PCFGs Can Do Better: Inducing Probabilistic Context-Free Grammars with Many Symbols

Songlin Yang[1], Yanpeng Zhao[2], Kewei Tu[1]

[1]School of Information Science and Technology, ShanghaiTech University

[2]ILCC, University of Edinburgh

Code

Paper

Introduction

Neural PCFGs have been proved effective in unsupervised constituency parsing. However, due to the high complexity of the inside algorithm, they cannot scale to a relatively large number of grammar symbols.
We use CP decomposition to decompose the binary rule probability tensor to decrease the complexity from cubic to at most quadratic in the number of grammar symbols. We enforce row/column normalization on the decomposed matrices to ensure the validity of the binary rule probability.

CP decomposition on PCFGs

Start rule:

$$S \to A \quad A \in \mathcal{N}$$

$$\mathrm{r}_{h_A} = \pi_{S\to A}\,, \quad \mathrm{r} \in \mathbb{R}^n$$

Binary rule:

$$A \to BC, \quad A \in \mathcal{N}, \quad B, C \in \mathcal{N} \cup \mathcal{P}$$

$$\mathrm{T}_{h_A, h_B, h_C} = \pi_{A\to BC}\,, \quad \mathrm{T} \in \mathbb{R}^{n\times m\times m}$$

Unary rule:

$$T \to w, \quad T \in \mathcal{P}, w \in \Sigma$$

$$\mathrm{Q}_{h_T, h_w} = \pi_{T\to w}\,, \quad \mathrm{Q} \in \mathbb{R}^{p\times q}$$

Recursive form of the inside algorithm

$$s^A_{i,j} = \sum_{k=i}^{j-1} \sum_{B,C} \pi_{A\to BC} \cdot s^B_{i,k} \cdot s^C_{k+1,j}\,. \tag{1}$$

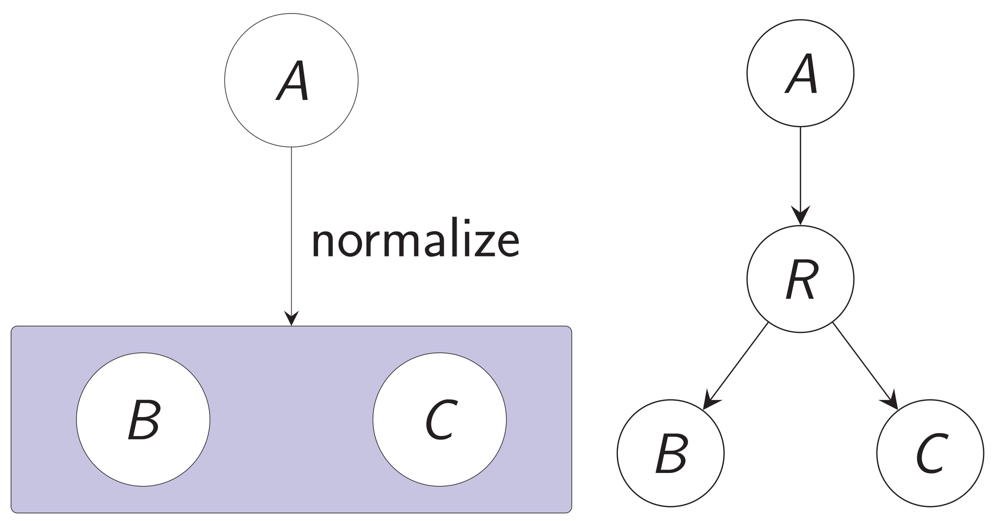
Kruskal form of the binary rule probability tensor:

$$\mathrm{T} = \sum_{l=1}^{r} \mathrm{T}^{(l)}, \quad \mathrm{T}^{(l)} = \mathrm{u}^{(l)} \otimes \mathrm{v}^{(l)} \otimes \mathrm{w}^{(l)}\,,$$

Eq. 1 becomes the following with complexity $O(l^3 m + l^2 mr)$:

$$\mathrm{s}_{i,j} = \mathrm{U} \cdot \sum_{k=i}^{j-1} \left((\mathrm{V}^T \mathrm{s}_{i,k}) \odot (\mathrm{W}^T \mathrm{s}_{k+1,j}) \right)$$

CP decomposition in probabilistic perspective



$$p(A \to B, C) = \sum_R p(A \to R) p(R \to B) p(R \to C)$$

$$p(A \to B, C) = \sum_{l=1}^{r} \mathrm{u}^{(l)} \otimes \mathrm{v}^{(l)} \otimes \mathrm{w}^{(l)}$$

The newly introduced dimension in CP decomposition can be regarded as a latent variable R. $\|R\|$ is tensor rank.

