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]Beijing Institute of Technology
[2]Tencent

Index

- Chinese Named Entity Recognition
- Motivation & Challenges
- The Proposed Joint Neural CRF Model
- Algorithms for the Joint Model
- Experimental Verifications
- Conclusions

Named Entity Recognition

MEDICAL ENTITY

研究	人员	将	血流变	作为	调研	重点
research	staff	let	hemorheology	become	survey	focus

Named entity recognition (NER) is to identify the ***boundaries*** as well as corresponding ***type*** of a named entity in a natural language sentence.

Named Entity Recognition

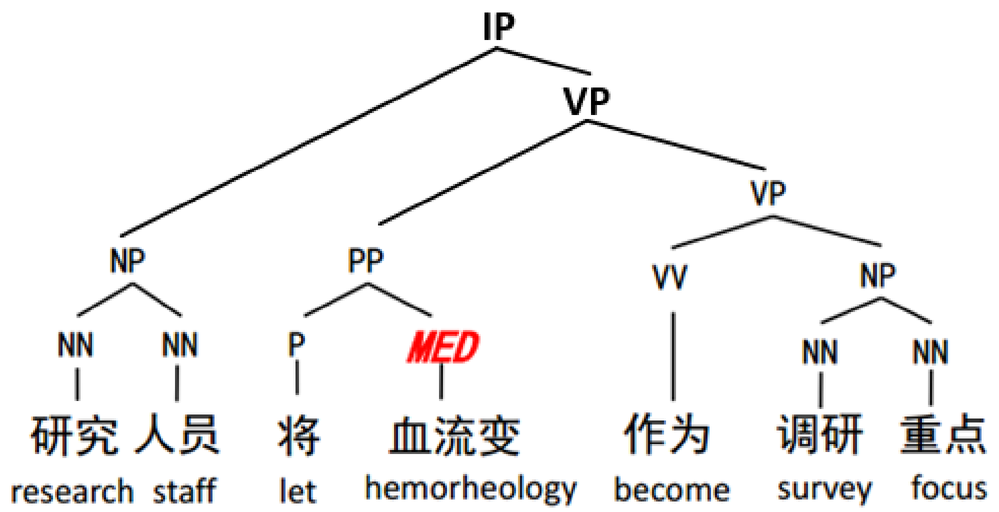
NER provides a fundamental support for a wide range of upstream natural language processing (NLP) tasks:

1. Relation extraction
2. Semantic role labeling
3. Co-reference resolution
4. …

Index

- Chinese Named Entity Recognition
- Motivation & Challenges
- The Proposed Joint Neural CRF Model
- Algorithms for Proposed Joint Model
- Experimental Verifications
- Conclusions

Motivation



1. Improving NER performance by exploiting structural dependency pattern among NER and parsing.
2. Guaranteeing the consistency of NER and parsing labels with joint model.

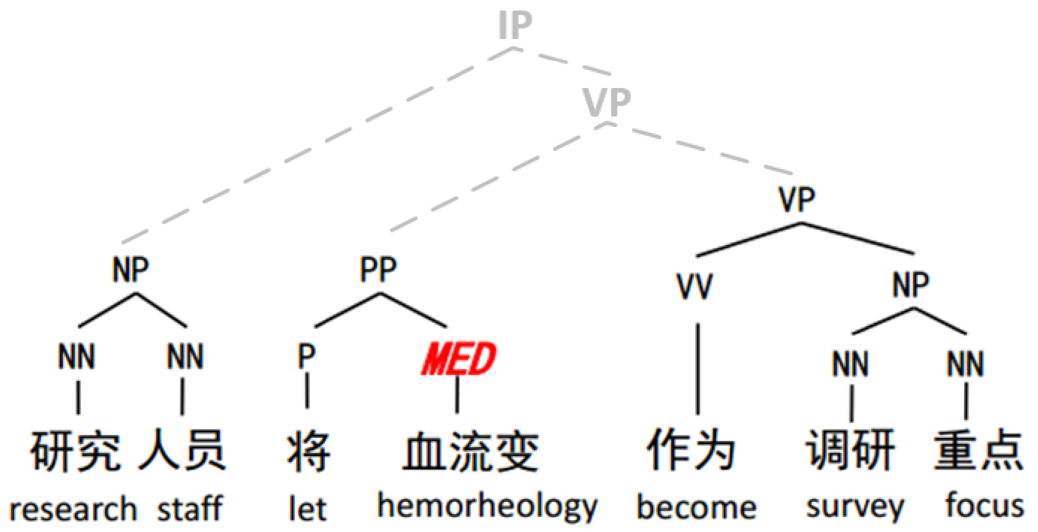
Challenge #1

- **Challenge**: High Time Complexity of tree-CRF Parsing Model

  The joint model significantly increases the computational cost from $O(n)$ in linear semi-CRF, to $O(n^3)$ in tree-CRF

