Chinese NER with Height-Limited Constituent Parsing

Rui Wang,[1] Xin Xin,[1]* Wei Chang,[1] Kun Ming,[1] Biao Li,[2] Xin Fan[2]
[1]BJ ER Center of HVLIP&CC, School of Comp. Sci. & Tech., Beijing Institute of Technology, Beijing, China
[2]Tencent, Beijing, China
{ruicn,xxin,2220160514,2120171045}@bit.edu.cn, {biotli,hsinfan}@tencent.com

Abstract

In this paper, we investigate how to improve Chinese named entity recognition (NER) by jointly modeling NER and constituent parsing, in the framework of neural conditional random fields (CRF). We reformulate the parsing task to height-limited constituent parsing, by which the computational complexity can be significantly reduced, and the majority of phrase-level grammars are retained. Specifically, an unified model of neural semi-CRF and neural tree-CRF is proposed, which simultaneously conducts word segmentation, part-of-speech (POS) tagging, NER, and parsing. The challenge comes from how to train and infer the joint model, which has not been solved previously. We design a dynamic programming algorithm for both training and inference, whose complexity is $O(n \cdot 4^h)$, where n is the sentence length and h is the height limit. In addition, we derive a pruning algorithm for the joint model, which further prunes 99.9% of the search space with 2% loss of the ground truth data. Experimental results on the OntoNotes 4.0 dataset have demonstrated that the proposed model outperforms the state-of-the-art method by 2.79 points in the F1-measure.

Introduction

Named entity recognition (NER) is to identify the boundaries of a named entity in a natural language sentence, and its corresponding type, such as persons, locations, and organizations. It provides a fundamental support for a wide range of upstream natural language processing (NLP) tasks, such as relation extraction (Hendrickx et al. 2009; Tang et al. 2008), semantic role labeling (Carreras 2004), and coreference resolution (Pradhan et al. 2012).

Jointly modeling NER with constituent parsing has been demonstrated as an effective way in improving the NER performance (Finkel and Manning 2009). Original joint models aim at solving both tasks. But in applications, if only NER is needed, the joint model significantly increases the computational cost from $O(n)$ in linear semi-CRF (Liu et al. 2016; Sarawagi and Cohen 2004), to $O(n^3)$ in tree-CRF (Finkel and Manning 2009; Finkel, Kleeman, and Manning 2008; Hall, Durrett, and Dan 2003). This makes the joint model not applicable in many cases in practice.

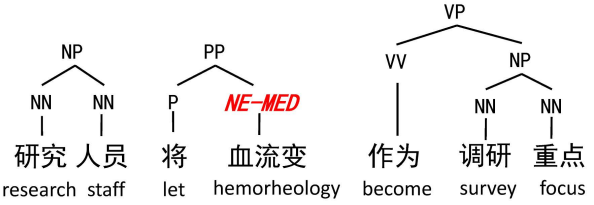

Figure 1: An illustration example of NER with height-limited constituent parsing (height limit = 2).

Aiming at solving this limitation, we reformulate the problem, and let the NER be jointly modeled with height-limited constituent parsing. The ground truth structure of height-limited constituent parsing is generated by cutting the nodes, which exceed the height limit, from the original constituent parsing tree. Figure 1 illustrates an example for a Chinese sentence, with the height limited to 2. Although height-limited parsing is not a complete parsing result, many phrases can indeed be extracted, such as noun phrases and preposition phrases, which are also informative for upstream applications. On one hand, for improving the NER accuracy, our main assumption is if an entity is covered by a subtree within a certain height from the original parsing tree, the subtree can also provide supportive information from the aspect of parsing. We conduct a statistical analysis on the Chinese corpus of the OntoNotes 4.0 dataset (Weischedel et al. 2011), which contains 15,700 sentences and 13,372 entity instances. We show how many entities can be covered by constituent grammars if the height of the parsing tree is limited, in Fig. 2. It is observed, if the height limit is set to be 3, near 80% of the entities can be covered by the subtree. This demonstrates that joint modeling NER and height-limited constituent parsing subtrees can indeed influence a majority of entities. On the other hand, for improving the computational cost, with the height of the parsing tree limited to a constant, the complexity of the joint model is reduced to $O(n)$ theoretically, which is comparable with previous linear CRF models for NER.

In this paper, we investigate how to build a joint model, in order to utilize height-limited constituent parsing structures to improve Chinese NER. Specifically, we investigate the problem under the graph-based CRF framework,

*the corresponding author.

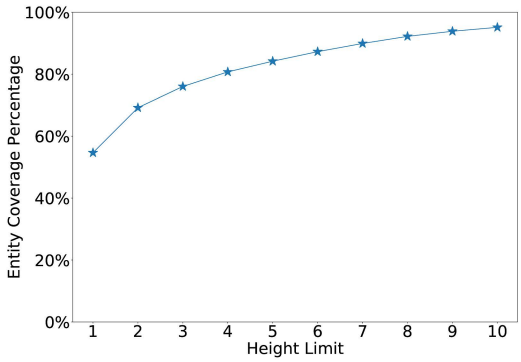

Figure 2: Entity Coverage Rate of Different Height Limits.

as it has been demonstrated as the state-of-the-art framework for NER. For the Chinese language, word segmentation is an important issue, which is tightly correlated with NER (Zhang and Yang 2018). Consequently, our idea is to build a joint model, which simultaneously solves the tasks of word segmentation, part-of-speech (POS) tagging, NER and height-limited constituent parsing. We propose an unified model of the semi-CRF (Kong, Dyer, and Smith 2016; Liu et al. 2016) and the tree-CRF (Durrett and Klein 2015), for this task. The semi-CRF is for segmentation structure learning, and the tree-CRF is for parsing structure learning. Both have been demonstrated successful in previous work. POS tags, name entities, and constituent grammars serve as labels in the combined structure.

