RoSeq: Robust Sequence Labeling

Joey Tianyi Zhou, Hao Zhang, Di Jin, Xi Peng, Yang Xiao, and Zhiguo Cao

***Abstract*— In this paper, we mainly investigate two issues for sequence labeling, namely, label imbalance and noisy data that are commonly seen in the scenario of named entity recognition (NER) and are largely ignored in the existing works. To address these two issues, a new method termed robust sequence labeling (RoSeq) is proposed. Specifically, to handle the label imbalance issue, we first incorporate label statistics in a novel conditional random field (CRF) loss. In addition, we design an additional loss to reduce the weights of overwhelming easy tokens for augmenting the CRF loss. To address the noisy training data, we adopt an adversarial training strategy to improve model generalization. In experiments, the proposed RoSeq achieves the state-of-the-art performances on CoNLL and English Twitter NER—88.07% on CoNLL-2002 Dutch, 87.33% on CoNLL-2002 Spanish, 52.94% on WNUT-2016 Twitter, and 43.03% on WNUT-2017 Twitter without using the additional data.**

***Index Terms*— Label imbalance, named entity recognition (NER), sequence labeling.**

I. Introduction

EXISTING multioutput learning mainly aims at determining multiple outputs for a given input. In many cases, the output often involves a structure that is helpful to the training models, e.g., sequences, strings, trees, lattices, or graphs. In order to infer the structured outputs from an observation sequence rather than a data point, label sequence learning or sequence labeling has been widely studied, where the output sequence has inherent interconnections rather than a simple concatenation of individual units. It is also an important step in the most natural language processing (NLP) applications and has been applied to numerous real-world tasks including but not limited to part-of-speech (POS) tagging [1], named entity recognition (NER) [2], [3], and speech recognition [4].

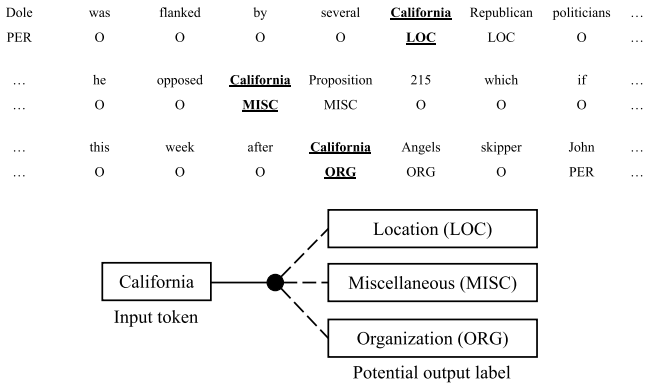

Fig. 1. Examples of ambiguous labels in CoNLL-2003 English NER data set: the token "California" in the first sentence is labeled as Location (*LOC*), whereas it is labeled as Miscellaneous (*MISC*) and Organization (*ORG*) in the second and third sentences, respectively.

In these tasks, NER has attracted increasing interests, which detects not only the type of named entity but also the entity boundaries. To disambiguate different entity types of same tokens, NER needs a deep understanding of the contextual semantics. As shown in Fig. 1, for example, the token "California" in the first sentence is labeled as location (*LOC*), whereas it is labeled as miscellaneous (*MISC*) and organization (*ORG*) in the second and third sentences, respectively. To tackle this challenging problem, a variety of methods have been proposed, which are usually based on hand-crafted rules and have suffered from limited performance in practice [5]–[7]. Recent works have devoted to developing learning-based algorithms, especially neural network-based methods, which have remarkably advanced the state of the arts [1], [2], [8]–[11]. Furthermore, these end-to-end models generalize well on new entities based on features automatically learned from data.

Although deep learning-based methods have achieved significant progress, they ignore two important and commonly seen properties of existing NER data sets as follows.

1) The label distribution is extremely imbalanced. Take the CoNLL-2002 Dutch NER data set as a showcase [Fig. 2(a)], 90.48% data are with unmeaning entity "O." Similarly, for the WNUT-2017 English Twitter NER data set, the label of "O" even occupies 94.62% of the entire data. [see Fig. 2(b)].
2) The data set is noisy. Like the WNUT data set collected from user-generated tweets, the NER data set is collected

Manuscript received November 1, 2018; revised February 20, 2019; accepted April 5, 2019. This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1004600, in part by Singapore Government's Research, Innovation and Enterprise 2020 Plan (Advanced Manufacturing and Engineering domain) under Grant A1687b0033 and Grant A18A1b0045, in part by the Fundamental Research Funds for the Central Universities under Grant YJ201748, and in part by NFSC under Grant 61806135, Grant 61625204, Grant 61836006, Grant 61876211, and Grant 61602193. *(Joey Tianyi Zhou and Hao Zhang contributed equally to this work.) (Corresponding author: Yang Xiao.)*

J. T. Zhou is with the Institute of High Performance Computing, A*STAR, Singapore 138632 (e-mail: zhouty@ihpc.a-star.edu.sg).

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

D. Jin is with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 USA (e-mail: jindi15@mit.edu).

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

Y. Xiao and Z. G. Cao are with the National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: Yang_Xiao@hust.edu.cn; 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.2019.2911236

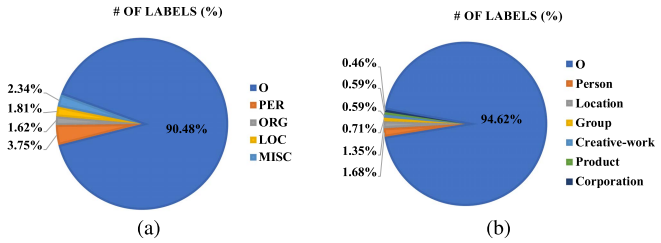

Fig. 2. Label statistics on CoNLL-2002 Dutch NER and WNUT-2017 English Twitter NER train set. The label type "O" is dominant in different data sets. (a) CoNLL-2002 Dutch. (b) WNUT-2017 English Twitter.

TABLE I
F1 SCORE OF BILSTM-CNNS-CRF MODEL ON WNUT-2017 DEVELOPMENT SET

Label Type	Precision (%)	Recall (%)	F1 Score (%)
O	95.56	99.46	97.47
Person	78.75	49.23	60.59
Location	51.02	46.73	48.78
Group	19.23	7.81	11.11
Creative-work	32.94	11.76	17.34
Product	33.33	7.69	12.50
Corporation	18.18	21.74	19.80

from the social media and thus contains lots of noise, such as emojis, punctuation, incomplete or misspelled words, urls, unknown tokens, and so on. For example, here is a sentence from tweets in WNUT data set, *"Soooo Glad its Finally Friday !!!! :P Thissss week went to slow fo'me :/."* Here, *"Soooo"* is misspelled and should be *"So."* *"fo'me"* should be *"for me."*

The first label imbalance issue severely affects the convergence performance of the NER models. In practice, the performance of these neural-based methods degrades significantly since the learned models favor the overwhelming label types. Here, we train one state-of-the-art (SOTA) neural network-based NER model (i.e., bidirectional long short term memory (LSTM) (BiLSTM)-convolutional neural networks (CNNs)-conditional random field (CRF) [10]) on WNUT-2017 data set and report its F1 scores of different entity types on the development set given in Table I. Note that for the NER task, the F1 score of unmeaning entity "O" does not be considered in the final result. However, we only compute F1 score of "O" for comparing with the rest meaningful entity types. We observe that the dominant "O" label achieves more than 97% F1 score, whereas the other meaningful entity types such as "Group" and "Product" only achieve around 10% F1 score. In the experiment, we also find that those data with "O" label can be correctly classified with very high confidence. Unfortunately, in the NER task, we only care about the performance of those meaningful token types rather than the nonmeaningful "O" entity.

