NEURAL CRF TRANSDUCERS FOR SEQUENCE LABELING

Kai Hu[†], Zhijian Ou[†], Min Hu[‡], Junlan Feng[‡]

[†]Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, China
[‡] China Mobile Research Institute
huk17@mails.tsinghua.edu.cn, ozj@tsinghua.edu.cn, {humin,fengjunlan}@chinamobile.com

ABSTRACT

Conditional random fields (CRFs) have been shown to be one of the most successful approaches to sequence labeling. Various linear-chain neural CRFs (NCRFs) are developed to implement the non-linear node potentials in CRFs, but still keeping the linear-chain hidden structure. In this paper, we propose NCRF transducers, which consists of two RNNs, one extracting features from observations and the other capturing (theoretically infinite) long-range dependencies between labels. Different sequence labeling methods are evaluated over POS tagging, chunking and NER (English, Dutch). Experiment results show that NCRF transducers achieve consistent improvements over linear-chain NCRFs and RNN transducers across all the four tasks, and can improve state-of-the-art results.

Index Terms— conditional random field, recurrent networks, transducer, sequence labeling

1. INTRODUCTION

Given a sequence of observations $x \triangleq x_1, \cdots x_n$, the task of sequence labeling is to predict a sequence of labels $y \triangleq y_1, \cdots y_n$, with one label for one observation in each position. Sequence labeling is of broad interest and has been popularly applied in the area of natural language processing (NLP), like in part-of-speech (POS) tagging [1, 2], named entity recognition (NER) [3, 4, 5], chunking [3, 6], syntactic parsing [7] and semantic slot filling [8], and also in other areas such as bioinformatics [9, 10].

Conditional random fields (CRFs) [11] have been shown to be one of the most successful approaches to sequence labeling. A recent progress is to develop Neural CRF (NCRF) models, which combines the sequence-level discriminative ability of CRFs and the representation ability of neural networks (NNs), particularly the recurrent NNs (RNNs). These models have achieved state-of-the-art results on a variety of sequence labeling tasks, and in different studies, are called conditional neural field [10], neural CRF [12], recurrent CRF [13], and LSTM-CRF [4, 5]. Though there are detailed differences between these existing models, generally they are all defined by using NNs (of different network architectures) to implement the non-linear node potentials in CRFs, while still keeping the linear-chain hidden structure (i.e. using a bigram table as the edge potential). For convenience, we refer to these existing combinations of CRFs and NNs as *linear-chain NCRFs* in general. This represents an extension from conventional CRFs, where both node potentials and edge potentials are implemented as linear functions using discrete indicator features.

In this paper, we present a further extension and propose *neural CRF transducers*, which introduce a LSTM-RNN to implement

a new edge potential so that long-range dependencies in the label sequence are captured and modeled. In contrast, linear-chain NCRFs capture only first-order interactions and neglect higher-order dependencies between labels, which can be potentially useful in real-world sequence labeling applications, e.g. as shown in [14] for chunking and NER.

There are two LSTM-RNNs in a NCRF transducer, one extracting features from observations and the other capturing (theoretically infinite) long-range dependencies between labels. In this view, a N-CRF transducer is similar to a RNN transducer [15], which also uses two LSTM-RNNs. Additionally, the recent attention-based seq2seq models [16] also use an LSTM-based decoder to exploit long-range dependences between labels. However, both RNN transducers and seq2seq models, as locally normalized models, produce position-by-position conditional distributions over output labels, and thus suffer from the label bias and exposure bias problems [11, 17, 18]. In contrast, NCRF transducers are globally normalized, which overcome these two problems. We leave more discussions about existing related studies to section 6.

Different sequence labeling methods are evaluated over POS tagging, chunking and NER (English, Dutch). Experiment results show that NCRF transducers achieve consistent improvements over linear-chain NCRFs and RNN transducers across all the four tasks. Notably, in the CoNLL-2003 English NER task, the NCRF transducer achieves state-of-the-art F_1 (92.36), better than 92.22 [19], using the same deep contextualized word representations.