The main challenge is how to learn and infer the unified CRF framework. It has not been thoroughly solved in previous joint models. Some work (Finkel and Manning 2009) builds a tree-CRF model for NER and parsing. As the task is for English only, the tree-CRF cannot well deal with the Chinese word segmentation issue. Some other work (Qian and Liu 2012) builds a joint model of semi-CRF and tree-CRF for word segmentation, POS tagging, and parsing. But 1-order semi-CRF has been simplified to 0-order semi-CRF in this work. This means the joint model loses the relational dependency of adjacent segments, which is important for segmentation.

In this paper, we solve the above challenge from two aspects. First, the dynamic programming algorithm is designed for the unified model of the semi-CRF and the tree-CRF, where 1-order semi-CRF is retained. By using the algorithm, for both training and inference, the computational complexity is $O(n \cdot 4^h)$, where n is the character number in a sentence, and h is the height limit (h=3 in our setting). Second, a pruning algorithm is derived, under the framework of structured prediction cascades (Weiss, Sapp, and Taskar 2012). From empirical statistics, when the search space is reduced to 0.001 of the original space, only 2% of the ground truth data are missed. This further reduces the time complexity, to make it comparable with semi-CRF. Finally, an unified model of the semi-CRF and the tree-CRF is constructed, based on neural features generated from word embeddings, which simultaneously solves the word segmentation, POS tagging, NER, and height-limited constituent parsing. Our work has three primary contributions.

- *A Novel Joint Formulation for Pipeline Tasks*. Height-limited constituent parsing is introduced, which significantly reduces the search space of jointly modeling word segmentation, POS tagging, NER, and parsing.
- *Joint CRF Framework of Segmentation and Parsing*. To learn and infer the unified CRF model, both dynamic programming algorithm and pruning algorithm are designed, and neural features are explored.
- *Experimental Evaluation*. By conducting experiments on the OntoNotes 4.0 dataset, we demonstrate that the proposed approach outperforms previous algorithms by 2.79 points in the F1-measure.

Related Work

Named Entity Recognition

NER is typically formulated as a sequential labeling problem, and conditional random fields have been demonstrated as the state-of-the-art architecture (Lafferty, Mccallum, and Pereira 2001; Liu et al. 2016; Luo et al. 2016). The labels can be assigned to either words (Huang, Xu, and Yu 2015) or characters (Dong et al. 2016). Currently, character-based methods perform better than word-based methods (Li et al. 2014). Originally, features are manually defined (Nguyen, Moschitti, and Riccardi 2010). Recently, nonlinear neural features have enhanced the performance, including convolutional neural networks (CNN) (Collobert et al. 2011; Ma and Hovy 2016; Santos and Guimaraes 2015), long short-term memory (LSTM) (Hammerton 2003; He and Sun 2017; Huang, Xu, and Yu 2015; Lample et al. 2016; Rondeau and Su 2016; Zhang and Yang 2018), and others such as fixed-size ordinally forgetting encoding (Xu, Jiang, and Watcharawittayakul 2017). Knowledge base is also an effective external source to improve NER performance (Chiu and Nichols 2015; Radford, Carreras, and Henderson 2015), and some work jointly deal with the NER and entity linking (Luo et al. 2016; Sil and Yates 2013). The lattice-based method achieves the state-of-the-art performance (Zhang and Yang 2018).

The main difference is that we jointly model NER with word segmentation, POS tagging and parsing. Previously, some work employs the results of these accompanied tasks as features (Jie, Muis, and Lu 2017). The advantage of a joint multi-task model is that it can learn the direct dependencies among the labels of these tasks, and therefore be avoid of error propagations.

Joint Segmentation and Parsing

Previous joint models of segmentation and parsing are composed of transition-based (Hatori et al. 2012; Kurita, Kawahara, and Kurohashi 2017; Zhang et al. 2013) and graph-based (Goldberg and Elhadad 2011; Qian and Liu 2012; Wang, Zong, and Xue 2013). In NER, previous work demonstrate that graph-based methods perform better. Thus we focus on graph-based methods. Previous graph-based models

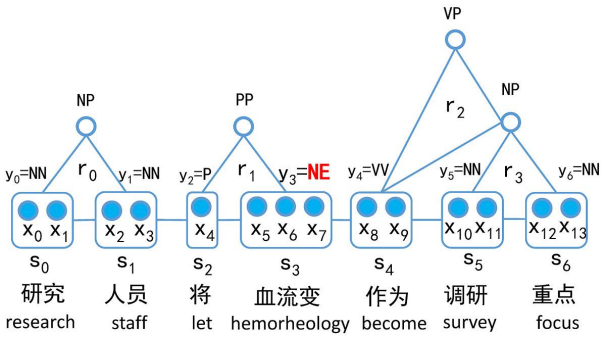

Figure 3: The probabilistic graph of the joint CRF.