Most existing works ignore the noisy data issue and usually assume that the NER data are clean. In consequence, the noisy and untrimmed data affects performance significantly. For example, the F1 score on the clean NER data set such as CoNLL data set collected from news could easily reach over than 80%. In contrast, the noisy data sets such as WNUT are collected from Twitter, which is able to reach around 50% only. To the best of our knowledge, there are only few works that explicitly discuss and give an in-depth analysis of how to handle these noisy data in NER.

Based on the above-mentioned observations, we propose a novel learning algorithm, termed as **robust sequence labeling (RoSeq)**, which focuses on how to handle label imbalance and noisy data issues in a single framework. Specifically, to handle the label imbalance issue, we propose a new *label-aware CRFs (LACRF)* algorithm which assigns different labels with different weights. Furthermore, inspired by the success of the focal loss (FL) [12] in solving object detection problem, we assign confidence-dependent weights to different samples so that the model will not be overwhelmed by such as the samples with "O" entity type in NER data sets. In other words, we add the FL to the aforementioned LACRF loss as the overall loss. On the other hand, to address the noisy data issue, we adopt the *adversarial training* (AT) to virtually create adversarial examples to improve the robustness of model training. The contributions of this paper could be summarized as follows.

1) To the best of our knowledge, this could be the first work to explicitly discuss the label imbalance and noisy data issues for sequence labeling. We find that the application of NER generally encounters these two issues.
2) We propose RoSeq to alleviate the overfitting and non-robust training problem caused by the label imbalance and noisy data. Specifically, RoSeq employs a word-level BiLSTM on the top of character-level and word embedding features to learn relevant high-level features in an end-to-end manner. Furthermore, a new objective function is proposed for RoSeq.
3) Extensive experimental results show the remarkable improvements over the SOTA NER models, especially in the case of imbalanced label. In brief, our method achieves SOTA performance on a series of NER benchmark data sets, namely, 87.33% F1 for CoNLL-2002 Spanish, 88.07% F1 for CoNLL-2002 Dutch, 52.94% F1 for WNUT-2016, and 43.03% F1 for WNUT-2017.

II. RELATED WORK

Our work is related to the following topics, i.e., NER and robust learning that are briefly introduced in this section.

A. Named Entity Recognition

NER is typically framed as a sequence labeling task that aims at automatic detection of named entities (e.g., person, organization, and location) from free text [13]. The early works are usually based on the hidden Markov model (HMM) [14], support vector machine (SVM) [15], CRF [16], or perception models with the hand-crafted features [5]–[7]. More specifically, they employed a system to read a large-scale annotated training data, memorize a list of entities, and create disambiguation rules based on discriminative features. Among them, CRFs and HMMs are very effective and have achieved SOTA performance in handling sequential data [1], [16]. Comparing with generative models (e.g., HMMs), discriminative models (e.g., CRFs) are usually more powerful, which aims at 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 NLP and biological sequence analysis.

Recent researches have shifted towards deep neural networks (DNNs) [2], [17]–[19]. Collobert *et al.* [8] proposed a feed-forward neural network with a fixed sized window for each word, but it failed in considering useful relations between long-distance words. To overcome this limitation, Chiu and Nichols [9] designed a BiLSTM-CNNs architecture that automatically detects word- and character-level features. Ma and Hovy [10] further extended [9] into BiLSTM-CNNs-CRF architecture wherein the CRF module is added to optimize the output label sequence. Liu *et al.* [20] proposed a task-aware neural language model (LM) termed as LM-LSTM-CRF which incorporates character-aware neural LMs to extract character-level embedding under a multitask framework. Yang *et al.* [21] proposed a transfer learning approach based on a deep hierarchical recurrent neural network (RNN) that utilizes the information from different lingual/domain data sets by fully or partially sharing the model parameters among different tasks. Ni and Florian [22] and Ni *et al.* [23] utilizes the Wikipedia entity-type mappings to improve low-resource NER. In addition, some recent works [24]–[29] show that multitask learning with joint training on multiple tasks/languages could improve its performance. Most recently, the pretrained LMs have also shown effectiveness in improving the performance of many NLP tasks [30], [31]. In this paper, we focus on solving the label imbalance and noisy data issues, thus we will not use any external resources nor knowledge transfer and will not investigate the performance with resource boosting methods.

B. Robust Learning

Imbalanced distribution is commonly seen in many nonNLP applications, especially in computer vision tasks such as object detection [12], segmentation [32], clustering [33], [34], and dimension reduction [35]. The fundamental issue with the imbalanced learning problem is the ability of the imbalanced data to significantly compromise the performance of most standard learning algorithms. There are many solutions [36] proposed to address this issue. Sampling methods [37] for imbalanced learning is one direction. Typically, the use of sampling methods in imbalanced learning applications consists of the modification of an imbalanced data set by some mechanisms in order to provide a balanced distribution. Sampling methods attempt to balance distributions by considering the representative proportions of class examples in the distribution. Studies [37] have shown that a balanced data set could improve the overall classification performance compared to an imbalanced data set for several base classifiers. Cost-sensitive methods [38] for imbalanced learning is another branch, which consider the costs associated with misclassifying or overwhelming easy samples [36]. Although the class imbalance problem has been discussed for years, the investigation of it on the task of NER is still limited.

To achieve robustness from noisy data, early works eliminate the effectiveness of noise using feature selection [39] and stochastic optimization [40]. In recent, AT [41], [42] is proposed to improve the robustness of image classification model by injecting malicious perturbations into input images. In the NLP community, Miyato *et al.* [43] proposed a semi-supervised text classification method by applying AT, where for the first time adversarial perturbations were added onto word embeddings. Yasunaga *et al.* [44] applied AT to POS tagging. Although these methods achieve some improvements by deploying AT, their analysis is insufficient for noisy data. In this paper, we provide a more in-depth analysis why AT is workable on the noisy NLP data.

III. Robust Sequence Labeling

A. Character-Level Feature Representation

Previous works have shown that character features can improve the NER performance by capturing morphological and semantic information [29]. In practice, character-level representation learning method could be roughly classified into two subjects: one is character-level BiLSTM (Char-BiLSTM) [2] and the other is character-level CNN (Char-CNN) [9], [10]. Reimers and Gurevych [45] have observed that the performance difference between the Char-BiLSTM and Char-CNN approaches is statistically insignificant on sequence labeling tasks, but Char-CNN approach has less parameters and is more competitive in running time and computation resources. Thus, our method adopts the Char-CNN approach as the character-level encoder to learn character representation for each word.

To build the character-level feature representation encoder, let $\mathcal{C}$ and d_c denote the character vocabulary and the dimension of character embeddings, respectively. Let the word w consist a sequence of characters $[c_1, c_2, \ldots, c_n]$, its character-level representation is denoted by the matrix $\mathbf{C}^w \in \mathbb{R}^{d_c \times n}$, where n is the length of word w. Then, the Char-CNN is designed using multiple filters with different widths to learn the character-level feature for word w. Typically, let $\mathbf{F} \in \mathbb{R}^{d_c \times k}$ be a convolutional filter with width k and $\mathbf{b}_F$ be the bias term, the feature map $\mathbf{f}^w \in \mathbb{R}^{n-k+1}$ is obtained by applying a convolution between $\mathbf{C}^w$ and $\mathbf{F}$ as follows:

$$\mathbf{f}^w[i] = \tanh(\langle \mathbf{C}^w[:, i : i + k - 1], \mathbf{F} \rangle_F + \mathbf{b}_F) \tag{1}$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product and $\mathbf{f}^w[i]$ represents the ith element of $\mathbf{f}^w$. With the following formula, we use the max-pooling to capture the most important feature as the representation of w corresponding to the kernel $\mathbf{F}$

$$e^w = \max_i \mathbf{f}^w[i]. \tag{2}$$

Since our Char-CNN contains multiple filters, suppose we have m filters $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m$, thus, the final character-level feature representation of word w is $\mathbf{e}^w = [e_1^w, e_2^w, \ldots, e_m^w]$. In other words, the concatenation of features is obtained by different filters.