2. BACKGROUND

Linear-chain NCRFs. A linear-chain CRF defines a conditional distribution for label sequence y given observation sequence x :

$$p(y|x) \propto \exp \left\{ \sum_{i=1}^{n} \phi_i(y_i, x) + \sum_{i=1}^{n} \psi_i(y_{i-1}, y_i, x) \right\}.$$

Here the labels y_i's are structured to form a chain, giving the term linear-chain. $\phi_i(y_i, x)$ is the node potential defined at position i, which, in recently developed NCRFs [1, 3, 4, 5, 10] is implemented by using features generated from a NN of different network architectures. $\psi_i(y_{i-1}, y_i, x)$ is the edge potential defined on the edge connecting y_{i-1} and y_i, which, in these existing NCRFs, is mostly implemented as a transition matrix A:

$$\psi_i(y_{i-1} = j, y_i = k, x) = A_{j,k}$$

This edge potentials admit efficient algorithms for training and decoding, but only capture first-order dependencies between labels.

RNN Transducers are originally developed for general sequence-to-sequence learning [15], which do not assume that the input

Table 1. Model comparison and connection.		
Model	**Globally normalized**	**Long-range dependencies between labels**
Linear-chain NCRF	$\checkmark$	$\times$
RNN Transducer	$\times$	$\checkmark$
NCRF Transducer	$\checkmark$	$\checkmark$

and output sequences are of equal lengths and aligned, e.g. in speech recognition. However, it can be easily seen that RNN transducers can be applied to sequence labeling as well, by defining $p(y|x) = \prod_{i=1}^{n} p(y_i|y_{0:i-1}, x)$ and implementing $p(y_i|y_{0:i-1}, x)$ through two networks - transcription network F and prediction network G as follows:

$$p(y_i = k|y_{0:i-1}, x) = \frac{\exp(f_i^k + g_i^k)}{\sum_{k'=1}^{K} \exp(f_i^{k'} + g_i^{k'})} \quad (1)$$

Here F scans the observation sequence x and outputs the transcription vector sequence $f \triangleq f_1, \cdots f_n$. G scans the label sequence $y_{0:n-1}$ and outputs the prediction vector sequence $g \triangleq g_1, \cdots g_n$. y_0 denotes the beginning symbol ($< bos >$) of the label sequence. For a sequence labeling task with K possible labels, f_i and g_i are K dimensional vectors. Superscript k is used to denote the kth element of the vectors. Remarkably, the prediction network G can be viewed as a label language model, capable of modeling long-range dependencies in y, which is exactly the motivation to introducing G in RNN transducers.

To ease comparison, we will also refer to the network below the CRF layer in linear-chain NCRFs as a transcription network, since we also implement $\phi_i(y_i = k, x; \theta)$ as f_i^k in our experiments.

3. NCRF TRANSDUCERS

In the following, we develop NCRF transducers, which combine the advantages of linear-chain NCRFs (globally normalized, using LSTM-RNNs to implement node potentials) and of RNN transducers (capable of capturing long-range dependencies in labels), and meanwhile overcome their drawbacks, as illustrated in Table 1.

3.1. Model definition

A NCRF transducer defines a globally normalized, conditional distribution $p(y|x; \theta)$ as follows:

$$p(y|x; \theta) = \frac{\exp\{u(y, x; \theta)\}}{Z(x; \theta)}.$$

where $Z(x; \theta) = \sum_{y' \in \mathcal{D}_n} \exp\{u(y', x; \theta)\}$ is the global normalizing term and $\mathcal{D}_n$ is the set of allowed label sequences of length n. The total potential $u(y, x; \theta)$ is decomposed as follows:

$$u(y, x; \theta) = \sum_{i=1}^{n} \{\phi_i(y_i, x; \theta) + \psi_i(y_{0:i-1}, y_i; \theta)\}.$$