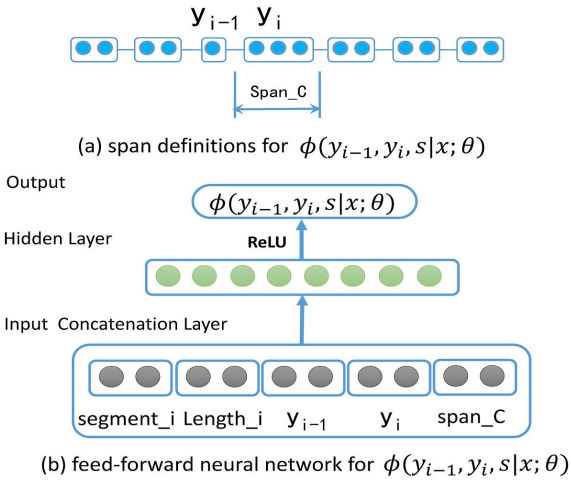

Figure 4: Neural feature for $\phi(y_{i-1}, y_i, s|x;\theta)$.

can be further divided into two categories. The first one is a two-stage model (Green and Manning 2010; Goldberg and Elhadad 2011; Wang, Zong, and Xue 2013). Words are segmented into word lattices first, and then fed to the parsing model. The second category is to jointly infer segmentation and parsing (Qian and Liu 2012).

The main difference of our work is that we retain all the relational structure features, but previous work cut the semi-CRF from 1-order to 0-order. The advantage of cutting is that the previous dynamic programming algorithm for tree-CRF can be directly employed. But the disadvantage is that the dependencies among adjacent segmentations are ignored. We design a novel dynamic programming for the joint model of 1-order semi-CRF and tree-CRF, and also derive a pruning algorithm for this joint model.

The Joint CRF Model

The Probabilistic Graph

Before constructing the joint model, we simplify the Chinese treebank from three aspects. (1) *POS and NE label combination*. We merge the POS label space and the NE label space. If an entity is composed by more than one word, the words are merged together into an entity. (2) *Unary rule elimination* . All unary rules are removed, and only the top label is retained. (3) *Binarization of rules*. We employ the ZPar system (Zhang and Clark 2011) for the binarization of rules.

The notations of the joint model follow JERL (Luo et al. 2016) and CRF-CFG (Finkel, Kleeman, and Manning 2008). Let $x = \{x_i\}$ be a sentence, which is composed of a sequence of characters, with x_i being the i^{th} character. Segmentation and parsing act as two kinds of structure for x. In segmentation, let $s = \{s_i\}$ be a legal segmentation for the sequence of x. $s_i = (u_i, v_i)$ denotes the i^{th} segment. Its boundary starts from u_i and ends at v_i, where $0 \leq u_i \leq v_i \leq |x|$ and $u_{i+1} = v_i + 1$. Let $y = \{y_i\}$ be the label assignments for the segments, where y_i denotes the label of the i^{th} segment. The label space of segmentation, denoted by $\mathcal{Y}_{seg}$, where $y_i \in \mathcal{Y}_{seg}$, is an union set of alphabet POS labels and alphabet NER labels. In parsing, after simplification, each sentence in the Chinese treebank is generated by a context-free grammar (CFG). The CFG is defined by, (1) a set of terminals $\mathcal{Y}_{seg}$; (2) a set of non-terminals $\{N^i\}, i = 1, ..., n$; (3) a designated start symbol N^1; and (4) a set of rules $U = \{\rho_j\}$, where $\rho_j = N^i \rightarrow \zeta^a \zeta^b$ and $\zeta \in \mathcal{Y}_{seg} \cup \{N^i\}$. Given s for the sequence of x, let t be a legal parsing tree on s. We utilize $r \in t$ to denote a one-level three-node subtree of t, which corresponds to a rule ρ.

The probabilistic graph of a sentence is shown in Fig. 3. Following the work of neural semi-CRF (Liu et al. 2016) and neural tree CRF (Durrett and Klein 2015), two kinds of energy potentials are defined in local cliques. $\phi(y_{i-1}, y_i, s|x;\theta)$ models relational features of adjacent segmentation labels, given the observations of x. $\phi(r|x;\theta)$ models the relational features of the parsing labels of a subtree r, which corresponds to the parsing rule, given x. θ is the parameters of energy potentials.

Based on the above two feature templates, the conditional probability of the entire sentence, $P(t, y, s|x;\theta)$, is defined by the standard CRF, as shown in the following equation. Since the proposed CRF is a joint model of segmentation and parsing, the normalizer Z_x is the sum energy potential of all possible segmentations, and all possible parsing trees given a segmentation. $\varphi(x)$ is the set of all segmentations, given the sentence x. $\psi(s', x)$ is the set of all segment label assignments, given x and $s' \in \varphi(x)$. $\tau(y', s', x)$ is the set of all parsing trees and the corresponding assigned rules, given x, $s' \in \varphi(x)$, and $y' \in \psi(s', x)$.

$$P(t, y, s|x;\theta) = \frac{1}{Z_x} \cdot \exp \left(\sum_{i=1}^{|y|} \phi(y_{i-1}, y_i, s|x;\theta) + \sum_{r \in t} \phi(r|x;\theta) \right)$$

$$Z_x = \sum_{s' \in \varphi(x)} \sum_{y' \in \psi(s', x)} \sum_{t' \in \tau(y', s', x)} \exp \left(\sum_{i=1}^{|y|} \phi(y'_{i-1}, y'_i, s'|x;\theta) + \sum_{r \in t'} \phi(r|x;\theta) \right)$$

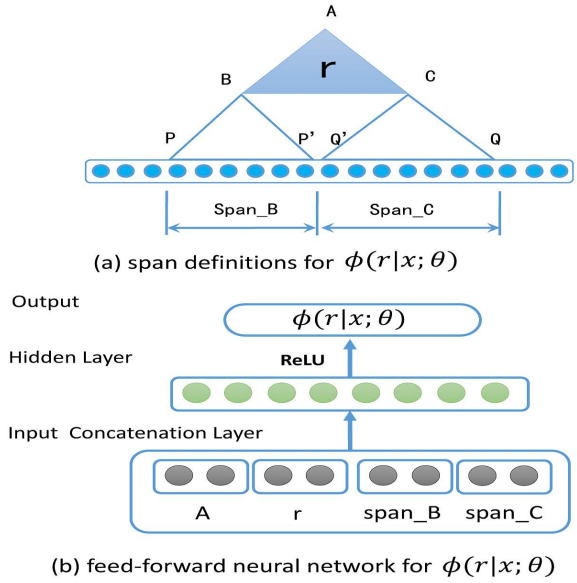

Figure 5: Neural feature for $\phi(r|x;\theta)$.