Instead of using the obtained character-level feature representation $\mathbf{e}^w$ of word w for the further process directly, Kim *et al.* [46] indicate that running $\mathbf{e}^w$ through the highway networks is able to obtain the improvements on LMs. The highway network [47] optimizes the neural networks and increases their depth by using the learned gating mechanisms

to regulate information flow. In other words, similar to the memory cell in LSTM, highway networks allow for training of deep networks by adaptively carrying some dimensions of the input directly to the output [46].

Formally, highway network consists of a feed-forward layer and two nonlinear transformations (i.e., transform gate and carry gate), where the former applies an affine transform followed by a nonlinear activation function to learn new features and the transformations express how much of the output is produced by transforming the input and carrying it. By applying highway network on the character-level feature representation, we have

$$\mathbf{z} = \mathbf{t} \odot g(\mathbf{W}_g \mathbf{e}^{w} + \mathbf{b}_g) + (1 - \mathbf{t}) \odot \mathbf{e}^{w} \tag{3}$$

where g is a nonlinear activation function. In our implementation, we use the tanh function. $\mathbf{t} = \sigma(\mathbf{W}_T \mathbf{e}^{w} + \mathbf{b}_T)$ is the transform gate, $1 - \mathbf{t}$ denotes the carry gate, σ denotes the Sigmoid activation function, and $\mathbf{z}$ is the output of highway network. As the dimensions of $\mathbf{e}^{w}$ and $\mathbf{z}$ are required be consistent, $\mathbf{W}_g$ and $\mathbf{W}_T$ are two square matrices. Generally, we adopt two layers of highway network at the top of Char-CNN and denote $\mathbf{e}^{\text{char}}$ as the final character-level feature representation of a word. The structure of the character-level encoder is shown in Fig. 4.

In this section, we introduce the proposed RoSeq model in details. Fig. 3 shows the architecture of the RoSeq model, which consists of character-level encoder, word-level encoder, label decoder, and adversarial training module.

B. Word-Level Feature Representation

To learn a better word-level representation, we concatenate character-level features of each word with a latent word embedding as $\mathbf{e}_i = [\mathbf{e}_i^{\text{char}}, \mathbf{e}_i^{\text{word}}]$, where the latent word embedding $\mathbf{e}_i^{\text{word}}$ is initialized with pretrained embeddings and fixed during training. One unique characteristic of NER is that the historical and future input for a given time step could be useful for label inference. To exploit such a characteristic, we use a BiLSTM [48] to extract contextualized word-level features. LSTM is a variant of RNN, which is capable of learning long-term dependencies and coping with the gradient vanishing and exploding problems. Basically, the LSTM unit is similar to the RNN unit, except that the hidden layer updates are replaced by purpose-built memory cells to control the proportions of information to forget and to pass to the next time step.

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

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

where σ is the Sigmoid activation function and $\odot$ represents the elementwise product. $\mathbf{i}$, $\mathbf{f}$, and $\mathbf{o}$ are the input gate, forget gate, and output gate, respectively. $\mathbf{e}_t$ represents the input instance at the time stamp t and $\mathbf{h}_t$ is the corresponding hidden state (also called output) at time of t. $\mathbf{W}_*$ denotes the weight for hidden state $\mathbf{h}_t$, $\mathbf{U}_*$ represents the weight of different gates for input $\mathbf{e}_t$, and $\mathbf{b}_*$ denotes the bias.

The hidden state $\mathbf{h}_t$ 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 BiLSTM that represents the input sequence forwards and backwards as two separate hidden states so that the past and future information are learned. Specifically, let $\mathbf{h}_t = \texttt{lstm}(\mathbf{h}_{t-1}, \mathbf{e}_t)$ denote the hidden state update process of unidirectional LSTM for a particular time frame. In this way, for the BiLSTM, we have

$$\overrightarrow{\mathbf{h}}_t = \texttt{lstm}(\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{e}_t) \tag{4}$$

$$\overleftarrow{\mathbf{h}}_t = \texttt{lstm}(\overleftarrow{\mathbf{h}}_{t+1}, \mathbf{e}_t) \tag{5}$$

where $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$ are the learned left context (forward) and right context (backward) representations, respectively. After the BiLSTM layer, the final representation of a word is obtained by fusing its left and right context representations, we apply a nonlinear layer for this process as

$$\mathbf{h}_t = \texttt{tanh}(\mathbf{W}_l \overrightarrow{\mathbf{h}}_t + \mathbf{W}_r \overleftarrow{\mathbf{h}}_t + \mathbf{b}). \tag{6}$$

Note that, the representation integrates both local and global information, which could capture contextually sensitive signals across sequences.

C. Label Decoder

For the sequence labeling tasks, dependencies or correlations between labels in neighborhoods are crucial to disambiguate different entity types of each word. Therefore, it is promising and helpful to utilize the correlations and decode the best chain of labels for a given input sequence so that the resulting label sequence could be meaningful. For instance, in the NER task with standard BIO2 annotation scheme [49], *I-LOC* is illegal to follow *B-ORG* (mixing different annotation types), *B-PER* cannot follow another *B-PER* (wrong annotation dependence), and so on. CRF is a widely used method to make joint labeling of the tokens in a sequence [16], hence, we use a linear-chain CRF as label decoder to capture the relationships of labels and model the label sequence jointly, instead of predicting each label independently.

In the CRF decoder, there are two kinds of cliques, namely, local cliques and transition cliques, where local cliques correspond to the individual elements in the sequence, whose representation is $\mathbf{h}_t$ as defined in (6), and transition cliques, on the other hand, reflect the evolution of states between the neighboring elements. Formally, let $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T\}$ represents a generic representation sequence where $\mathbf{h}_t$ is the representation vector of the tth word and $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$ denotes a generic sequence of labels for $\mathbf{h}$, hence the probabilistic model for linear-chain CRF defines a family of conditional probability $p(\mathbf{y}|\mathbf{h})$ over all possible label sequences $\mathbf{y}$ given $\mathbf{h}$ can be written as

$$p(\mathbf{y}|\mathbf{h}) = \frac{\exp\left\{\sum_{t=2}^{T} \theta_{\mathbf{y}_{t-1}, \mathbf{y}_t} + \sum_{t=1}^{T} (\mathbf{W}_{\mathbf{y}_t} \mathbf{h}_t + \mathbf{b}_{\mathbf{y}_t})\right\}}{Z(\mathbf{h})} \tag{7}$$

where $Z(\mathbf{h})$ 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 $(\mathbf{y}_i, \mathbf{y}_j)$.