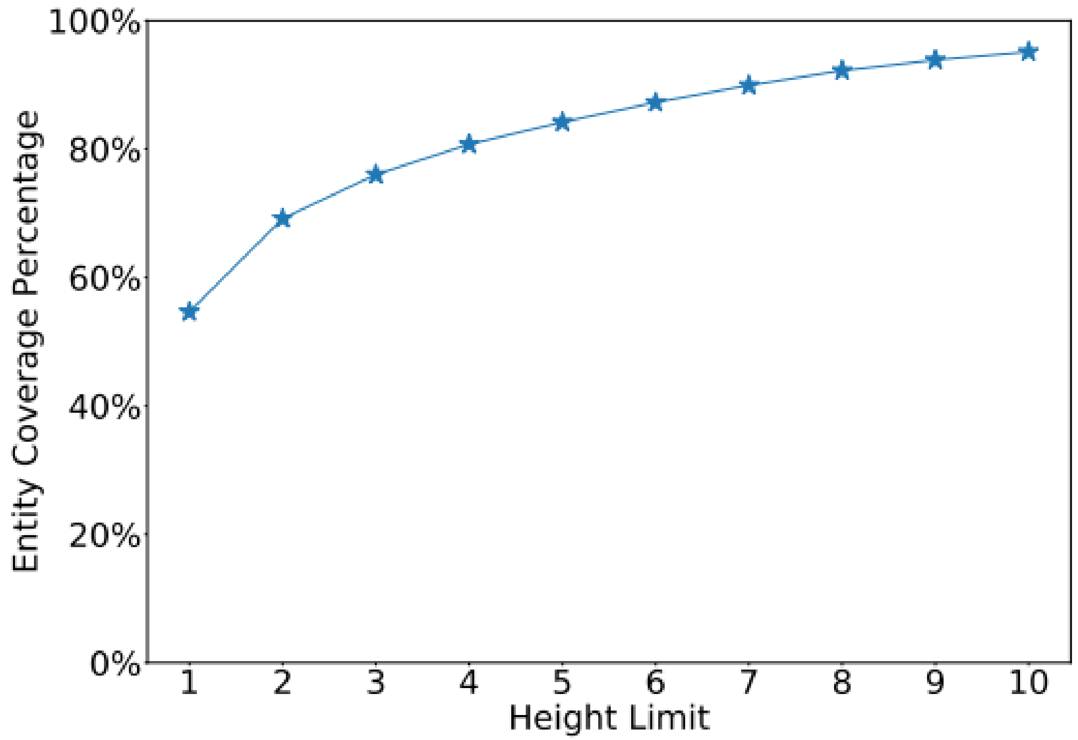
- **Our Solution**

  - We joint model NER with reformulated ***height limited constituent parsing.***
  - Nodes exceeding the height limit are removed.
  - The time complexity of height-limited parsing is $O(n \cdot 4^h)$, where h is the height limit.
  - CYK algorithm is modified for height limited search space.



Challenge #1

- **Our Solution**: Entity Coverage Rate Analysis
  - Conducted on the OntoNotes 4.0 corpus
  - With height limit set to be 3, near 80% of the entities can be covered by the sub-trees



Challenge #2

- **Challenge**: Segmentation Issue in Chinese
  - With gold word segmentation, previous joint model cut the semi-CRF from 1-order to 0-order, and directly employed tree-CRF to joint model NER and parsing.
  - For Chinese NER, the 1-order dependency is an import factor.
  - There is not a solution which joint models NER and parsing with 1-order semi-CRF dependency.
- **Our Solution**

  We design a novel dynamic programming for solving the joint model of 1-order semi-CRF and tree-CRF.

Challenge #3

- **Challenge**: High Time Complexity of Joint Model

  The combination of the search space in joint model leads to much higher time complexity than semi-CRF.

- **Our Solution**

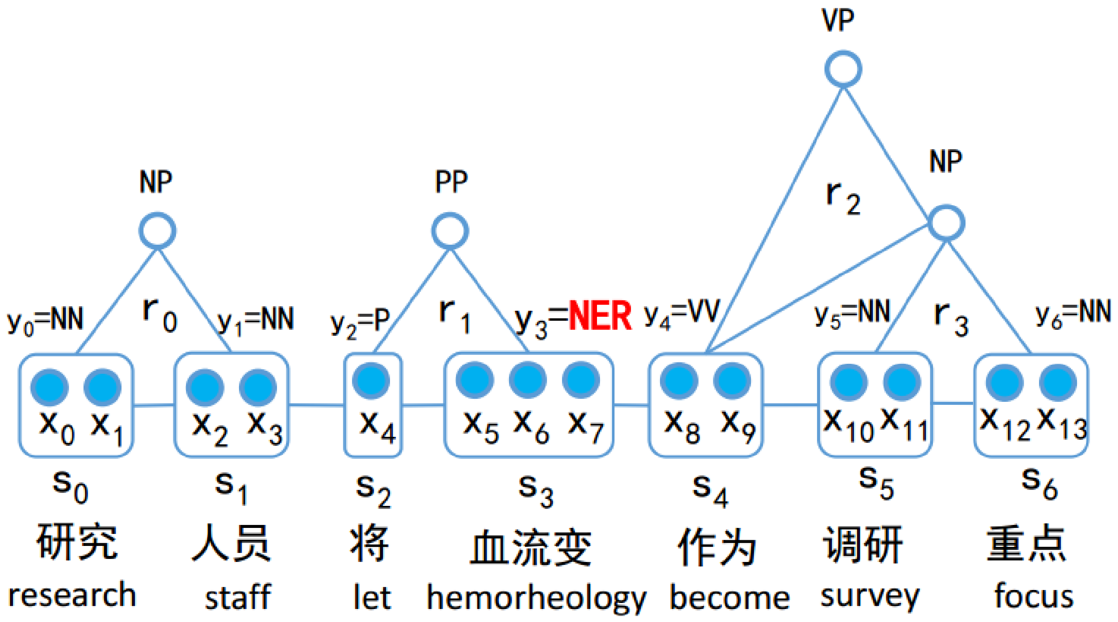
  1. We derive a pruning algorithm under the framework of structured prediction cascades.
  2. The joint search space is pruned using max marginal from semi-CRF.
  3. The pruning algorithm makes the time complexity comparable with semi-CRF.

Index

- Chinese Named Entity Recognition
- Motivation & Challenges
- The Proposed Joint Neural CRF Model
- Algorithms for the Joint Model
- Experimental Verifications
- Conclusions

The Probabilistic Graph

Observation	x_i	The i^{th} character
Segmentation	$s_i = (u_i, v_i)$	The i^{th} segment
	u_i and v_i	The boundary of $\boldsymbol{s_i}$
	y_i	The POS/NE label of $\boldsymbol{s_i}$
Parsing	r_i	The i^{th} CFG rule $N^i \rightarrow (\xi^a, \xi^b)$
	N^i	Non-terminal
	ξ^a	POS/NE label



The Conditional Probability

The Conditional Probability is defined by the standard conditional random field (CRF)

Two kinds of energy potentials:

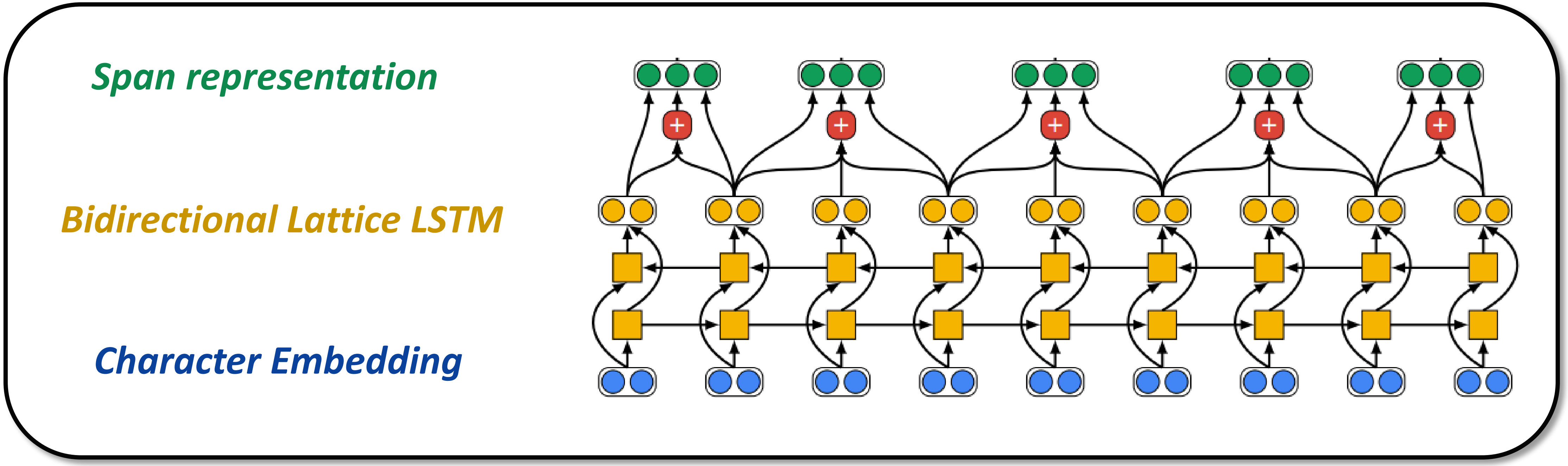
$\Phi(y_{i-1}, y_i \vert s, x; \theta)$	Adjacent Segmentation
$\Phi(r \vert y, s, x; \theta)$	Local Parsing Rule

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

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

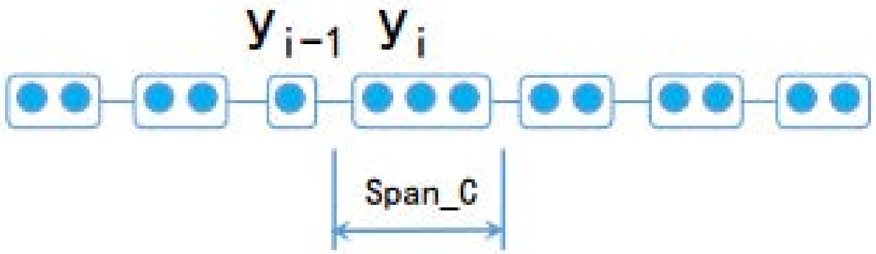
Neural Features for Observations

1. Character Embeddings are prelearned on external data
2. The character sequence is encoded using bidirectional Lattice LSTM (Zhang and Yang 2018)
3. The character spans are represented using an attention-based method (Lee et al. 2017)

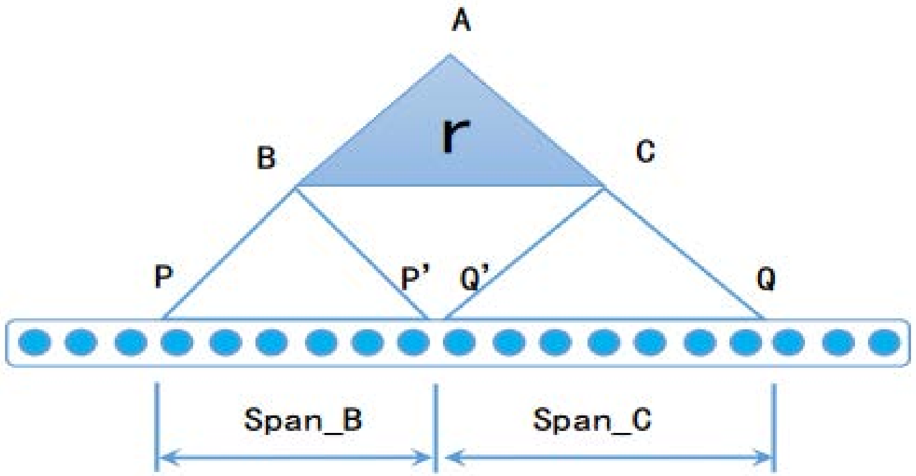


Character Spans

- Span definition for $\Phi(y_{i-1}, y_i | s, x; \theta)$



- Span definition for $\Phi(r | y, s, x; \theta)$



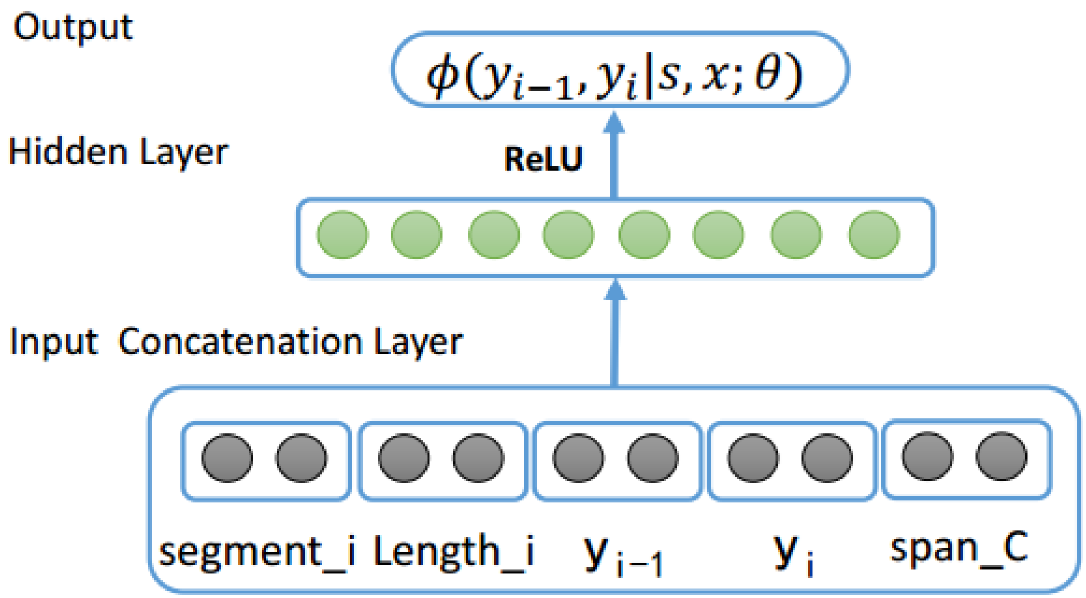
Neural Features for $\Phi(y_{i-1}, y_i|s, x; \theta)$