where $\phi_i(y_i, x; \theta)$ is the node potential at position i, $\psi_i(y_{0:i-1}, y_i; \theta)$ is the clique potential involving labels from the beginning up to position i. Thus the underlying undirected graph for the label sequence y is fully-connected, which potentially can capture long-range dependencies from the beginning up to each current position.
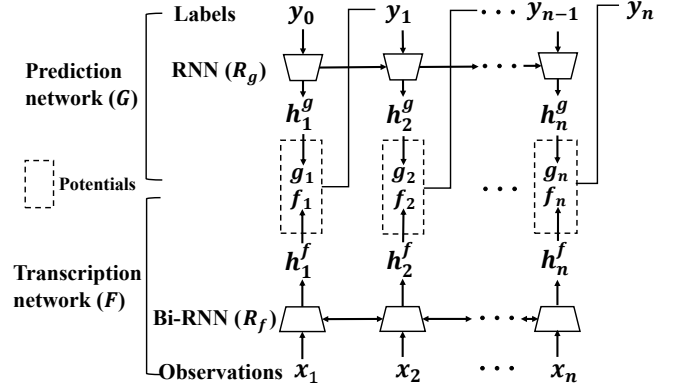

Fig. 1. The architecture of a NCRF transducer.

3.2. Neural network architectures

Like in RNN transducers, we introduce two networks in NCRF transducers, as shown in Fig. 1. The transcription network F implements the node potential $\phi_i(y_i, x; \theta)$, which represents the score for y_i based on observations x. In our experiments on NLP sequence labeling, each word x_i is represented by a concatenation of a pretrained word embedding vector and another embedding vector obtained from a character-level CNN. The transcription network F is a bidirectional RNN (R_f) that scans the sequence of the concatenated vectors for words to generate hidden vectors $h_i^f = [\overrightarrow{h_i^f}; \overleftarrow{h_i^f}]$, which are then fed to a linear layer with output size of K to generate $f_i \in \mathbb{R}^K$.

The prediction network G implements the clique potential $\psi_i(y_{0:i-1}, y_i; \theta)$, which represents the score for y_i by taking account of dependencies between y_i and previous labels $y_{0:i-1}$. In our experiments, each label y_i is represented by a label embedding vector, initialized randomly. G is a unidirectional RNN (R_g) that accepts the label sequence y and generates hidden vectors $h_i^g = \overrightarrow{h_i^g}$, which are then fed to a linear layer with output size of K to generate $g_i \in \mathbb{R}^K$.

It can be seen from above that a NCRF transducer is similar to a RNN transducer. The difference is that a RNN transducer is local normalized through softmax calculations as shown in Eq. (1), while a NCRF transducer is globally normalized, locally producing (unnormalized) potential scores.

3.3. Potential design

Based on f_i and g_i, there are two possible designs to implement the potentials ϕ_i and ψ_i, which are chosen empirically in our experiments. The first design is:

$$\phi_i(y_i = k, x; \theta) = f_i^k$$
$$\psi_i(y_{0:i-1}, y_i = k; \theta) = g_i^k \quad (2)$$

The second design is:

$$\phi_i(y_i = k, x; \theta) = log \frac{\exp(f_i^k)}{\sum_{k'=1}^{K} \exp(f_i^{k'})}$$
$$\psi_i(y_{0:i-1}, y_i = k; \theta) = log \frac{\exp(g_i^k)}{\sum_{k'=1}^{K} \exp(g_i^{k'})} \quad (3)$$

3.4. Decoding and training

NCRF transducers break the first-order Markov assumption in the label sequence as in linear-chain NCRFs and thus do not admit dynamic programming for decoding. Instead, we use beam search to approximately find the most probable label sequence:

$$\hat{y} = \underset{y' \in \mathcal{D}_n}{\operatorname{argmax}} \, p(y'|x;\theta) = \underset{y' \in \mathcal{D}_n}{\operatorname{argmax}} \, u(y', x; \theta).$$