Neural Features

Span Feature Representation We employ three previous methods to represent the span feature. The first one is based on an independent RNN in the span (Kong, Dyer, and Smith 2016). The second one is based on the subtraction of hidden states (Cross and Huang 2016; Wang and Chang 2016). The third one is based on an attention mechanism (Lee et al. 2017). Lattice LSTM (Zhang and Yang 2018) is employed to encode the character sequence.

Constructing $\phi(y_{i-1}, y_i, s|x;\theta)$ The neural feature construction for two adjacent POS/NE labels, $\phi(y_{i-1}, y_i, s|x;\theta)$, is shown in Fig. 4. The feature is defined by a feed-forward neural network, with ReLU as the active function. The input is the concatenation of several component feature vectors. The first four components are embedding feature vectors for segment-i, length-i, label y_{i-1}, and label y_i, and the last component is the span feature vector. The embedding of segment-i is obtained by looking up a pre-learned word embedding knowledge base. The embeddings of others are learned together with the feed-forward neural network.

Constructing $\phi(r|x;\theta)$ The neural feature construction for $\phi(r|x;\theta)$ is shown in Fig. 5. The feature is also defined by a feed-forward neural network, with the concatenation of several component feature vectors as the input vector. The first two components are embedding feature vectors for the parsing label and the parsing rule, and the last two components are span feature vectors.

Algorithms

Dynamic Programming Algorithm

The parameters of the joint model, denoted by θ, include the weights of the neural network in the two kinds of feature potentials and some embeddings that are not pre-learned.

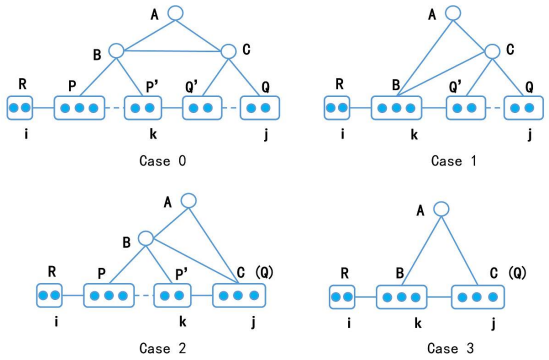

Figure 6: Four cases in the dynamic programming.

The learning process is to find θ to maximize the log conditional likelihood of the training set $\mathcal{D} = \{(t^{(k)}, y^{(k)}, s^{(k)}, x^{(k)})\}$ as the following equation.

$$\mathcal{L}(\mathcal{D};\theta) = \sum_{(t^{(k)},y^{(k)},s^{(k)},x^{(k)})\in\mathcal{D}}$$

$$[(\sum_{i=1}^{|y^{(k)}|}\phi(y_{i-1}^{(k)}, y_i^{(k)}, s_i^{(k)}) + \sum_{r\in t^{(k)}}\phi(r)) - \log Z_{x^{(k)}}]$$

In calculating $Z_{x^{(k)}}$, a dynamic programming algorithm is designed. Given a character span (i, j), two iteration functions $\alpha(i, j, A, Q, R)$ and $\beta(i, j, A, R)$ are defined. Q and R are POS/NE labels. A, B and C can be either POS/NE labels or parsing labels.The calculation of α and β follows the dynamic programming manner, as shown in the following equations. The four indicator functions correspond to four cases, which is shown in Fig. 6.

$$\begin{aligned}
&\alpha(i,j,A,Q,R) = \\
&I_{(h_B<h,h_C<h)}\cdot\sum_{B,C}\sum_{k}\exp\left(\phi\left(A\rightarrow BC|i,j,k\right)\right) \\
&\{ \\
&\quad I_0\cdot\sum_{P'}\alpha(i,k,B,P',R)\cdot\alpha(k,j,C,Q,P') \\
&+I_1\cdot\beta(i,k,B,R)\cdot\alpha(k,j,C,Q,B) \\
&+I_2\cdot\sum_{P'}\alpha(i,k,B,P',R)\cdot\beta(k,j,C,P') \\
&+I_3\cdot\beta(i,k,B,R)\cdot\beta(k,j,C,B) \\
&\}
\end{aligned}$$

$$\beta(i,j,A,R) = \exp\left(\phi(R,A|i,j)\right)$$

$$\xi(j,Q) = \sum_{i,j,A,Q,R}\xi(i,R)\alpha(i,j,A,Q,R)$$

$$Z_{x^{(k)}} = \sum_{Q}\xi(|x^{(k)}|,Q)$$

The inference is to find a group of (t, y, s) for a sentence x to maximize the conditional probability. Dynamic programming can also be utilized by substituting the sum function to the maximizing function in the above equations.