The relational features of two adjacent POS/NE label

- ***Input***
  1. The segment embedding vector
  2. The length embedding vector
  3. The POS/NE label embedding vectors
  4. The span representation vector

- ***Feed-forward neural network***
  1. The concatenation of the input vectors
  2. Hidden layers with RELU activation
  3. Linear output layer

- ***Output***
  - The energy potentials $\Phi(y_{i-1}, y_i|s, x; \theta)$



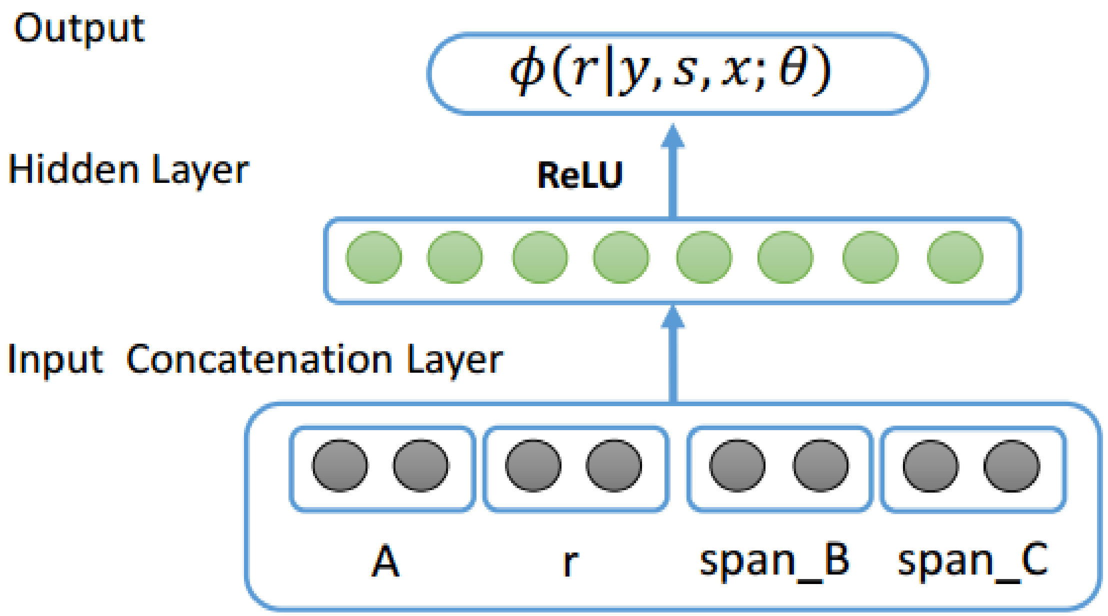
Neural Features for $\Phi(r|y,s,x;\theta)$

The relational features of the local parsing rule

- ***Input***
  1. The constituent label embedding vector
  2. The parsing rule embedding vector
  3. The left span representation vector
  4. The right span representation vector

- ***Feed-forward neural network***
  1. The concatenation of the input vectors
  2. Hidden layers with RELU activation
  3. Linear output layer

- ***Output***
  - The energy potentials $\Phi(r|y,s,x;\theta)$



Index

- Chinese Named Entity Recognition
- Motivation & Challenges
- The Proposed Joint Neural CRF Model
- Algorithms for the Joint Model
- Experimental Verifications
- Conclusions

Parameters Estimation

The parameters $\boldsymbol{\theta}$ are estimated by maximizing the log conditional likelihood of the training set $\mathcal{D}$.

Stochastic gradient descent (SGD) is employed for the optimization.

$$\mathcal{L}(\mathcal{D};\theta) = \sum_{(t^{(k)},y^{(k)},s^{(k)},x^{(k)})\in\mathcal{D}} \left[\left(\sum_{i=1}^{|x^{(k)}|} \phi(y_{i-1}^{(k)}, y_i^{(k)}) + \sum_{r\in t^{(k)}} \phi(r) \right) - \log Z_{x^{(k)}} \right]$$

The Dynamic Programming Algorithm

Novel dynamic programming algorithm to calculate $Z_{x^{(k)}}$ for the joint CRF model

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

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

The Dynamic Programming Algorithm

Novel dynamic programming algorithm to calculate $Z_{x(k)}$ for the joint CRF model

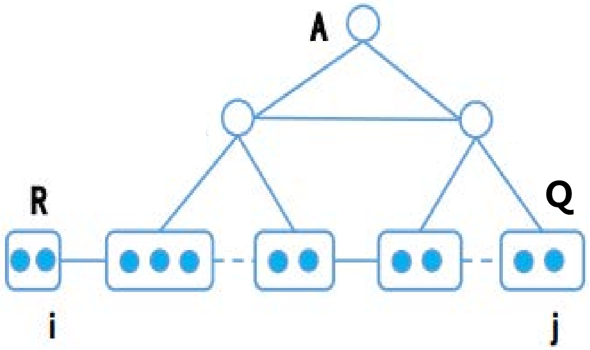
$$
\begin{aligned}
\alpha(i,j,A,Q,R) \quad &= \\
I_{(h_B<h, h_C<h)} \quad &\cdot \quad \sum_{A,B,C} \sum_{k} \phi(A \rightarrow BC, R | i,j) \\
\{ & \\
I_{case0} \quad &\cdot \quad \sum_{P'} \alpha(i,k,B,P',R) \cdot \alpha(k,j,C,Q,P') \\
+I_{case1} \quad &\cdot \quad \beta(i,k,B,R) \cdot \alpha(k,j,C,Q,B) \\
+I_{case2} \quad &\cdot \quad \sum_{P'} \alpha(i,k,B,P',R) \cdot \beta(k,j,C,P') \\
+I_{case3} \quad &\cdot \quad \beta(i,k,B,R) \cdot \beta(k,j,C,B) \\
\} &
\end{aligned}
$$