Training data consists of inputs x paired with oracle label sequences y^*. We use stochastic gradient descent (SGD) on the negative log-likelihood of the training data:

$$L(y^*; \theta) = -u(y^*, x; \theta) + \log Z(x; \theta).$$

It is easy to calculate the gradient of the first term. However, the gradient of the log normalizing term involves model expectation:

$$\nabla_\theta \log Z(x; \theta) = E_{p(y'|x;\theta)} \left[\nabla_\theta u(y', x; \theta) \right]$$

The calculations of the normalizing term and the model expectation can be exactly performed for linear-chain NCRFs (via the forward and backward algorithm), but are intractable for NCRF transducers. It is empirically found in our experiments that the method of beam search with early updates [20] marginally outperforms Monte Carlo based methods for training NCRF transducers.

The basic idea is that we run beam search and approximate the normalizing term by summing over the paths in the beam. Early updates refer to that as the training sequence is being decoded, we keep track of the location of the oracle path in the beam; If the oracle path falls out of the beam at step j, a stochastic gradient step is taken on the following objective:

$$L(y^*_{1:j}; \theta) = -u(y^*_{1:j}; \theta) + log \sum_{y' \in \mathcal{B}_j} \exp \left\{ u(y'_{1:j}; \theta) \right\}$$

where $u(y_{1:j}; \theta) = \sum_{i=1}^{j} \left\{ \phi_i(y_i, x; \theta) + \psi_i(y_{0:i-1}, y_i; \theta) \right\}$ denotes the partial potential (with abuse of the notation of u). The set $\mathcal{B}_j$ contains all paths in the beam at step j, together with the oracle path prefix $y^*_{1:j}$.

4. EXPERIMENTAL SETUP

Different sequence labeling methods are evaluated over four tasks - POS tagging, chunking and NER (English, Dutch). We replace all digits with zeros and rare words (frequency less than 1) by <UNK>, as a common pre-processing step for all methods.

Datasets. The following benchmark datasets are used - PTB POS tagging, CoNLL-2000 chunking, CoNLL-2003 English NER and CoNLL-2002 Dutch NER. For the task of POS tagging, we follow the previous work [5] to split the dataset and report accuracy. For the NER tasks, we follow the previous work [5] to use the BIOES tagging scheme and report micro-average F_1. For the chunking task, we follow the previous work [21]; 1000 sentences are randomly sampled from the training set to be the development set. BIOES tagging scheme and F_1 are used.

Model configuration. For word embeddings, 100-dim Glove embeddings [22] are used for the tasks of POS tagging and English NER. 64-dim skip-n-gram embeddings [4] are used for the Dutch NER task. 50-dim Senna embeddings [1] are used for the chunking task. For character embeddings, we use 30-dim embeddings (randomly initialized) and a CNN consisting of 30 filters with 3-character width. For label embeddings, 10-dim embeddings are used (randomly initialized).

Table 2. POS tagging results over PTB dataset.

Model	Accuracy
Collobert et al [1]	97.29
Manning [23]	97.28
Santos & Zadrozny [24]	97.32
Sun [25]	97.36
Ma & Hovy [5]	97.55
Linear-chain NCRF	97.52
RNN transducer	97.50
NCRF transducer	**97.52**

Table 3. Chunking results, trained only with CoNLL-2000 data.

Model	$F_1 \pm$ **std**	F_1 **max**
Sϕgaard & Goldberg [6]	95.28	
Hashimoto et al [26]	95.02	
Yang et al [27]	94.66	
Peters et al [21]	95.00 ± 0.08	
Linear-chain NCRF	95.01 ± 0.12	95.15
RNN transducer	95.02 ± 0.11	95.13
NCRF transducer	$\mathbf{95.14 \pm 0.05}$	**95.23**

For the transcription network F, a bidirectional LSTM, consisting of one layer of 200 hidden units, is used in the POS tagging and NER tasks; the bidirectional LSTM used in the chunking task consists of two layers with 200 hidden units each layer. For the prediction network G, an unidirectional LSTM, consisting of one layer of 50 hidden units, is used in all tasks.

