug. 2018.

[19] Y. Yang, J. Feng, N. Jojic, J. Yang, and T. S. Huang, "Subspace learning by $\ell 0$-induced sparsity," *Int. J. Comput. Vis.*, vol. 126, no. 10, pp. 1138–1156, Oct. 2018.

[20] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

[21] X. Peng, C. Lu, Y. Zhang, and H. Tang, "Connections between nuclear-norm and frobenius-norm-based representations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 218–224, Jan. 2018.

[22] X. Li, J. Lv, and Z. Yi, "An efficient representation-based method for boundary point and outlier detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 51–62, Jan. 2018.

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

[24] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. (2016). "Neural architectures for named entity recognition." [Online]. Available: https://arxiv.org/abs/1603.01360

[25] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 6645–6649.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[27] H. Zhu *et al.*, "YoTube: Searching action proposal via recurrent and static regression networks," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2609–2622, Jun. 2018.

[28] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3893–3903, Aug. 2018.

[29] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2018.

[30] S. Xiao, D. Xu, and J. Wu, "Automatic face naming by learning discriminative affinity matrices from weakly labeled images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2440–2452, Oct. 2015.

[31] Q. Wu, Y. Ye, Y. Liu, and M. K. Ng, "SNP selection and classification of genome-wide SNP data using stratified sampling random forests," *IEEE Trans. Nanobiosci.*, vol. 11, no. 3, pp. 216–227, Sep. 2012.

[32] B. Han, Y. Pan, and I. W. Tsang, "Robust Plackett–Luce model for k-ary crowdsourced preferences," *Mach. Learn.*, vol. 107, no. 4, pp. 675–702, 2018.

[33] B. Han, I. W. Tsang, L. Chen, C. P. Yu, and S.-F. Fung, "Progressive stochastic learning for noisy labels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5136–5148, Oct. 2018.

[34] Y. Tsuboi, H. Kashima, H. Oda, S. Mori, and Y. Matsumoto, "Training conditional random fields using incomplete annotations," in *Proc. ACL*, 2008, pp. 897–904.

[35] D. Marcheggiani and T. Artieres, "An experimental comparison of active learning strategies for partially labeled sequences," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 898–906.

[36] E. R. Fernandes and U. Brefeld, "Learning from partially annotated sequences," in *Machine Learning and Knowledge Discovery in Databases*, D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, Eds. Berlin, Germany: Springer, 2011, pp. 407–422.

[37] Q. Wu, M. K. Ng, and Y. Ye, "Markov-Miml: A Markov chain-based multi-instance multi-label learning algorithm," *Knowl. Inf. Syst.*, vol. 37, no. 1, pp. 83–104, 2013.

[38] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[39] J. Lafferty *et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2001, pp. 282–289.

[40] G. Durrett and D. Klein. (2015). "Neural CRF parsing." [Online]. Available: https://arxiv.org/abs/1507.03641

[41] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. ICCV*, Dec. 2015, pp. 1529–1537.

[42] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.

[43] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016, doi: 10.1109/TPAMI.2015.2505283.

[44] G. Lin, C. Shen, A. van den Hengel, and I. Reid, "Exploring context with deep structured models for semantic segmentation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 40, no. 6, pp. 1352–1366, Jun. 2017.

[45] T.-M.-T. Do and T. Artieres, "Neural conditional random fields," in *Proc. AISTATS*, 2010, pp. 177–184.

[46] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *Proc. ICASSP*, May 2014, pp. 4077–4081.

[47] L. Zhang, M. Mahdavi, R. Jin, T. Yang, and S. Zhu, "Random projections for classification: A recovery approach," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 7300–7316, Nov. 2014.

[48] J. Kupiec, "Robust part-of-speech tagging using a hidden Markov model," *Comput. Speech Lang.*, vol. 6, no. 3, pp. 225–242, 1992.

[49] W. Ammar, C. Dyer, and N. A. Smith, "Conditional random field autoencoders for unsupervised structured prediction," in *Proc. NIPS*, 2014, pp. 3311–3319.