Despite that linear-chain CRF takes the correlations between subsequent labels into consideration, however, it does not solve the problem of label imbalance in the sequence. In order to address this issue, we introduce a label-aware weight $\boldsymbol{\zeta} \in \mathbb{R}^{|L|}$, where $|L|$ is the number of unique labels. The label-aware weight is a resource specific weight vector, which is derived from the statistical distribution of various label types in the resource data set, where a label with higher frequency tends to have lower weight. The idea of introducing this label-aware weight is to suppress the effect of massive but unmeaning label, i.e., "O" label, by reducing its contribution and increasing the importance of other meaningful labels while jointly decoding the best chain of label sequence in the CRF. Formally, we first calculate the label proportion for each label. Let $\vartheta_i \in \mathbb{R}^{|L|}$ denote the label proportion of label Y_i (e.g., $Y_i \in \{\text{O}, \text{PER}, \text{LOC}, \text{ORG}, \text{MISC}\}$), the label-aware weight vector $\boldsymbol{\zeta}$ is computed as

$$\boldsymbol{\zeta} = \left(\frac{\max_{i \in |L|} \vartheta_i}{\boldsymbol{\vartheta}} \right)^{\tau} \tag{8}$$

where $\boldsymbol{\vartheta} = [\vartheta_1, \vartheta_2, \cdots, \vartheta_L]^\top$ and τ is a scalar to adjust the magnitude of weight values. In this way, if the label proportion is higher, then the weight contribution is smaller. Hence, by employing the label-aware weight into the CRF, termed as LACRF, the (7) can be rewritten as

$$\tilde{p}(\mathbf{y}|\mathbf{h}) = \frac{\exp\left\{ \sum_{t=2}^{T} \theta_{\mathbf{y}_{t-1}, \mathbf{y}_t} + \sum_{t=1}^{T} \boldsymbol{\zeta} \cdot (\mathbf{W}_{\mathbf{y}_t} \mathbf{h}_t + \mathbf{b}_{\mathbf{y}_t}) \right\}}{\tilde{Z}(\mathbf{h})} \tag{9}$$

while $\tilde{Z}(\mathbf{h})$ is the corresponding normalization term after introducing the label-aware weight vector $\boldsymbol{\zeta}$. Our objective is to maximize the conditional log-likelihood estimation, which is also equivalent to minimize the negative log-likelihood, and the negative logarithm of the likelihood is defined as

$$\ell_{\text{LACRF}} = -\sum_i \log \tilde{p}(\mathbf{y}|\mathbf{h}). \tag{10}$$

The LACRF can alleviate the label-imbalance; however, it does not differentiate between easy and hard samples, which is still remained to be addressed. Despite the correlations between labels in NER task and by treating NER as a traditional classification task where each input token has the corresponding class, it is obvious that "O" class is dominant and other classes only occupy extremely small part in the NER data set. According to Table I, the dominating data with "O" class, termed easy samples, achieves notably higher confidence compared with other rare classes. This may be caused by the fact that the training procedure is still dominated by the easily classified examples without sufficient valuable learning signal, which leads to degenerated models.

This issue is also commonly seen in the object detection task and Lin *et al.* [12] proposed a simple but effective approach, namely, FL to address it. The main idea is to reshape the loss function to down-weight overwhelming easy samples (i.e., background objects) and thus focus training on hard samples (i.e., foreground objects). Similar to background objects in object detection, in NER, the data with "O" label: 1) occupies the overwhelming portion in the training phrase; 2) are classified with very high accuracy (around 97% F1 score); and 3) are not considered in the evaluation. Motivated by the effectiveness of FL for handling the object detection, we apply it to address the easy/hard sample issue in NER task. For the convenience of presentation, we here only present the binary case, and it is easy to be applied to the multiclass scenario. Formally, we first define

$$q_t = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if otherwise} \end{cases}$$

where $y \in \{1, -1\}$ specifies the ground-truth class and $q \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$ based on the softmax of the word-level fusion features (Fig. 3). The FL is to add a modulating factor $(1-q_t)^\gamma$ to the cross-entropy loss of the classification task, with tunable focusing parameter $\gamma \geq 0$. The definition of FL is given as

$$\ell_{\text{FL}} = \sum_i -(1 - q_t)^\gamma \log(q_t). \tag{11}$$

In this way, when the input data are misclassified and q_t is small, the modulating factor is near 1 and the loss is unaffected. As $q_t \to 1$, the factor goes to 0 and the loss for well-classified examples is down-weighted.

So far we combine the LACRF loss and label-balanced cross entropy loss as

$$\ell = \ell_{\text{LACRF}} + \ell_{\text{FL}} \tag{12}$$

where ℓ_{LACRF} and ℓ_{FL} play regularization roles on the label level and the instance level, respectively. The model can be trained end-to-end with standard back-propagation by minimizing the ℓ.

D. Adversarial Training

In the computer vision community, a lot of experiments [50] have demonstrated the fragility of deep learning models to *adversarial examples* [42], which are created by changing a very small proportion of pixels. Those adversarial examples and original examples are virtually indistinguishable to human perception [51]. Recently, adversarial samples are wisely incorporated into training to improve the generalization and robustness of the model, which is so-called AT [43]. It emerges as a powerful regularization tool to stabilize training and prevent the model from being stuck in a local minimum.

As the discussion in the introduction, the noisy data contain a lot of abbreviations, emojis, and so on, which harm the NER performance significantly. The main reason is that the pretrained features such as word2vec are trained from clean data and they are not helpful to represent those noisy data such as emojis and misspelled words. To handle this challenge, in this paper, we explore AT. Specifically, we construct an adversarial sample by compounding the original sample with a perturbation bounded by a small norm ϵ to maximize the loss function as follows:

$$\hat{\eta} = \arg \max_{\eta: \|\eta\|_2 \leq \epsilon} \ell(\Theta; \mathbf{x} + \eta) \tag{13}$$

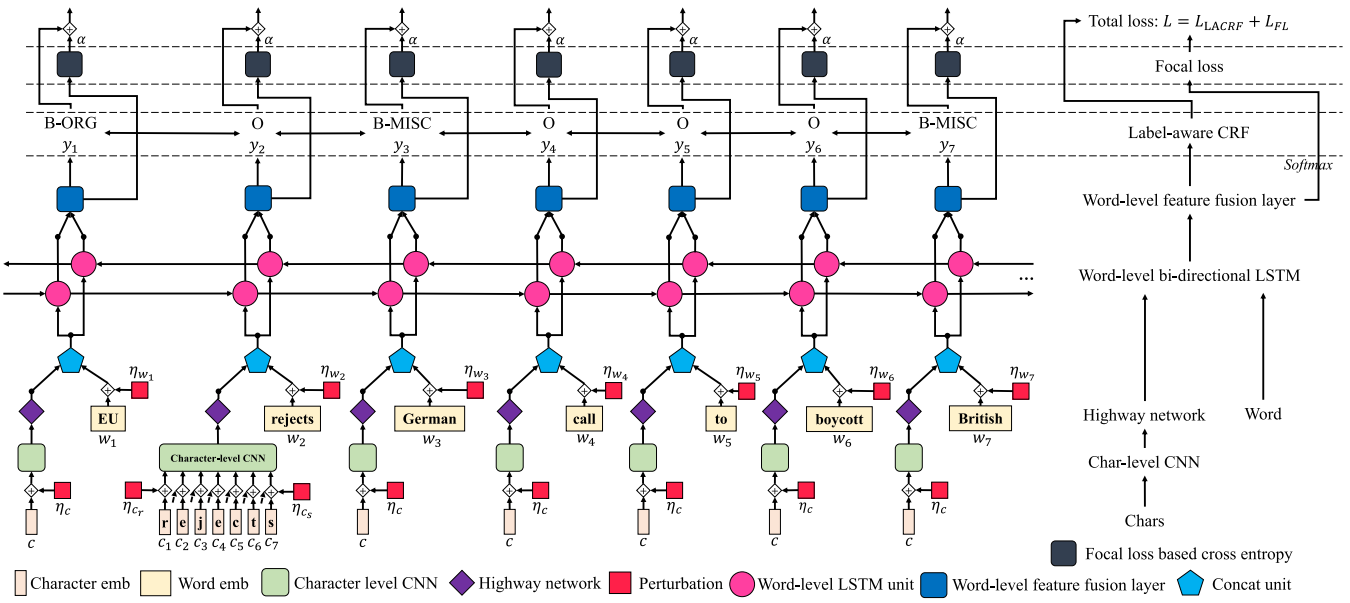

Fig. 3. Architecture of RoSeq.