For optimizers, SGD with momentum 0.9 is used in the tasks of POS tagging and English NER, the initial learning rate is 0.01 with a 0.05 decay rate per epoch as in the previous work [5]. In the tasks of chunking and Dutch NER, Adam is used, the initial learning rate being fixed at 1e-3 without decay. Beam width 128 is used for training and 512 is used for decoding. The mini-batch size is set to be 16. Following the previous work [5], we add 0.5 dropout.

In our experiments, NCRF transducers are initialized with the weights from pre-trained RNN transducers, which is found to yield better and faster learning. When finetuning the pre-trained model, we set the initial learning rate to be 5e-3 and adopt SGD with momentum 0.9. In each task, we tune hyperparameters on the development set and use early stopping.

5. EXPERIMENTAL RESULTS

For each method, we perform five independent runs with random initializations and report the mean and standard deviation. For comparison with other top-performance systems, we report the best results from the two potential designs - the design in Eq. (2) for POS tagging and chunking and the design in Eq. (3) for NER.

5.1. Comparison of NCRF transducers with linear-chain N-CRFs and RNN Transducers

For fair comparison of different methods, the same transcription network architecture is used in linear-chain NCRFs, RNN transducers and NCRF transducers in our experiments.

First, it can been seen from Table 2, 3, 4, 5 and 6 that NCRF transducers outperform linear-chain NCRFs for three out of the four sequence labeling tasks, which clearly shows the benefit of long-range dependency modeling. Numerically, the improvements

Table 4. English NER results, trained only with CoNLL-2003 data.

Model	$F_1 \pm$ **std**	F_1 **max**
Luo et al [28]	89.90	
Chiu & Nichols [29]	90.91 ± 0.20	
Lample et al [4]	90.94	
Ma & Hovy [5]	91.21	
Yang et al [27]	91.20	
Peters et al [21]	90.87 ± 0.13	
Liu et al [30]	91.24 ± 0.12	91.35
Linear-chain NCRF	91.11 ± 0.16	91.30
RNN transducer	91.02 ± 0.15	91.23
NCRF transducer	$\mathbf{91.40 \pm 0.11}$	**91.66**

Table 5. English NER results, trained only with CoNLL-2003 data but using other external resources.

Model	External resources	$F_1 \pm$ **std**	F_1 **max**
Collobert et al [1]	gazetteers	89.59	
Luo et al [28]	entity linking	91.2	
Chiu & Nichols [29]	gazetteers	91.62	
Yang et al [27]	PTB-POS	91.26	
Peters et al [19]	ELMo	92.22 ± 0.10	
Linear-chain NCRF	ELMo	92.23 ± 0.13	92.51
RNN transducer	ELMo	92.03 ± 0.21	92.30
NCRF transducer	ELMo	$\mathbf{92.36 \pm 0.20}$	**92.61**

are: 0.13 mean F_1 in chunking, 0.29/0.13 mean F_1 in English NER (without/with ELMo), and 0.31 mean F_1 in Dutch NER. Such equal performances in POS tagging suggests that long-range dependencies are not significant in POS tagging.

Second, NCRF transducers outperform RNN transducers across the four sequence labeling tasks. Numerically, the improvements are: 0.02 accuracy in POS tagging, 0.12 mean F_1 in chunking, 0.38/0.33 mean F_1 in English NER (without/with ELMo), 0.25 mean F_1 in Dutch NER. This demonstrates the advantage of overcoming the label bias and exposure bias problems in NCRF transducers. Note that to solve the exposure bias problem, scheduled sampling has been applied to training RNNs [33]. We also train RNN transducers with scheduled sampling, but no better results are obtained.

5.2. Comparison with prior results

In the following, we compare the results obtained from NCRF transducers with prior results to show the significance of NCRF transducers, but noting that there are differences in network architectures and other experimental conditions.