The complexity of the training and inference of the above process is $O(n \cdot L \cdot q^2 \cdot 4^h \cdot |U|)$, where n is the number of characters in the sentence, L is the maximum span length to be considered as word/entity, q is the number of POS/NE tags, h is the height limit, $|U|$ is the number of constituent rules.

Pruning Algorithm

The pruning algorithm is designed to reduce the number of atomic segments. Figure 7 shows an example of an atomic segment, it is defined by the segment index i and j, as well as its POS/NE tag P and the POS/NE tag R of its previous segment, and denoted as $c(i, j, P, R)$. It is a fundamental unit in the above dynamic programming. It is located at a leaf node of the parsing tree, and there is no grammar rules inside. As the parsing tree is constructed on these atomic segments, if its total number can be reduced, the overall complexity of the joint model will be reduced linearly. If pruning is not conducted, there are $n \cdot L \cdot q^2$ atomic segments for a sentence. In practice, $n \approx 40$, $L = 10$, and $q \approx 30$. Thus the total number is 360,000. The pruning target is to remove unlikely segments and reduce this number to $m \cdot n \cdot L \cdot q^2$. For example, in our setting, $m = 0.001$. Then only 360 segments will be retained for constructing the parsing tree. Finally, the total complexity of the joint model is reduced to 0.001 of the original complexity, and the loss of the ground truth data is within 2%.

The key of pruning is to calculate the max marginals of these atomic segments' energy potential. Our pruning algorithm is under the framework of structured prediction cascades (Weiss, Sapp, and Taskar 2012), but we derive and implement it for the proposed CRF model. The dynamic programming starts from the left to the right. As shown in Fig. 7, at position i, given that an atomic segment ends at i with the POS tag R, the maximum energy potential from the left to i is denoted as $\delta(i, R)$. The dynamic programming is conducted as follows, where $f(.)$ is the energy potential on the segment of $c(a, i, R, T)$, and $\delta(0, .) = 1$ for initial case.

$$\delta(i, R) = \max_{a,T} \delta(a, T) \cdot f(a, i, R, T)$$

Similarly, if the dynamic programming is conducted from the right to the left, the maximum energy potential from the right to i, $\delta'(j, P)$, is as follows, where $\delta'(n, .) = 1$.

$$\delta'(j, P) = \max_{b,Q} \delta'(b, Q) \cdot f(j, b, Q, P)$$

Consequently, the max marginal of the segment $c(i, j, P, R)$ is

$$\gamma(i, j, P, R) = \delta(i, R) \cdot \delta'(j, P) \cdot f(i, j, P, R).$$

For a sentence with the length n, there are totally $n \cdot L \cdot q^2$ atomic segments. Therefore, we calculate the max marginal of each atomic segment, and rank them by their max marginal values in a descent order, as $\{c^{(1)}, c^{(2)}, ..., c^{(n \cdot L)}\}$, where the corresponding max marginals follow

$$\gamma^{(1)} \geq \gamma^{(2)} \geq ... \geq \gamma^{(n \cdot L)}.$$

In this manner, suppose $k = \lfloor m \cdot n \cdot L \cdot q^2 \rfloor$, $\{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$ is selected as the retained atomic segments after pruning. Please note if $\gamma^{(i)} = \gamma^{(i+1)} = ... = \gamma^{(i+j)}$, and $i \leq k \leq i + j$, then we set $k = i + j$.

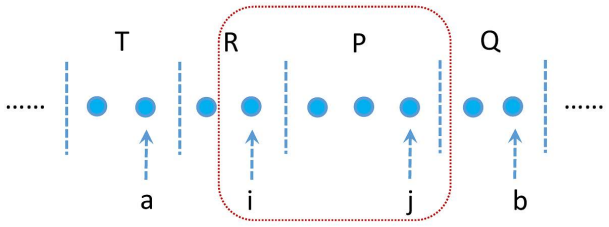

Figure 7: An illustration of atomic segment.

Suppose s is a segmentation strategy for a sentence, $C_s = \{c_{s1}, c_{s2}, ..., c_{sk}\}$ denotes its corresponding atomic segment set. $v(s)$ denotes the total energy potential of the strategy s.

Lamma 1. *For arbitrary atomic segment* $c^{(i)} \in \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$*, suppose* $s^{(i)}$ *is a strategy, where* $c^{(i)} \in C_{s^{(i)}}$*, and* $v(s^{(i)}) = \gamma(c^{(i)})$*. Then for arbitrary* $c_j \in C_{s^{(i)}}$*, we have* $c_j \in \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$.

Proof of Lemma 1. For arbitrary $c_j \in C_{s^{(i)}}$, its marginal in the strategy $s^{(i)}$ is equal to $\gamma(c^{(i)})$. It means its max marginal $\gamma(c_j) \geq \gamma(c^{(i)})$. As $c^{(i)} \in \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$, therefore, $c_j \in \{c^{(1)}, c^{(2)}, ..., c^{(i)}\} \subseteq \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$. □

Lamma 2. *For arbitrary atomic segment* $c^{(i)} \notin \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$*, suppose* $s^{(i)}$ *is arbitrary strategy that fits* $c^{(i)} \in C_{s^{(i)}}$*, with the corresponding total energy potential* $v(s^{(i)})$*. There must be a strategy* s*, where* $C_{s^{(i)}} \subseteq \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$*, having* $v(s) > v(s^{(i)})$.