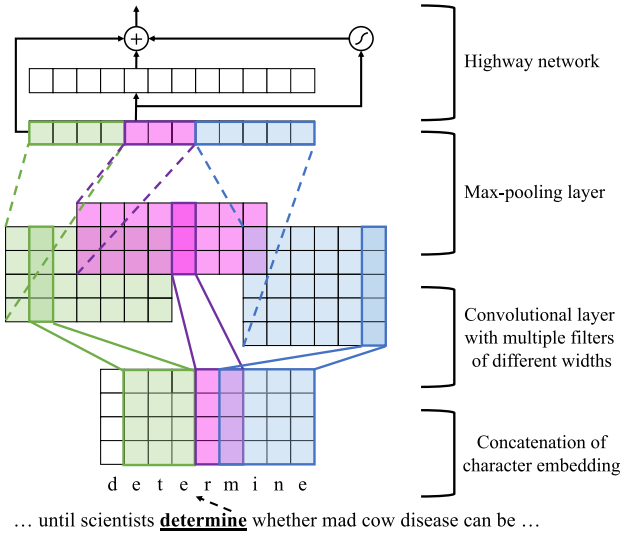

Fig. 4. Character-level feature representation with convolutional and highway networks.

where Θ denotes the current model parameters set. By adding loss-aware perturbation during training, the model becomes more resistant to noisy data.

Unfortunately, we are unable to compute η in (13) exactly since the exact optimization with respect to η is unfeasible. Following the strategy in [42], this value can be approximated by linearizing it as follows:

$$\hat{\eta} = \epsilon \frac{\mathbf{g}}{\|\mathbf{g}\|_2}, \quad \text{where } \mathbf{g} = \nabla \ell(\Theta; \mathbf{x}) \tag{14}$$

where ϵ can be determined on the validation set. In this way, adversarial examples are generated by adding small perturbations to the inputs in the direction that most significantly increases the loss function of the model. We find such η against the current model parameterized by Θ, at each training step, and construct an adversarial example by

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \hat{\eta}. \tag{15}$$

Noted that we generate this adversarial example on both the word and character embedding layer, respectively, as shown in Fig. 3.

Then, the classifier is trained on the mixture of original and adversarial examples to improve the generalization. To this end, we augment the loss in (12) and define the loss function for AT as

$$\ell_{AT} = \ell(\Theta; \mathbf{x}) + \ell(\Theta; \mathbf{x}_{\text{adv}}) \tag{16}$$

where $\ell(\Theta; \mathbf{x})$ and $\ell(\Theta; \mathbf{x}_{\text{adv}})$ represent the loss from an original example and its adversarial counterpart, respectively. Note that we present the AT in a general form for the convenience of presentation. For different samples, the loss and parameters should correspond to their counterparts. For example, we can compute the perturbations $\eta_{\mathbf{c}}$ for char-embedding and $\eta_{\mathbf{w}}$ for the word embedding.

IV. Experiments

A. Data Sets

To verify the effectiveness of our method, we conduct the experiments on the following widely used NER data sets: CoNLL-2002 Dutch & Spanish NER [52], CoNLL-2003 English NER [53], and WNUT-2016/17 English Twitter NER [54], and we use begin, inside, outside, end, single tagging scheme [2], [5], [10], [55] in our experiments. For CoNLL data sets, there are four different types of named entities: *Location (LOC)*, *Person (PER)*, *Organization (ORG)*, and *Miscellaneous (MISC)*. The WNUT-2016 data set contains ten types of named entities, whereas WNUT-2017 data set has six types, and *O* is used as unmeaning label for all of those data sets. The statistics of the data sets are described in Table II. Note that different from CoNLL data sets, WNUT is the annotated NER data sets on the noisy user-generated text, e.g., tweets.

TABLE II
STATISTICS OF NER DATA SETS

Benchmark	Language	# of Training Tokens (# of Entities)	# of Dev Tokens (# of Entities)	# of Test Tokens (# of Entities)
CoNLL-2002	Spanish	207,484 (18,797)	51,645 (4,351)	52,098 (3,558)
CoNLL-2002	Dutch	202,931 (13,344)	37,761 (2,616)	68,994 (3,941)
CoNLL-2003	English	204,567 (23,499)	51,578 (5,942)	46,666 (5,648)
WNUT-2016	English	46,469 (2,462)	16,261 (1,128)	61,908 (5,955)
WNUT-2017	English	62,730 (3,160)	15,733 (1,250)	23,394 (1,740)

B. Experimental Setup

We use publicly available pretrained word embeddings for English, Spanish, and Dutch languages in our experiments. For English, we choose the 100-dimensional GloVe [56] word embeddings, which is trained on Wikipedia-2014 and Gigaword-5,[1] whereas for Spansih and Dutch, we use the pretrained 50-dimensional word embeddings are trained using the word2vec package[2] on the corresponding Wikipedia articles (2017-12-20 dumps) [29].

Specifically, we also use the orthographic encoder [27], [57], [58] for characters and words while training on the WNUT-2016/17 data sets for fair comparison. The orthographic encoder is used to encapsulate capitalization, punctuation, word shape, and other orthographic features. Let "n" denotes number, "c" is a letter ("C" if capitalized), punctuation is marked as "p." Then, for example, given a sentence "I don't like 13!," its corresponding orthographic representation is "C cccpc cccc nnp." In the experiment, we represent each orthographic character and orthographic word with 30-dimensional and 50-dimensional randomly initialized vectors, respectively. Then those features are concatenated to their corresponding words or characters.

For model hyperparameters, we use three filters with widths $[2, 3, 4]$ for Char-CNN encoder and set each filter number as 20, the dimension of hidden states of word-level BiLSTM is 100, τ in the label-aware weight is set to 0.25, γ in the FL is fixed to 2.0, and the ϵ of AT is fixed to 5.0 in our experiment. Batch size is set as 16 for all experiments. The parameters optimization is performed by Adam optimizer [59] with gradient clipping of 5.0 to avoid gradient exploding problem and learning rate decay strategy. We choose the initial learning rate of $\beta_0 = 0.001$ for all experiments. At each epoch t, learning rate β_t is updated using $\beta_t = \beta_0/(1+\rho \times t)$, where ρ is the decay rate with value 0.05, we also set a minimal learning rate $\beta_{\min} = 1e^{-4}$ and let $\beta_t = \beta_{\min}$ if $\beta_t < \beta_{\min}$. To reduce the overfitting issue, we also apply dropout mechanism [60] to the word and character embedding layers with drop rate 0.2 and the output of the BiLSTM layer with drop rate 0.5, respectively.

For the base model, we create a standard BiLSTM-CNNs-CRF model that simply removes the FL, AT, and LACRF components, and train this model on each data set. All the hyperparameter settings and parameters optimization strategies of this base model are the same as the aforementioned setup, except that we only use the basic linear-chain CRF for the base model and no AT, as well as FL, are adopted.

[1]https://nlp.stanford.edu/projects/glove/
[2]https://github.com/tmikolov/word2vec

C. Comparison With State-of-the-Art Methods

In this section, we compare our approach, i.e., RoSeq model, with the SOTA methods on the five benchmark data sets. In the experiment, we run both the base model and our RoSeq model to show the improvements on the benchmark data sets and then compare with SOTA methods. The results on the test set of each benchmark data set are reported in Table III. The proposed RoSeq is able to achieve a new SOTA performance on different data sets, i.e., 88.07% on CoNLL-2002 Dutch, 87.33% on CoNLL-2002 Spanish, 91.42% on CoNLL-2003 English, 52.94% on WNUT-2016 Twitter, and 43.03% on WNUT-2017 Twitter. Note that we do not include the performance of some of the most recent works [9], [20], [30], [31], and [61], for a fair comparison. Although those works achieve slightly higher results on some of the benchmark data sets, they either incorporate external resources, transfer knowledge from other lingual/domain data sets, augment POS tags or lexicons as additional inputs, or jointly training with cross-lingual/domain data sets by sharing model parameters, and so on. In contrast, our model does not use any additional resources and only focus on the task data set itself during the training phase. How to incorporate additional information to further boost performance is not the focus of this paper.