[50] C.-C. Lin, W. Ammar, C. Dyer, and L. Levin. (2015). "Unsupervised POS induction with word embeddings." [Online]. Available: https://arxiv.org/abs/1503.06760

[51] Z. Marinho, A. F. Martins, S. B. Cohen, and N. A. Smith, "Semi-supervised learning of sequence models with method of moments," in *Proc. EMNLP*, 2016, pp. 287–296.

[52] O. Täckström, D. Das, S. Petrov, R. McDonald, and J. Nivre, "Token and type constraints for cross-lingual part-of-speech tagging," *Trans. Assoc. Comput. Linguistics*, vol. 1, pp. 1–12, 2013.

[53] X. Zhang, Y. Jiang, H. Peng, K. Tu, and D. Goldwasser, "Semi-supervised structured prediction with neural CRF autoencoder," in *Proc. EMNLP*, 2017, pp. 1702–1712.

[54] J. Verbeek and B. Triggs, "Scene segmentation with conditional random fields learned from partially labeled images," in *Proc. NIPS*, 2007, pp. 1–8.

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

[56] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNS-CRF," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1064–1074. [Online]. Available: http://www.aclweb.org/anthology/P16-1101

[57] K. Cho *et al.* (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation." [Online]. Available: https://arxiv.org/abs/1406.1078

[58] A. Taylor, M. Marcus, and B. Santorini, "The penn treebank: An overview," in *Treebanks*. Dordrecht, The Netherlands: Springer, 2003, pp. 5–22.

[59] E. F. T. K. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proc. 7th Conf. Natural Lang. Learn. (HLT-NAACL)*, vol. 4, 2003, pp. 142–147.

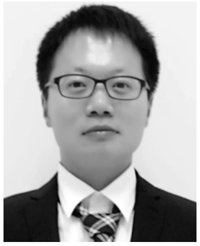
[60] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[61] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

Joey Tianyi Zhou received the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2015.

He is currently a Scientist with the Institute of High Performance Computing, Research Agency for Science, Technology, and Research, Singapore. His current research interests include differentiable programming, transfer learning, and sparse coding.

Dr. Zhou was a recipient of the Best Poster Honorable Mention at ACML 2012, the Best Paper Award from the BeyondLabeler Workshop on IJCAI 2016, the Best Paper Nomination at ECCV 2016, and the NIPS 2017 Best Reviewer Award.



Meng Fang received the Ph.D. degree from the University of Technology Sydney, Ultimo NSW, Australia.

He was a Research Fellow with the School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC, Australia. He is currently a Senior Researcher with Tencent AI Lab, Shenzhen, China. His current research interests include machine learning and natural language processing.

Hao Zhang received the B.S. degree in communications engineering from the Dalian University of Technology, Dalian, China, in 2015, and the M.S. degree in communications engineering from Nanyang Technological University, Singapore, in 2016.

He is currently a Research Engineer with Artificial Intelligence Initiative, Agency of Science, Technology, and Research, Singapore.

Chen Gong received the B.E. degree from the East China University of Science and Technology, Shanghai, China, in 2010, and the dual Ph.D. degree from Shanghai Jiao Tong University, Shanghai, and the University of Technology Sydney, Ultimo NSW, Australia, in 2016 and 2017, respectively.

He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His current research interests include machine learning, data mining, and learning-based vision problems.

Xi Peng received the Ph.D. degree in computer science from Sichuan University, Chengdu, China, in 2013.

He currently is a Research Professor with the College of Computer Science, Sichuan University. His current research interests include unsupervised representation learning and differentiable programming, as well as their applications in computer vision and image processing. In these areas, he has authored over 40 papers.

Zhiguo Cao received the B.S. and M.S. degrees in communication and information system from the University of Electronic Science and Technology of China, Chengdu, China, and the Ph.D. degree in pattern recognition and intelligent system from the Huazhong University of Science and Technology, Wuhan, China.

He is currently a Professor with the School of Automation, Huazhong University of Science and Technology. His current research interests include image understanding and analysis, depth information extraction, 3-D video processing, motion detection, and human action analysis.

Rick Siow Mong Goh received the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore.

He is currently the Director of the Computing Science Department, Institute of High Performance Computing, Agency of Science, Technology, and Research, Singapore. His current research interests include artificial intelligence, high-performance computing, distributed computing, machine and deep learning, complex systems, human–machine interaction, and modeling and simulation.