Proof of Lemma 2. From the definition, if $c^{(i)} \notin \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$, we have $\gamma(c^{(k)}) > \gamma(c^{(i)})$. Suppose $s^{(k)}$ is a strategy with $v(s^{(k)}) = \gamma(c^{(k)})$. Then $v(s^{(k)}) \geq v(s^{(i)})$, as $\gamma(c^{(k)})$ is the max marginal of $c^{(k)}$. From Lemma 1, we have $C_{s^{(k)}} \subseteq \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$. Thus we have found the $s^{(k)}$ fitting the condition. □

Experiments

Experimental Settings

The experiments are conducted on the dataset of OntoNotes 4.0 (Weischedel et al. 2011). It is the unique dataset in the community, which have all the labels of word segmentation, POS tagging, NER, and parsing, in Chinese. We split the dataset in the same manner as previous work (Zhang and Yang 2018) for comparisons. The dataset contains 15,724 sentences in the training set, 4,301 sentences in the development set, and 4,346 sentences in the testing set, with more

Parameter	Value	Parameter	Value
char emb size	50	word emb size	50
tag emb size	20	rule emb size	40
length emb size	10	CRF hidden	128
LSTM layer	1	LSTM hidden	200
regularization λ	1e-08	AttSpan hidden	100
char/word dropout	0.5	LSTM dropout	0.5
learning rate (lr)	0.015	lr decay	0.05

Table 1: Hyper-parameters.

Models	P	R	F1	O-R	O-F1
Yang16a (gw)	65.59	71.84	68.57		
Yang16b (gw)	72.98	**80.15**	76.40		
Che13(gw)	77.71	72.51	75.02	–	–
Wang13(gw)	76.43	72.32	74.32		
Zhang18(gw)	**78.62**	73.13	75.77		
Zhang18(aw)	73.36	70.12	71.70		
Zhang18CRF	68.79	60.35	64.30	44.55	54.08
Zhang18Latt	76.35	71.56	73.88	60.04	67.22
SRSemiCRF	76.79	70.99	73.78	58.09	66.14
MiSemiCRF	76.41	73.19	74.77	60.59	67.59
AtSemiCRF	78.11	72.91	75.42	61.50	68.82
+POS+CWS	76.68	74.69	75.67	64.70	70.19
UnifiedCRF	77.18	76.16	**76.67**	**66.38**	**71.37**

Table 2: NER Performances on the testing set.

than 490,000 characters in total. Standard metrics of precision, recall, and F1-measure are utilized for evaluation.

The word embedding and character embeddings are employed from the work of (Zhang and Yang 2018). Other embeddings are trained as parameters with the joint model. The detailed embedding dimension information and all the configurations of the joint model are shown in Table 1. Some of them are employed from the work of (Zhang and Yang 2018), and others are set according to parameter analysis. The optimization is stochastic gradient descent with batch size = 1. The height limit is 3.

Performances

We follow the work of (Zhang and Yang 2018) for performance comparisons. The overall performances are shown in Table 2. The first block in the table is the word-based baseline methods (Che et al. 2013; Wang, Che, and Manning ; Yang et al. 2016; Zhang and Yang 2018), with gold word segmentation (gw) and automatic word segmentation (aw). The second block is the character-based baseline methods. "Zhang18CRF" denotes the BIO-CRF with LSTM features, and "zhang18Latt" denotes the BIO-CRF with lattice LSTM features. "SRSemiCRF" denotes the semi-CRF with LSTM span features (Kong, Dyer, and Smith 2016). The third block is our implemented semi-CRF methods. "MiSemiCRF" denotes the semi-CRF with substraction LSTM span features, and "AtSemiCRF" denotes the semi-CRF with attention-based span features. The last block is our proposed unified CRF model. "O-R" denotes the recall for OOV entities, and "O-F1" denotes the F1-measure with the overall precision and the OOV recall. The values of these two columns are calculated by our implementations.

Models	P	R	F1	O-R	O-F1
Zhang18(aw)	72.63	67.60	70.03	–	–
Zhang18CRF	67.12	58.42	62.47	40.47	50.49
Zhang18Latt	74.64	68.83	71.62	53.91	62.6
SRSemiCRF	76.55	68.59	72.35	55.71	64.49
MiSemiCRF	75.23	70.39	72.72	56.44	64.49
AtSemiCRF	76.04	70.33	73.25	58.27	66.12
+POS+CWS	75.91	74.07	74.98	62.82	68.75
UnifiedCRF	76.09	74.66	75.37	63.56	69.27

Table 3: NER Performances on the development set.

Task	Model	P	R	F1
Word Seg.	SemiCRF	95.31	95.29	95.30
	UnifiedCRF	95.62	95.28	95.45
POS Tagging	SemiCRF	84.02	83.97	83.99
	UnifiedCRF	84.55	84.25	84.40
Parsing	UnifiedCRF	59.00	68.01	63.19
Parsing Struct.	UnifiedCRF	64.69	74.57	69.28

Table 4: Performances for other tasks.

For the overall entities, the unified model is observed to outperform previous character-based methods by 2.79 points in F1 (from 73.88% to 76.67%), and it also outperforms previous word-based method with gold segmentation (76.40%). The improvement comes from two aspects. The first one is we explore more neural features for semi-CRF. It is observed by using the attention-based span feature (Lee et al. 2017), the performance of semi-CRF increases from 73.78% to 75.67% in F1, compared with previous LSTM span features. The second one is the improvement from the joint model, from 75.67% to 76.67% in F1. This demonstrates that the parsing, as well as the tasks of word segmentation and POS parsing, improve the NER performance. We conduct an error analysis, on entities with different lengths in the testing set. The length refers to the number of characters in an entity. It is observed when the length increases, the F1-measure drops accordingly. When the entity length is above 7 (with around three or more words), the F1 of SemiCRF is 70.26% and the F1 of UnifiedCRF is 74.42%.