The Dynamic Programming Algorithm

- **Iteration Function**

$$\alpha(i, j, A, Q, R)$$

- i, j is the span boundary
- A is the parsing label
- Q, R are the POS/NE labels



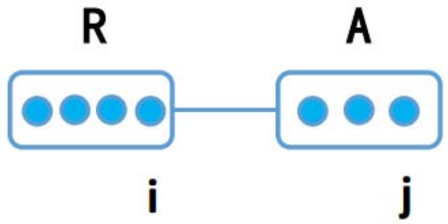
$$
\boxed{\alpha(i, j, A, Q, R)} =
$$
$$
I_{(h_B<h, h_C<h)} \cdot \sum_{A,B,C} \sum_k \phi(A \rightarrow BC, R|i, j)
$$
$$
\{
$$
$$
I_{case0} \cdot \sum_{P'} \alpha(i, k, B, P', R) \cdot \alpha(k, j, C, Q, P')
$$
$$
+I_{case1} \cdot \beta(i, k, B, R) \cdot \alpha(k, j, C, Q, B)
$$
$$
+I_{case2} \cdot \sum_{P'} \alpha(i, k, B, P', R) \cdot \beta(k, j, C, P')
$$
$$
+I_{case3} \cdot \beta(i, k, B, R) \cdot \beta(k, j, C, B)
$$
$$
\}
$$

The Dynamic Programming Algorithm

- **Energy Potentials for the Adjacent Labels**

$$\beta(i, j, A, R)$$

  - i, j is the segment boundary
  - A, R are the POS/NE labels



$$
\begin{aligned}
\alpha(i, j, A, Q, R) \quad = & \\
I_{(h_B<h, h_C<h)} \quad \cdot & \quad \sum_{A,B,C} \sum_k \phi(A \rightarrow BC, R | i, j) \\
\{ & \\
I_{case0} \quad \cdot & \quad \sum_{P'} \alpha(i, k, B, P', R) \cdot \alpha(k, j, C, Q, P') \\
+I_{case1} \quad \cdot & \quad \boxed{\beta(i, k, B, R)} \cdot \alpha(k, j, C, Q, B) \\
+I_{case2} \quad \cdot & \quad \sum_{P'} \alpha(i, k, B, P', R) \cdot \boxed{\beta(k, j, C, P')} \\
+I_{case3} \quad \cdot & \quad \boxed{\beta(i, k, B, R)} \cdot \boxed{\beta(k, j, C, B)} \\
\} &
\end{aligned}
$$

The Dynamic Programming Algorithm

- **Energy Potentials for the Local Sub-tree**

$$\sum_{A,B,C} \sum_{k} \phi(A \to BC, R|i,j)$$

  - $A \to BC$ is the parsing rule
  - i, j, k are the span/split positions

- **Height Limit for Sub-trees**

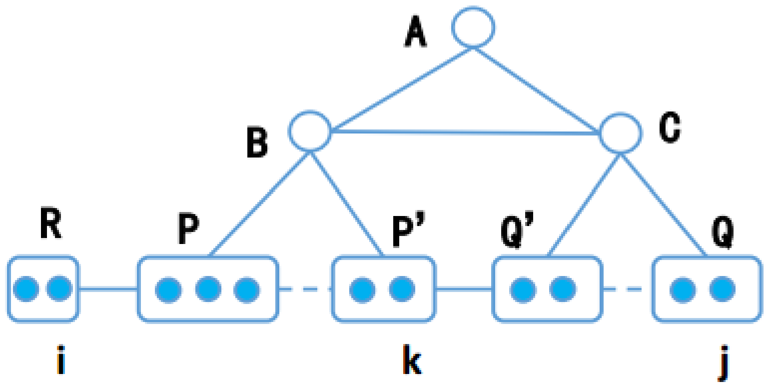
$$I_{(h_B<h, h_C<h)}$$

  - h_b, h_c are the heights of node B, C
  - h is the height limit

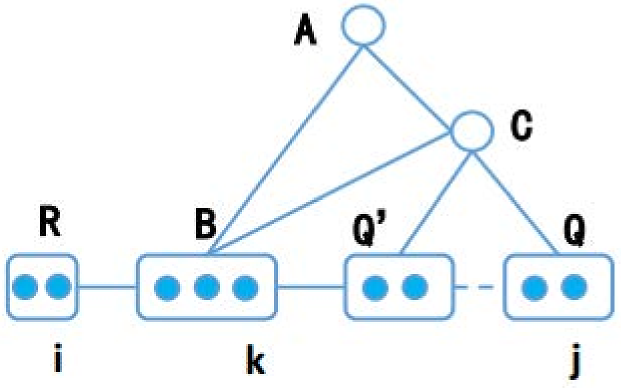
$$
\begin{aligned}
\alpha(i,j,A,Q,R) \quad = \\
\boxed{I_{(h_B<h, h_C<h)}} \quad \cdot \quad & \boxed{\sum_{A,B,C} \sum_{k} \phi(A \to BC, R|i,j)} \\
\{ \\
I_{case0} \quad \cdot \quad & \sum_{P'} \alpha(i,k,B,P',R) \cdot \alpha(k,j,C,Q,P') \\
+I_{case1} \quad \cdot \quad & \beta(i,k,B,R) \cdot \alpha(k,j,C,Q,B) \\
+I_{case2} \quad \cdot \quad & \sum_{P'} \alpha(i,k,B,P',R) \cdot \beta(k,j,C,P') \\
+I_{case3} \quad \cdot \quad & \beta(i,k,B,R) \cdot \beta(k,j,C,B) \\
\}
\end{aligned}
$$