D. Ablation Study of RoSeq Model

In the proposed RoSeq model, we introduce the LACRF, FL, and AT for addressing the label imbalance, easy/hard samples, and noisy data issues. In this section, we investigate the effects of those introduced components and study how they help to improve the model. We conduct the experiments on the CoNLL-2002 Dutch NER and WNUT-2016 English Twitter NER data sets and the results are summarized in Table IV.

From the table, we observe that all the proposed components have contributions to improve the base model, while the AT contributes the most. It is also interesting to note that the performance improvement on WNUT-2016 English Twitter NER data set is higher than that on CoNLL-2002 Dutch NER data set, and we have the similar observation on rest two CoNLL and WNUT-2017 data sets. Since the CoNLL data sets are collected from news wire articles, which are clean and can be easily understood by the learning model. In contrast, the WNUT data sets, which were obtained from user-generated tweets, contain a lot of noises, such as emojis, punctuation, incomplete or misspelled words, and so on. Those noises affect the model's performance largely due to a large amount of

TABLE III
COMPARISON WITH THE SOTA METHODS (F1 SCORE (%))

Method	CoNLL-2002 Dutch	CoNLL-2002 Spanish	CoNLL-2003 English	WNUT-2016 Twitter	WNUT-2017 Twitter
Copara1 *et al.* [62]	-	82.44	-	-	-
Huang *et al.* [1]	-	-	90.10	-	-
Gillick *et al.* [63]	82.84	82.59	86.50	-	-
Luo *et al.* [7]	-	-	91.20	-	-
Lample *et al.* [2]	81.74	85.75	90.94	-	-
Ma *et al.* [10]	-	-	91.21	-	-
Yang *et al.* [21]	85.19	85.77	91.26	-	-
Lin *et al.* [29]	86.55	85.88	-	-	-
Sikdar *et al.* [64]	-	-	-	40.06	-
Espinosa *et al.* [65]	-	-	-	44.77	-
Partalas *et al.* [57]	-	-	-	46.16	-
Limsopatham & Collier [58]	-	-	-	52.41	-
Patrick *et al.* [66]	-	-	-	-	39.98
Lin *et al.* [67]	-	-	-	-	40.42
Von & Cieliebak [68]	-	-	-	-	40.78
Aguilar *et al.* [27]	-	-	-	-	41.86
Base Model	85.23	85.37	90.33	46.13	39.12
RoSeq Model	**88.07**	**87.33**	**91.42**	**52.94**	**43.03**

TABLE IV
PERFORMANCE COMPARISON BETWEEN MODELS WITH DIFFERENT COMPONENTS

Method	NER Datasets	
	CoNLL-2002 Dutch	WNUT-2016 English Twitter
Base Model	85.23	46.13
+LACRF	85.61	46.99
+FL	86.08	47.86
+AT	86.95	50.85
+LACRF +FL	86.52	48.20
+LACRF +AT	87.32	51.11
+FL +AT	87.67	52.09
RoSeq Model	**88.07**	**52.94**

out-of-vocabulary tokens, wrong semantic/syntactic relations between words, and so on.

For the CoNLL-2002 Dutch data set, the base model achieves 85.23% F1 score. The LACRF component slightly improves the performance by 0.38%. The FL component improves the performance with around 0.85%. The AT improves the model most significantly with pushing the F1 score to 86.95%. Therefore, AT not only helps to generalize the model and suppress the noise but also create new training samples by injecting the perturbation into the input token embeddings. Finally, the RoSeq model, which merges those three components, achieves 88.07% on the CoNLL-2002 Dutch NER data set.

For the WNUT-2016 English Twitter data set, the performance increasing trends are similar to that on Dutch data set. Although the absolute F1 scores obtained by the model are relatively lower than that of the CoNLL-2002 Dutch data set, the performance improvement is much larger. Similarly, the AT contributes the most, since WNUT-2016 data set is the noisy user-generated data and AT component is good at suppressing the noises in the data set. Different from CoNLL-2002 Dutch data set, the RoSeq model achieves more than 6.8% improvements on WNUT-2016 data set.

E. Labelwise Performance Analysis

In the proposed RoSeq model, the introduced components, i.e., LACRF, FL, and AT, show improvements for NER task on different data sets. In this experiment, we further conduct the labelwise performance analysis of the RoSeq model and the base model. Here, we also take CoNLL-2002 Dutch NER and WNUT-2016 English Twitter NER as a showcase and results are illustrated in Fig. 5. Note that the result of "O" label in the figure is only visualized for comparison, which is not considered as the contribution in the final result. Labels in the figure are sorted by frequency in descending order.

In Fig. 5(a), the RoSeq model outperforms the base model on all the five labels. Although it only shows a slight improvement on the unmeaning and dominating "O" label compared to the base model, the RoSeq gives more significant improvements for the rest meaningful labels. For example, for "PER," "LOC," and "ORG" labels, our method obtains around 1.7%, 0.9%, and 3.01% absolute improvement in F1 score, respectively. For the label "MISC" with the lowest F1 score in the base model, our method improves the F1 score by 5.19% and it is the largest improvement among all the labels, which indicates that our method is more effective for the sparse labels and hard samples.

For WNUT-2016 English Twitter data set, as shown in Fig. 5(b), the results are similar to that on CoNLL-2002 Dutch NER, while the improvements of RoSeq model are more significant. The performance of the base model and our method is comparable in nonmeaningful "O" label. In the remaining meaningful labels, our method increases 5.46%, 4.51%, 13.0%, 4.0%, and 0.15% absolute F1 scores on "person," "geo-loc," "facility," "company," and "movie" labels, respectively. The base model performs worst on the "other," "product," and "sportsteam" labels, and it even fails to recognize the "musicartist" and "tvshow" labels (both of the two labels are 0.0% F1 scores in the base model). The samples with those labels are treated as the hard samples since their training size are limited. Compared with the base model, our method pushes the result of "other" and "sportsteam" to

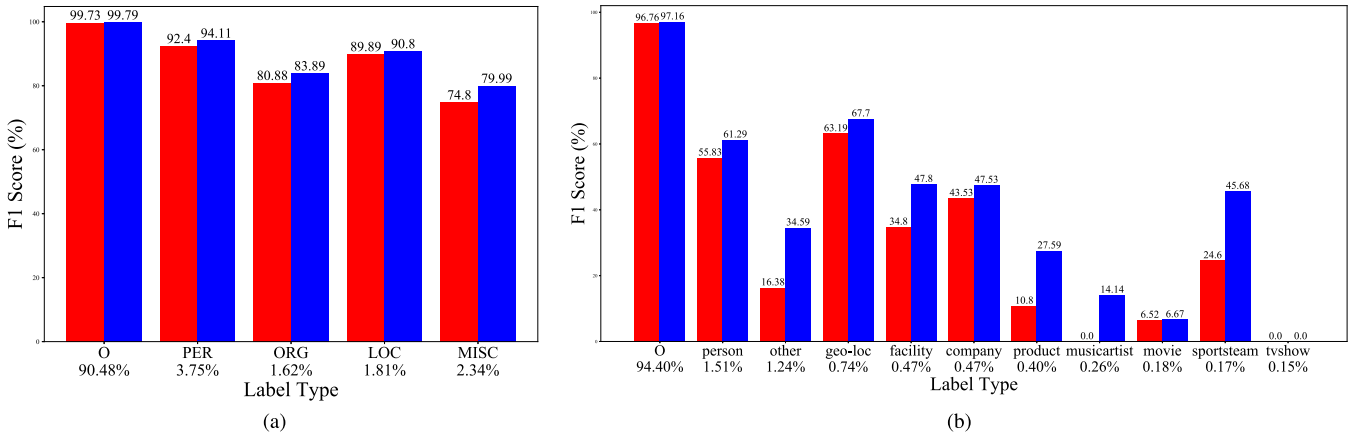

Fig. 5. Comparison of the results between Base and RoSeq models at the label level on CoNLL-2002 Dutch NER and WNUT-2016 English Twitter NER test sets. Red bar: base model. Blue bar: results of RoSeq model. The number below each label denotes the label proportion in the training data set. (a) CoNLL-2002 Dutch. (b) WNUT-2016 English Twitter.