For the OOV entities, our model outperforms previous methods by 6.34 points in recall, and 4.15 points in F1. This has demonstrated that the joint model obtained significant improvement for the OOV entities. Table 3 shows the performances on the development set, where observations can be obtained. Figure 8 shows the performances with different sentence lengths and different NER types.

Table 4 shows comparisons between semi-CRF and the unified CRF on the tasks of word segmentation, POS parsing, and height-limited constituent parsing. Although these performances are not the focus of the paper, we still observe that the joint model also achieves improvements on word segmentation and POS tagging. "Parsing" denotes the performances for height-limited constituent parsing, and "Parsing Struct." denotes the parsing structure without labels.

Table 5 shows the processing time of different methods in the testing set, which contains 4636 sentences. The time

Model	Total Time	Avg. /Sentence
Linear-chain CRF	5min	0.07s
SemiCRF+POS+CWS	50min	0.70s
UnifiedCRF	70min	1.00s

Table 5: The processing time of different methods.

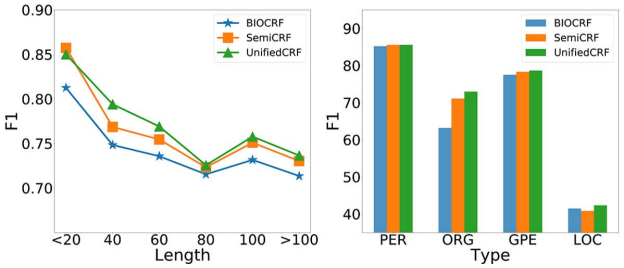

Figure 8: Performances with different sentence length and different entity types.

cost for BIO-CRF, semi-CRF, unified-CRF is compared. It is observed that time complexity of the proposed unified model is comparable with previous semi-CRF.

Figure 9 shows the loss of ground truth with different retaining rate of search space, from 0.0001 to 0.0010. It can be observed that the proposed pruning method is very effective for reducing the search space. We set the rate be 0.001, with the loss of ground truth data being less than 2%.

Figure 10 illustrates a case study performed by the proposed unified CRF, where the NER can be improved with the help of grammar rules. In this example, "DaRunFa" is a market name, which is an OOV entity. In both BIO and semi-CRF, the entity "DaRunFa" cannot be recognized. But with our proposed model, it can be successfully labeled.

Conclusion

In this paper, we investigate the problem of jointly modeling NER and parsing, to promote Chinese NER performances. We reformulate the parsing task to height-limited constituent parsing, which significantly reduces the computational cost. An unified model of neural semi-CRF and neural tree-CRF is proposed with designed dynamic programming and pruning algorithms. Experimental results demonstrated that the proposed unified model outperforms previous methods by 2.79 point in the F1-measure.

Acknowledgments

The work is supported by grants from the National Natural Science Foundation of China (61672100), the National Key Research and Development Program of China under Grant (2017YFB0803302), the Joint Research Fund in Astronomy (U1531242) under cooperative agreement between the National Natural Science Foundation of China (NSFC) and Chinese Academy of Sciences (CAS).

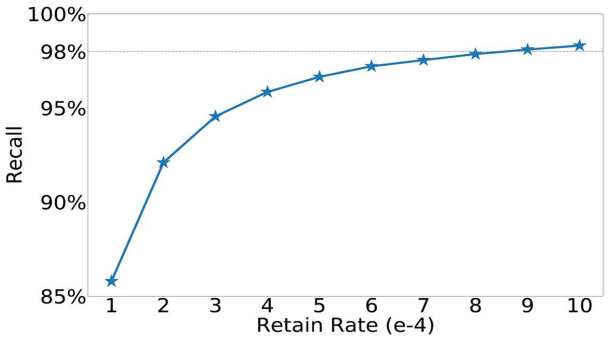

Figure 9: Pruning performances in different settings.

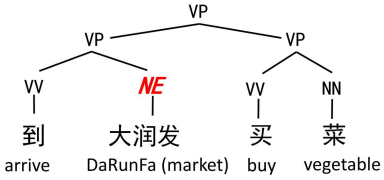

Figure 10: A case study of the unified model.

References

Carreras, X. 2004. Introduction to the conll-2004 shared task : Semantic role labeling. In *Proc. of CoNLL 2004*, 5–9.

Che, W.; Wang, M.; Manning, C. D.; and Liu, T. 2013. Named entity recognition with bilingual constraints. *Proc. of ACL 2013* 52–62.

Chiu, J. P. C., and Nichols, E. 2015. Named entity recognition with bidirectional lstm-cnns. *TACL 2015* 4(0):357–370.

Collobert, R.; Weston, J.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(1):2493–2537.

Cross, J., and Huang, L. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proc. of EMNLP 2016*, 1–11.

Dong, C.; Zhang, J.; Zong, C.; Hattori, M.; and Di, H. 2016. *Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition*. Springer International Publishing.

Durrett, G., and Klein, D. 2015. Neural crf parsing. *Proc. of ACL 2015* 302–312.

Finkel, J. R., and Manning, C. D. 2009. Joint parsing and named entity recognition. *Proc. of NAACL 2009* 326–334.

Finkel, J. R.; Kleeman, A.; and Manning, C. D. 2008. Efficient, feature-based, conditional random field parsing. In *Proc. of ACL 2008*, 959–967.

Goldberg, Y., and Elhadad, M. 2011. Joint hebrew segmentation and parsing using a pcfg-la lattice parser. In *Proc. of ACL 2011*, 704–709.