POS tagging. As shown in Table 2, NCRF transducer achieves highly competitive accuracy of 97.52, close to 97.55 obtained by Bi-LSTM-CNNs-CRF in [5].

Chunking. As shown in Table 3, NCRF transducer achieves 95.14 mean F1, representing an improvement of 0.14 mean F_1 over an advanced linear-chain NCRF [21]. The higher result of 95.28 [6] is obtained with deep bi-RNNs models.

English NER. Table 4 and 5 show the results without/with external resources respectively. In both cases, NCRF transducers produce new state-of-the-art results. When no external resources are used, NCRF transducer gives 91.40 mean F_1, outperforming the previous best result 91.24 (linear-chain NCRF) [30]. When augmented with ELMo [19], NCRF transducer further achieves 92.36 mean F_1, bet-

Table 6. Dutch NER results, trained only with CoNLL-2002 data.

Model	$F_1 \pm$ **std**	F_1 **max**
Nothman et al [31]	78.6	
Gillick et al [32]	78.08	
Lample et al [4]	81.74	
Linear-chain CRF	81.53 ± 0.31	81.76
RNN transducer	81.59 ± 0.09	81.70
NCRF transducer	$\mathbf{81.84 \pm 0.07}$	**81.94**

ter than the original ELMo result.

Dutch NER. As shown in Table 6, NCRF transducer achieves 81.84 mean F_1, which is better than 81.74 from [4] (linear-chain NCRF), when using the same word embeddings in [4]. The results of 85.19 in [27] and 82.84 in [32] are obtained using different word embeddings with the help of transfer learning [27] or multilingual data [32], so these two results cannot be directly compared to the results of NCRF transducers.

6. RELATED WORK

Extending CRFs to model higher-order interactions than pairwise relationships between labels is an important issue for sequence labeling. There are some prior studies, e.g. higher-order CRFs [34], semi-Markov CRFs [35] and latent-dynamic CRFs [36], but not using NNs. Using NNs to enhance the modeling of long-range dependencies in CRFs is under-appreciated in the literature. A related work is structured prediction energy networks (SPENs) [37], which use neural networks to define energy functions that potentially can capture long-range dependencies between structured outputs/labels. SPENs depend on relaxing labels from discrete to continuous and use gradient descent for test-time inference, which is time-consuming. Training and inference with SPENs are still challenging, though with progress [38].

Our work on developing (globally normalized) NCRF transducers is related to prior studies on developing globally normalized models [17, 18], which aim to overcome the label bias and exposure bias problems that locally normalized models suffer from. The beam search training scheme in our work is close to [17, 18], but the developed models are different. The globally normalized model in [17] defines a CRF which uses feedforward neural networks to implement potentials but does not address long-range dependencies modeling in CRFs. The work in [18] extends the seq2seq model by removing the final softmax in the RNN decoder to learn global sequence scores.

7. CONCLUSION

In this paper, we propose neural CRF transducers, which consists of two RNNs, one extracting features from observations and the other capturing (theoretically infinite) long-range dependencies between labels. Experiment results show that NCRF transducers achieve consistent improvements over linear-chain NCRFs and RNN transducers across four sequence labeling tasks, and can improve state-of-the-art results. An interesting future work is to apply NCRF transducers to those structured prediction tasks, e.g. speech recognition, where long-range dependencies between labels are more significant.

8. REFERENCES

[1] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, "Natural Language Processing (Almost) from Scratch," *JMLR*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[2] Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis, "Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation," in *EMNLP*, 2015.

[3] Zhiheng Huang, Wei Xu, and Kai Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[4] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer, "Neural Architectures for Named Entity Recognition," in *NAACL-HLT*, 2016.

[5] Xuezhe Ma and Eduard Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," in *ACL*, 2016.

[6] Anders Søgaard and Yoav Goldberg, "Deep multi-task learning with low level tasks supervised at lower layers," in *ACL*, 2016.

[7] Greg Durrett and Dan Klein, "Neural CRF Parsing," in *ACL*, 2015.