The Dynamic Programming Algorithm

- **Case 0**



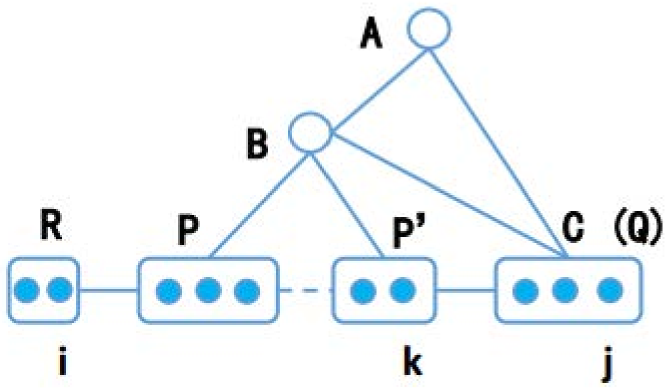
- **Case 1**



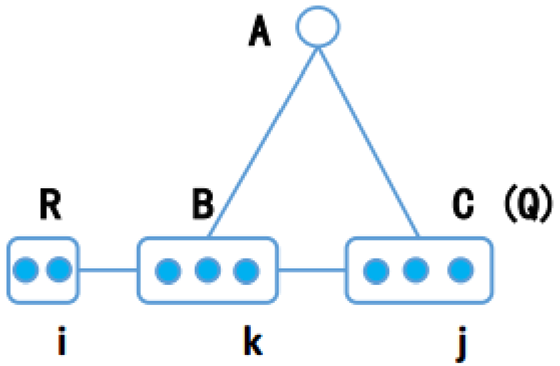
$$
\begin{aligned}
\alpha(i,j,A,Q,R) \quad = \quad & \\
I_{(h_B<h, h_C<h)} \quad \cdot \quad & \sum_{A,B,C} \sum_{k} \phi(A \to BC, R \mid i,j) \\
\{ & \\
\boxed{I_{case0}} \quad \cdot \quad & \sum_{P'} \alpha(i,k,B,P',R) \cdot \alpha(k,j,C,Q,P') \\
\boxed{+I_{case1}} \quad \cdot \quad & \beta(i,k,B,R) \cdot \alpha(k,j,C,Q,B) \\
+I_{case2} \quad \cdot \quad & \sum_{P'} \alpha(i,k,B,P',R) \cdot \beta(k,j,C,P') \\
+I_{case3} \quad \cdot \quad & \beta(i,k,B,R) \cdot \beta(k,j,C,B) \\
\} &
\end{aligned}
$$

The Dynamic Programming Algorithm

- **Case 2**



- **Case 3**



$$
\begin{aligned}
\alpha(i,j,A,Q,R) \quad = \\
I_{(h_B<h, h_C<h)} \quad \cdot \quad & \sum_{A,B,C} \sum_k \phi(A \rightarrow BC, R|i,j) \\
\{ \\
I_{case0} \quad \cdot \quad & \sum_{P'} \alpha(i,k,B,P',R) \cdot \alpha(k,j,C,Q,P') \\
+I_{case1} \quad \cdot \quad & \beta(i,k,B,R) \cdot \alpha(k,j,C,Q,B) \\
\boxed{+I_{case2}} \quad \cdot \quad & \sum_{P'} \alpha(i,k,B,P',R) \cdot \beta(k,j,C,P') \\
\boxed{+I_{case3}} \quad \cdot \quad & \beta(i,k,B,R) \cdot \beta(k,j,C,B) \\
\}
\end{aligned}
$$

The Inference Algorithm

- The inference process is to find a group of (t, y, s) for a sentence x to maximize the conditional probability.

$$(t, y, s)^* = \arg\max_{t,y,s} P(t, y, s | x; \theta)$$

- By substituting the ***sum*** function to the ***maximizing*** function, the DP algorithm for calculating $Z_{x^{(k)}}$ can also be utilized for the ***inference*** algorithm.

Time Complexity of the DP Algorithm

The complexity of the training and inference algorithm is

$$O(n \cdot L \cdot q^2 \cdot 4^h \cdot |U|)\,,$$

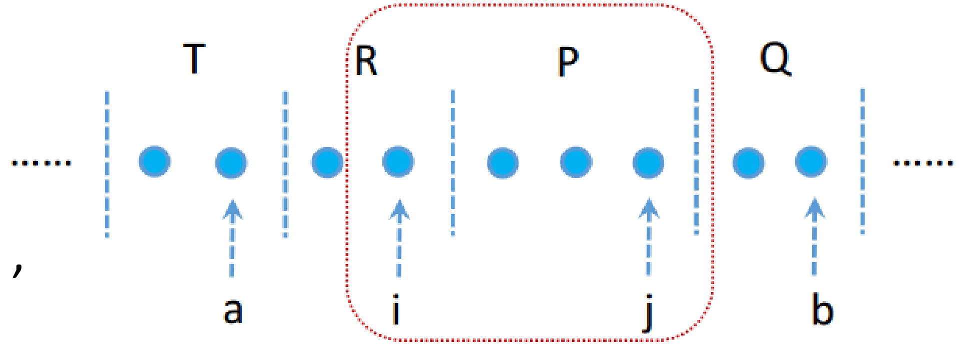
where

- n is the number of characters in the sentence
- L is the maximum segment 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

The Pruning Algorithm

- Atomic segment $c(i, j, P, R)$ in the DP algorithm

  An ***atomic segment*** is defined by the ***segment index*** i and j , as well as ***its label*** P and ***the previous label*** R



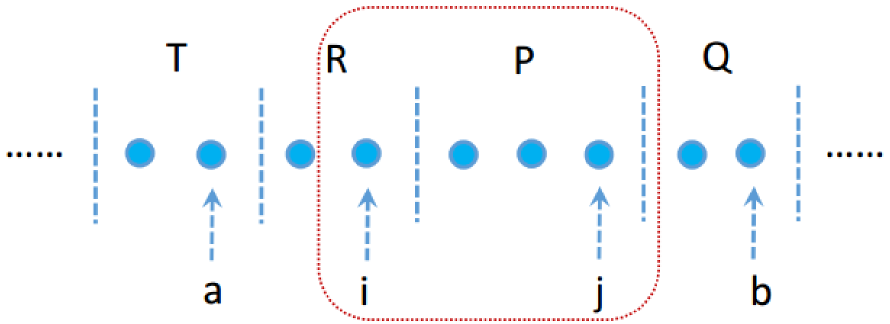
- Pruning the search space by reducing the number of atomic segments

  - Unlikely atomic segments are removed according to the ***max marginal*** of the segments.
  - In practice, $n \approx 40$, $L \approx 10$, and $q \approx 30$.
  - m is set to 0.001.