Green, S., and Manning, C. D. 2010. Better arabic parsing: Baselines, evaluations, and analysis. *Proc. of COLING 2010* 394–402.

Hall, D.; Durrett, G.; and Dan, K. 2003. Less grammar, more features. In *Proc. of ACL 2003*, 228–237.

Hammerton, J. 2003. Named entity recognition with long short-term memory. In *Proc. of NAACL 2003*, 172–175.

Hatori, J.; Matsuzaki, T.; Miyao, Y.; and Tsujii, J. 2012. Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. *Proc. of ACL 2012* 1045–1053.

He, H., and Sun, X. 2017. F-score driven max margin neural network for named entity recognition in chinese social media. *Proc. of EACL 2017* 713–718.

Hendrickx, I.; Su, N. K.; Kozareva, Z.; Nakov, P.; Pennacchiotti, M.; Romano, L.; and Szpakowicz, S. 2009. Semeval-2010 task 8: multi-way classification of semantic relations between pairs of nominals. 94–99.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional lstm-crf models for sequence tagging. *Computer Science*.

Jie, Z.; Muis, A. O.; and Lu, W. 2017. Efficient dependency-guided named entity recognition. In *Proc. of AAAI 2017*.

Kong, L.; Dyer, C.; and Smith, N. A. 2016. Segmental recurrent neural networks. *Proc. of ICLR 2016*.

Kurita, S.; Kawahara, D.; and Kurohashi, S. 2017. Neural joint model for transition-based chinese syntactic analysis. In *Proc. of ACL 2017*, 1204–1214.

Lafferty, J. D.; Mccallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML 2011*, 282–289.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. *Proc. of NAACL 2016* 260–270.

Lee, K.; He, L.; Lewis, M.; and Zettlemoyer, L. 2017. End-to-end neural coreference resolution. *Proc. of EMNLP 2017* 188–197.

Li, H.; Hagiwara, M.; Li, Q.; and Ji, H. 2014. Comparison of the impact of word segmentation on name tagging for chinese and japanese. *language resources and evaluation* 2532–2536.

Liu, Y.; Che, W.; Guo, J.; Qin, B.; and Liu, T. 2016. Exploring segment representations for neural segmentation models. *Proc. of IJCAI 2016* 2880–2886.

Luo, G.; Huang, X.; Lin, C. Y.; and Nie, Z. 2016. Joint entity recognition and disambiguation. In *Proc. of EMNLP 2016*, 879–888.

Ma, X., and Hovy, E. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *Proc. of ACL 2016*.

Nguyen, T. V. T.; Moschitti, A.; and Riccardi, G. 2010. Kernel-based reranking for named-entity extraction. In *Proc. of COLING 2010*, 901–909.

Pradhan, S.; Moschitti, A.; Xue, N.; Uryupina, O.; and Zhang, Y. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. 1–40.

Qian, X., and Liu, Y. 2012. Joint chinese word segmentation, pos tagging and parsing. *Proc. of EMNLP 2012* 501–511.

Radford, W.; Carreras, X.; and Henderson, J. 2015. Named entity recognition with document-specific kb tag gazetteers. In *Proc. of EMNLP 2015*, 512–517.

Rondeau, M. A., and Su, Y. 2016. Lstm-based neurocrfs for named entity recognition. In *Proc. of INTERSPEECH 2016*, 665–669.

Santos, C. N. D., and Guimaraes, V. 2015. Boosting named entity recognition with neural character embeddings. *Computer Science*.

Sarawagi, S., and Cohen, W. W. 2004. Semi-markov conditional random fields for information extraction. In *Proc. of ICONIP 2004*, 1185–1192.

Sil, A., and Yates, A. 2013. Re-ranking for joint named-entity recognition and linking. In *Proc. of CIKM 2013*, 2369–2374.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: extraction and mining of academic social networks. In *Proc. of SIGKDD 2018*, 990–998.

Wang, W., and Chang, B. 2016. Graph-based dependency parsing with bidirectional lstm. In *Proc. of ACL 2016*, 2306–2315.

Wang, M.; Che, W.; and Manning, C. D. Effective bilingual constraints for semi-supervised learning of named entity recognizers.

Wang, Z.; Zong, C.; and Xue, N. 2013. A lattice-based framework for joint chinese word segmentation, pos tagging and parsing. *Proc. of ACL 2013* 623–627.

Weischedel, R.; Palmer, M.; Marcus, M.; Hovy, E.; Pradhan, S.; Ramshaw, L.; Xue, N.; Taylor, A.; Kaufman, J.; and Franchini, M. 2011. Ontonotes release 4.0. *Linguistic Data Consortium*.

Weiss, D.; Sapp, B.; and Taskar, B. 2012. Structured prediction cascades. *Journal of Machine Learning Research*.

Xu, M.; Jiang, H.; and Watcharawittayakul, S. 2017. A local detection approach for named entity recognition and mention detection. In *Proc. of ACL 2017*, 1237–1247.

Yang, J.; Teng, Z.; Zhang, M.; and Zhang, Y. 2016. Combining discrete and neural features for sequence labeling. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 140–154.

Zhang, Y., and Clark, S. 2011. Syntactic processing using the generalized perceptron and beam search. *Comput. Linguist.* 37(1):105–151.

Zhang, Y., and Yang, J. 2018. Chinese ner using lattice lstm. *Proc. of ACL 2018*.

Zhang, M.; Zhang, Y.; Che, W.; and Liu, T. 2013. Chinese parsing exploiting characters. *Proc. of ACL 2013* 125–134.