[8] Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou, "Neural Models for Sequence Chunking.," in *AAAI*, 2017.

[9] Kengo Sato and Yasubumi Sakakibara, "RNA secondary structural alignment with conditional random fields," *Bioinformatics*, vol. 21, pp. 237–42, 2005.

[10] Jian Peng, Liefeng Bo, and Jinbo Xu, "Conditional Neural Fields," in *NIPS*, 2009.

[11] John Lafferty, Andrew McCallum, and Fernando CN Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *ICML*, 2001.

[12] Thierry Artieres et al., "Neural conditional random fields," in *AISTATS*, 2010.

[13] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Z. Hakkani-Tür, Xiaodong He, Larry P. Heck, Gökhan Tür, Dong Yu, and Geoffrey Zweig, "Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, pp. 530–539, 2015.

[14] Yi Zhang, Xu Sun, Shuming Ma, Yang Yang, and Xuancheng Ren, "Does Higher Order LSTM Have Better Accuracy for Segmenting and Labeling Sequence Data?," in *COLING*, 2018.

[15] Alex Graves, "Sequence Transduction with Recurrent Neural Networks," in *Representation Learning Worksop,ICML*, 2012.

[16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *ICLR*, 2015.

[17] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins, "Globally Normalized Transition-Based Neural Networks," in *ACL*, 2016.

[18] Sam Wiseman and Alexander M Rush, "Sequence-to-sequence Learning as Beam-Search Optimization," in *EMNLP*, 2016.

[19] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, "Deep Contextualized Word Representations," in *NAACL-HLT*, 2018.

[20] Michael Collins and Brian Roark, "Incremental Parsing with the Perceptron Algorithm," in *ACL*, 2004.

[21] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power, "Semi-supervised sequence tagging with bidirectional language models," in *ACL*, 2017.

[22] Jeffrey Pennington, Richard Socher, and Christopher Manning, "Glove: Global Vectors for Word Representation," in *EMNLP*, 2014.

[23] Christopher D Manning, "Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?," in *International conference on intelligent text processing and computational linguistics*. Springer, 2011, pp. 171–189.

[24] Cicero D Santos and Bianca Zadrozny, "Learning Character-level Representations for Part-of-Speech Tagging," in *ICML*, 2014.

[25] Xu Sun, "Structure Regularization for Structured Prediction," in *NIPS*, 2014.

[26] Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al., "A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks," in *EMNLP*, 2017.

[27] Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen, "Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks," in *ICLR*, 2017.

[28] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie, "Joint Entity Recognition and Disambiguation," in *EMNLP*, 2015.

[29] Jason Chiu and Eric Nichols, "Named Entity Recognition with Bidirectional LSTM-CNNs," *TACL*, vol. 4, no. 1, pp. 357–370, 2016.

[30] Liyuan Liu, Jingbo Shang, Frank Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han, "Empower Sequence Labeling with Task-Aware Neural Language Model," in *AAAI*, 2018.

[31] Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran, "Learning multilingual named entity recognition from Wikipedia," *Artificial Intelligence*, vol. 194, pp. 151–175, 2013.

[32] Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya, "Multilingual Language Processing From Bytes," in *NAACL-HLT*, 2016.

[33] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks," in *NIPS*, 2015.

[34] Sotirios P. Chatzis and Yiannis Demiris, "The Infinite-Order Conditional Random Field Model for Sequential Data Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1523–1534, 2013.

[35] Sunita Sarawagi and William W. Cohen, "Semi-Markov Conditional Random Fields for Information Extraction," in *NIPS*, 2004.

[36] Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell, "Latent-Dynamic Discriminative Models for Continuous Gesture Recognition," in *CVPR*, 2007.

[37] David Belanger and Andrew McCallum, "Structured Prediction Energy Networks," in *ICML*, 2016.

[38] Lifu Tu and Kevin Gimpel, "Learning Approximate Inference Networks for Structured Prediction," in *ICLR*, 2018.