TABLE V
COMPARISON OF PREDICTED EXAMPLES BETWEEN THE BASE MODEL AND ROSEQ

Base Model	RoSeq Model
Dole$_{\text{(B-PER)}}$ also said he opposed California$_{\text{(B-LOC)}}$ Proposition 215 which ...	Dole$_{\text{(B-PER)}}$ also said he opposed California$_{\text{(B-MISC)}}$ Proposition$_{\text{(E-MISC)}}$ 215 which ...
California$_{\text{(B-LOC)}}$ at New$_{\text{(B-LOC)}}$ York$_{\text{(E-LOC)}}$.	California$_{\textbf{(B-ORG)}}$ at New$_{\text{(B-LOC)}}$ York$_{\text{(E-LOC)}}$.
... who appeared in his major-league record 2,282nd straight game today, a 13-0 loss to the California$_{\text{(B-LOC)}}$ Angles$_{\text{(B-ORG)}}$.	... who appeared in his major-league record 2,282nd straight game today, a 13-0 loss to the California$_{\text{(B-ORG)}}$ Angles$_{\text{(E-ORG)}}$.
... (a Newmont-Santa$_{\text{(B-MISC)}}$ Fe$_{\text{(B-ORG)}}$ deal) is positive, it does all the right things.	... (a Newmont-Santa$_{\text{(B-MISC)}}$ Fe$_{\text{(E-MISC)}}$ deal) is positive, it does all the right things.
Newmont$_{\text{(B-ORG)}}$ proposed to Santa$_{\text{(B-ORG)}}$ Fe$_{\text{E-ORG}}$ a stock-swap merger at a ratio of 0.33 ...	Newmont$_{\text{(B-ORG)}}$ proposed to Santa$_{\text{(B-LOC)}}$ Fe$_{\text{E-LOC}}$ a stock-swap merger at a ratio of 0.33 ...

* Green color indicates correctly predicted while red color represents falsely predicted.

34.59% and 45.68%, whose F1 scores absolutely increase by 18.21% and 21.08%, respectively. For the "product" label, our method increases the F1 score from 10.8% of the base model to 27.59%, which is more than 2.5 times improvement. For those failed labels in the base model, our method achieves 14.14% F1 score on "musicartist" label. Furthermore, our method also fails to recognize any "tvshow" label from the data set.

Overall, by comparing the labelwise performance of the base model and our RoSeq model on CoNLL-2002 Dutch and WNUT-2016 English Twitter data sets, we demonstrate that the RoSeq model is able to address the label imbalance issue and also shows its effectiveness to improve the performance, especially on the noisy user-generated data sets (e.g., WNUT data sets) and limited training size of some classes.

Moreover, we also show some predicted examples for both of the base model and our proposed RoSeq algorithm to verify the effectiveness of RoSeq, as justified in Table V. In particular, we choose several testing samples from CoNLL-2003 data sets where the base model fails to assign correct labels for the specific phrases, whereas RoSeq still works well. Those samples also verify the capacity of RoSeq for handling the ambiguous labels, compared with the base method. For example, the ground truth label for "California" in the first sample is "MISCELLANEOUS," but it is "ORGANIZATION" in the second and third samples. Although RoSeq yields the correct predictions, the base model predicts the "California" in all the three samples as "LOCATION" label type (which is the most frequent label for this word. In the fourth sample, "Newmont-Santa" and "Fe" are combined to be labeled as "MISCELLANEOUS." However, the base model assigns two different label types for "Newmont-Santa" and "Fe" as "MISCELLANEOUS" and "ORGANIZATION," respectively. In the last testing sample, "Santa" and "Fe" are merged to represent a "LOCATION." Nevertheless, the base model mistakenly predicts "Santa Fe" as "ORGANIZATION." In contrast, RoSeq makes the correct predictions for the last two testing samples. We have similar observations in other testing samples. Generally, we can conclude that RoSeq is more effective than the base method to reduce the wrong predictions.

V. CONCLUSION

To address the, namely, label imbalance and noisy data problems, we develop a RoSeq model for NER. To the end, we introduce three components, LACRF, FL, and AT, to handle the proposed issues. Extensive experiments show the superiority of RoSeq over existing models on CoNLL-NER and WNUT-NER benchmark data sets without external resources.

REFERENCES

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

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

[3] J. T. Zhou *et al.*, "Learning with annotation of various degrees," *IEEE Trans. Neural Netw. Learning Syst.*, to be published.

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

[5] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proc. 13th Conf. Comput. Natural Lang. Learn.*, May 2009, pp. 147–155.

[6] A. Passos, V. Kumar, and A. McCallum. (2014). "Lexicon infused phrase embeddings for named entity resolution." [Online]. Available: https://arxiv.org/abs/1404.5367

[7] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie, "Joint entity recognition and disambiguation," in *Proc. EMNLP*, Aug. 2015, pp. 879–888.

[8] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," in *Proc. JMLR*, Jun. 2011, pp. 2493–2537.

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

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

[11] 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.

[12] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2017, pp. 2980–2988.

[13] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís, "Named entity recognition: Fallacies, challenges and opportunities," *Comput. Standards Inter.*, no. 5, pp. 482–489, 2013.

[14] 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.

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

[16] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Eighteenth Int. Conf. Mach. Learn.*, 2001, pp. 125–136.

[17] A. Zukov Gregoric, Y. Bachrach, and S. Coope, "Named entity recognition with parallel recurrent neural networks," in *Proc. ACL*, Jul. 2018, pp. 69–74.

[18] 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, Aug. 2018.

[19] 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.

[20] L. Liu *et al.*, "Empower sequence labeling with task-aware neural language model," in *Proc. AAAI*, 2018, pp. 1–10.

[21] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Transfer learning for sequence tagging with hierarchical recurrent networks," in *Proc. ICLR*, 2017, pp. 24–63.

[22] J. Ni and R. Florian, "Improving multilingual named entity recognition with wikipedia entity type mapping," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, May 2016, pp. 1275–1284.

[23] J. Ni, G. Dinu, and R. Florian, "Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection," in *Proc. ACL*, Jun. 2017, pp. 1470–1480.

[24] Z. Yang, R. Salakhutdinov, and W. W. Cohen. (2016). "Multi-task cross-lingual sequence tagging from scratch." [Online]. Available: https://arxiv.org/abs/1603.06270?context=cs

[25] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," in *Proc. ICLR*, Aug. 2016, pp. 45–85.

[26] M. Rei, "Semi-supervised multitask learning for sequence labeling," in *Proc. ACL*, 2017, pp. 2121–2130.