	# Atomic Segment
Before Pruning	$n \cdot L \cdot q^2 \approx 360{,}000$
After Pruning	$m \cdot n \cdot L \cdot q^2 \approx 360$

The Pruning Algorithm

- Calculating max marginal for atomic segments



The maximum energy potential from the left to i is calculated as

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

Similarly, the maximum energy potential from the right to j is calculated as

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

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

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

The Pruning Algorithm

- Properties of the pruning algorithm

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* $c_j \in \{c^{(1)}, c^{(2)}, ..., c^{(k)}\}$.

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* $v(s) > v(s^{(i)})$.

Please refer to our paper for the detailed proof.

Index

- Chinese Named Entity Recognition
- Motivation & Challenges
- The Proposed Joint Neural CRF Model
- Algorithms for the Joint Model
- Experimental Verifications
- Conclusions

Experimental Settings

- The Dataset
  - OntoNotes 4.0
  - The unique dataset that has all the labels of word segmentation, POS tagging, NER, and parsing
  - Splitting as previous work

Statistics	Train	Dev	Test
Sentence	15.7k	4.3k	4.3k
Char	491.9k	200.5k	208.1k

- The Metric
  - Precision/Recall
  - F1-measure
  - Recall & F1-measure on OOV entities

$$Precision = \frac{\#true\ positive}{\#trup\ positive + \#false\ positive}$$

$$Recall = \frac{\#true\ positive}{\#trup\ positive + \#false\ negative}$$

$$F1 = \frac{2 \cdot recall \cdot precision}{recall + precision}$$

Overall Performance

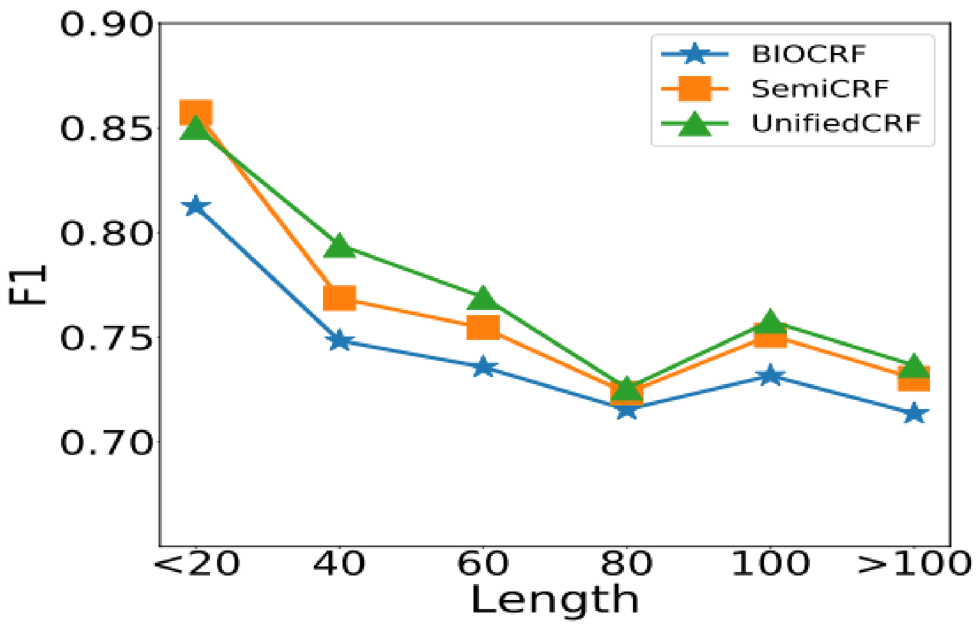
- Our proposed UnifiedCRF outperforms previous character-based methods by 2.79 points in F1 (from 73.88% to 76.67%).

- The improvement comes from:
  1. The exploration of more neural features for Semi-CRF.
  2. The joint model.

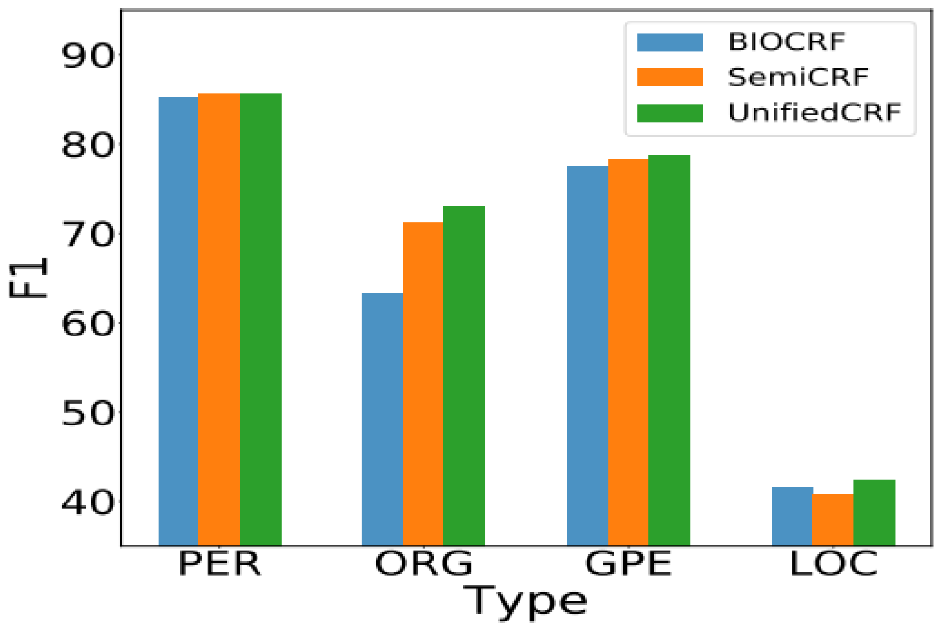
Models	**P**	**R**	**F1**	**O-R**	**O-F1**
Yang16a (gw)	65.59	71.84	68.57	–	–
Yang16b (gw)	72.98	**80.15**	76.40		
Che13(gw)	77.71	72.51	75.02		
Wang13(gw)	76.43	72.32	74.32		
Zhang18(gw)	**78.62**	73.13	75.77		
Zhang18(aw)	73.36	70.12	71.70		
Zhang18CRF	68.79	60.35	64.30	44.55	54.08
Zhang18Latt	76.35	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**