[27] G. Aguilar, S. Maharjan, A. P. López Monroy, and T. Solorio, "A multi-task approach for named entity recognition in social media data," in *Proc. 3rd Workshop Noisy User-Generated Text*, Sep. 2017, pp. 148–153.

[28] K. Hashimoto, C. xiong, Y. Tsuruoka, and R. Socher, "A joint many-task model: Growing a neural network for multiple NLP tasks," in *Proc. EMNLP*, 2017, pp. 1923–1933.

[29] Y. Lin, S. Yang, V. Stoyanov, and H. Ji, "A multi-lingual multi-task architecture for low-resource sequence labeling," in *Proc. ACL*, Jul. 2018, pp. 799–809.

[30] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proc. NAACL*, Jun. 2018, pp. 2227–2237.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." [Online]. Available: https://arxiv.org/abs/1810.04805

[32] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests," in *Proc. BMVC*, 2008, pp. 1–10.

[33] Q. Wang, M. Chen, F. Nie, and X. Li, "Detecting coherent groups in crowd scenes by multiview clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[34] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 20, no. 99, pp. 1–7, Aug. 2018.

[35] Z. Huang, H. Zhu, J. T. Zhou, and X. Peng, "Multiple marginal fisher analysis," *IEEE Trans Industr Electron*, to be published.

[36] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 9, pp. 1263–1284, Dec. 2008.

[37] A. Estabrooks, T. H. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Comput. Intell.*, vol. 20, no. 1, pp. 18–36, 2004.

[38] C. Elkan, "The foundations of cost-sensitive learning," *Int. Joint Conf. Artif.*, vol. 17, no. 1, pp. 973–978, 2001.

[39] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," in *Data Classification: Algorithms and Applications*, vol. 37. Boca Raton, FL, USA: CRC Press, 2014.

[40] N. Cesa-Bianchi, S. Shalev-Shwartz, and O. Shamir, "Online learning of noisy data," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7907–7931, Aug. 2011.

[41] C. Szegedy, W. Zaremba, D. E. I. G. Ilya Sutskever, J. Bruna, and R. Fergus, "Intriguing properties of neural networks," in *Proc. ICLR*, 2014, pp. 1–9.

[42] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. ICLR*, May 2015, pp. 58–69.

[43] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *Proc. ICLR*, Aug. 2017, pp. 241–256.

[44] M. Yasunaga, J. Kasai, and D. Radev, "Robust multilingual part-of-speech tagging via adversarial training," in *Proc. NAACL HLT*, 2018, pp. 976–986.

[45] N. Reimers and I. Gurevych, "Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging," in *Proc. EMNLP*, Sep. 2017, pp. 338–348.

[46] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proc. 13th AAAI Conf. Artif. Intell.*, May 2016, pp. 2741–2749.

[47] S. R. Kumar, G. Klaus, and S. Jürgen. (2015). "Highway networks." [Online]. Available: https://arxiv.org/abs/1505.00387

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 15, no. 8, pp. 1735–1780, 1997.

[49] T. E. F. and S. V. Kim Jorn, "Representing text chunks," in *Proc. 9th Conf. Eur. Chapter Assoc. Comput. Linguistics*, Jun. 1999, pp. 173–179.

[50] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li. (2017). "Adversarial examples: Attacks and defenses for deep learning." [Online]. Available: https://arxiv.org/abs/1712.07107

[51] C. Pin-Yu, S. Yash, Z. Huan, Y. Jinfeng, and C.-J. Hsieh, "Ead: Elastic-net attacks to deep neural networks via adversarial examples," in *Proc. AAAI*, 2018, pp. 20–30.

[52] S. E. F. T. Kim, "Introduction to the conll-2002 shared task: Language-independent named entity recognition," in *Proc. 6th Conf. Natural Lang. Learn.*, 2002, pp. 58–63.

[53] S. E. F. T. Kim and M. F. De, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proc. 7th Conf. Natural Lang. Learn.*, 2003, pp. 142–147.

[54] D. E. A. Zeman, "CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies," in *Proc. ACL Anthology*, 2017, pp. 1–19.

[55] D. Hong-Jie, L. Po-Ting, C. Yung-Chun, and T. R. Tzong-Han, "Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization," *J. Cheminform.*, vol. 7, no. 1, p. S14, 2015.
[56] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.
[57] I. Partalas, C. Lopez, N. Derbas, and R. Kalitvianski, "Learning to search for recognizing named entities in twitter," in *Proc. 2nd Workshop Noisy User-Generated Text (WNUT)*, Dec. 2016, pp. 171–177.
[58] N. Limsopatham and N. Collier, "Bidirectional LSTM for named entity recognition in twitter messages," in *Proc. 2nd Workshop Noisy User-Generated Text (WNUT)*, Mar. 2016, pp. 145–152.
[59] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980
[60] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Proc. JMLR*, May 2014, pp. 1929–1958.
[61] X. Feng, X. Feng, B. Qin, Z. Feng, and T. Liu, "Improving low resource named entity recognition using cross-lingual knowledge transfer," in *Proc. IJCAI*, Jul. 2018, pp. 4071–4077.
[62] J. L. C. Zea, J. E. O. Luna, C. Thorne, and G. Glavaš, "Spanish ner with word representations and conditional random fields," in *Proc.6th Named Entity Workshop*, 2016, pp. 34–40.
[63] D. Gillick, C. Brunk, O. Vinyals, and A. Subramanya, "Multilingual language processing from bytes," in *NAACL HLT*, 2016, pp. 1296–1306.
[64] S. U. Kumar and G. Björn, "Feature-rich twitter named entity recognition and classification," in *Proc. 2nd Workshop Noisy User-Generated Text (WNUT)*, 2016, pp. 164–170.
[65] E. K. Junshean, B.-N. R. Theresa, and A. Sophia, "Learning to recognise named entities in tweets by exploiting weakly labelled data," in *Proc. 2nd Workshop Noisy User-Generated Text (WNUT)*, 2016, pp. 153–163.
[66] J. Patrick and L. Shuhua, "Distributed representation, LDA topic modelling and deep learning for emerging named entity recognition from social media," in *Proc. 3rd Workshop Noisy User-Generated Text*, 2017, pp. 154–159.
[67] B. Y. Lin, F. Xu, Z. Luo, and K. Zhu, "Multi-channel BiLSTM-CRF model for emerging named entity recognition in social media," in *Proc. 3rd Workshop Noisy User-Generated Text*, 2017, pp. 160–165.
[68] P. von Dániken and M. Cieliebak, "Transfer learning and sentence level features for named entity recognition on tweets," in *Proc. 3rd Workshop Noisy User-Generated Text*, Jun. 2017, pp. 166–171.

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 Research Agency for Science, Technology, and Research, Institute of High Performance Computing, Singapore.

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. He has served as an Associate Editor for the IEEE ACCESS and a Guest Editor for *IET Image 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 the Agency of Science, Technology, and Research, Artificial Intelligence Initiative, Singapore.

Di Jin received the B.S. degree in precision instruments from Tsinghua University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA.

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

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

Yang Xiao received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2004, 2007, and 2011, respectively.

He was a Research Fellow with the School of Computer Engineering, Institute of Media Innovation, Nanyang Technological University, Singapore. He is currently an Associate Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His current research interests include computer vision, image processing, and machine learning.

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, in 1985 and 1990, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the Huazhong University of Science and Technology, Wuhan, China, in 2001.

He is currently a Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His current research interests include spread across image understanding and analysis, depth information extraction, 3-D video processing, motion detection and human action analysis. His current research results, which have published dozens of papers at International Journals and Prominent Conferences, have been applied to automatic observation system for crop growth in agricultural, for weather phenomenon in meteorology and for object recognition in video surveillance system based on computer vision.