Fine-grained Performance

Performance with different sentence length



Performance with different entity type

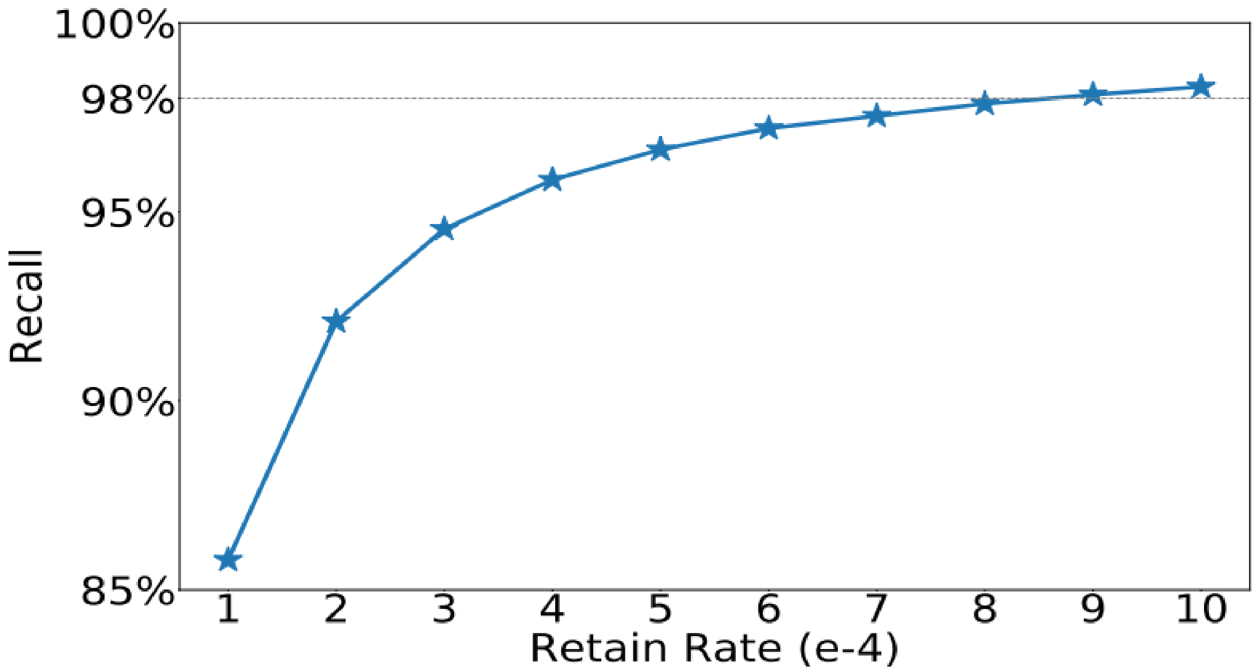


Performance for Other Tasks

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

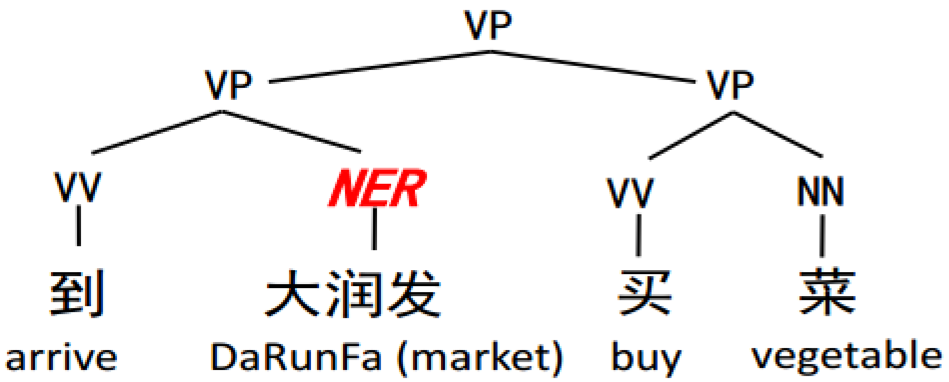
- It is observed that joint model also achieves improvements on word segmentation and POS tagging.
- “Parsing” denotes the performances for height-limited constituent parsing, and “Parsing Struct.” denotes the parsing structures without labels

Pruning Performance



- The recall of ground truth with different retaining rate.
- The retaining rate is set to be 0.001, with the loss of ground truth data being less than 2%.

Case Study



- A case where the NER can be improved with the help of grammar rules.
- “DaRunFa” is a market name, which is an OOV entity.
- In both BIO and semi-CRF, the entity “DaRunFa” cannot be recognized.
- With our proposed model, it can be successfully labeled.

Index

- Chinese Named Entity Recognition
- Motivation & Challenges
- The Proposed Joint Neural CRF Model
- Algorithms for the Joint Model
- Experimental Verifications
- Conclusions

Conclusions

- We have investigated the problem of utilizing a joint model of NER and parsing, to promote the performance of Chinese NER.
- The parsing task is reformulated to height-limited 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 have demonstrated that the proposed unified model outperforms previous methods by 2.79 point in the F1-measure.