Left: normalize $A \to B, C$, total complexity $O(m^3)$. Right: normalize $A \to R, R \to B, R \to C$, total complexity: $O(mr)$.
Row normalization of U is equivalent to normalize $p(A \to R)$,
Column normalization of V, W is equivalent to normalize $p(R \to B), p(R \to C)$ respectively.

Neural Parameterization

Use distributed representation for each symbol. Use neural networks to calculate grammar rule probabilities.
Use Softmax function to ensure the row/column normalization of U, V, W.
Without neural parameterization, performance drops significantly (even underperforms right-branching baseline in WSJ)
Activation functions other than ReLU perform much worse.

Experiments

Comparison on WSJ test set.

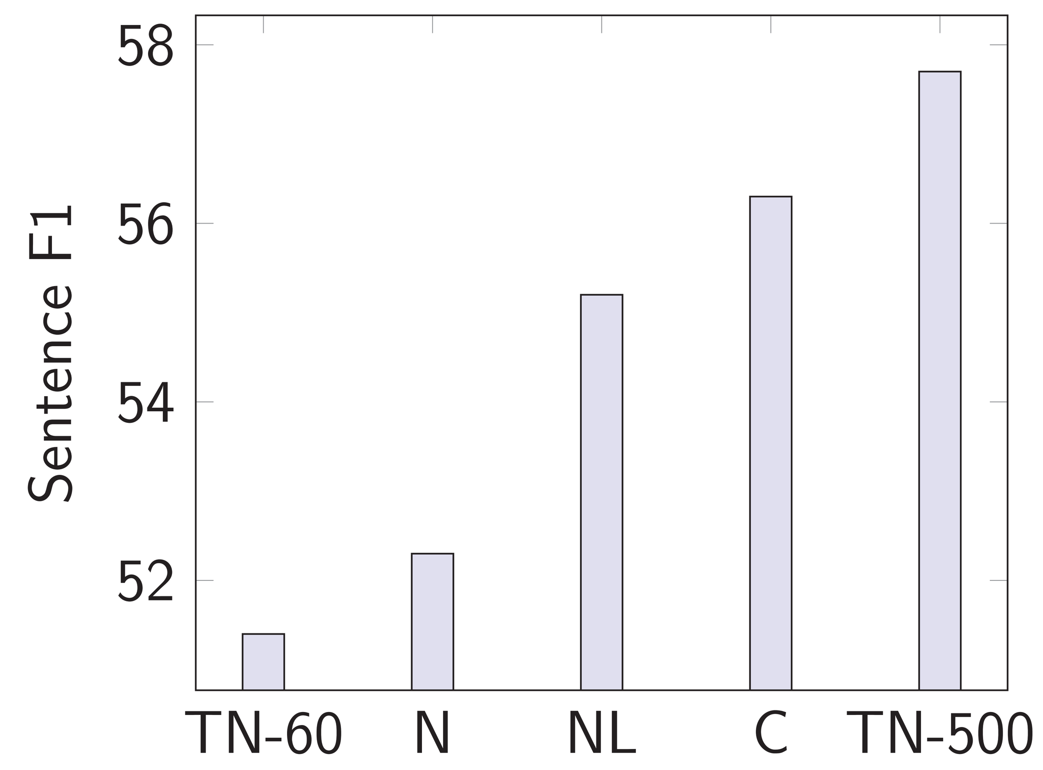


Figure 3. We omit the PCFG suffix. TN: our proposed method. -X: X perterminals. N: Neural. NL: Neural Lexicalized. C: Compound

Influence of symbol number

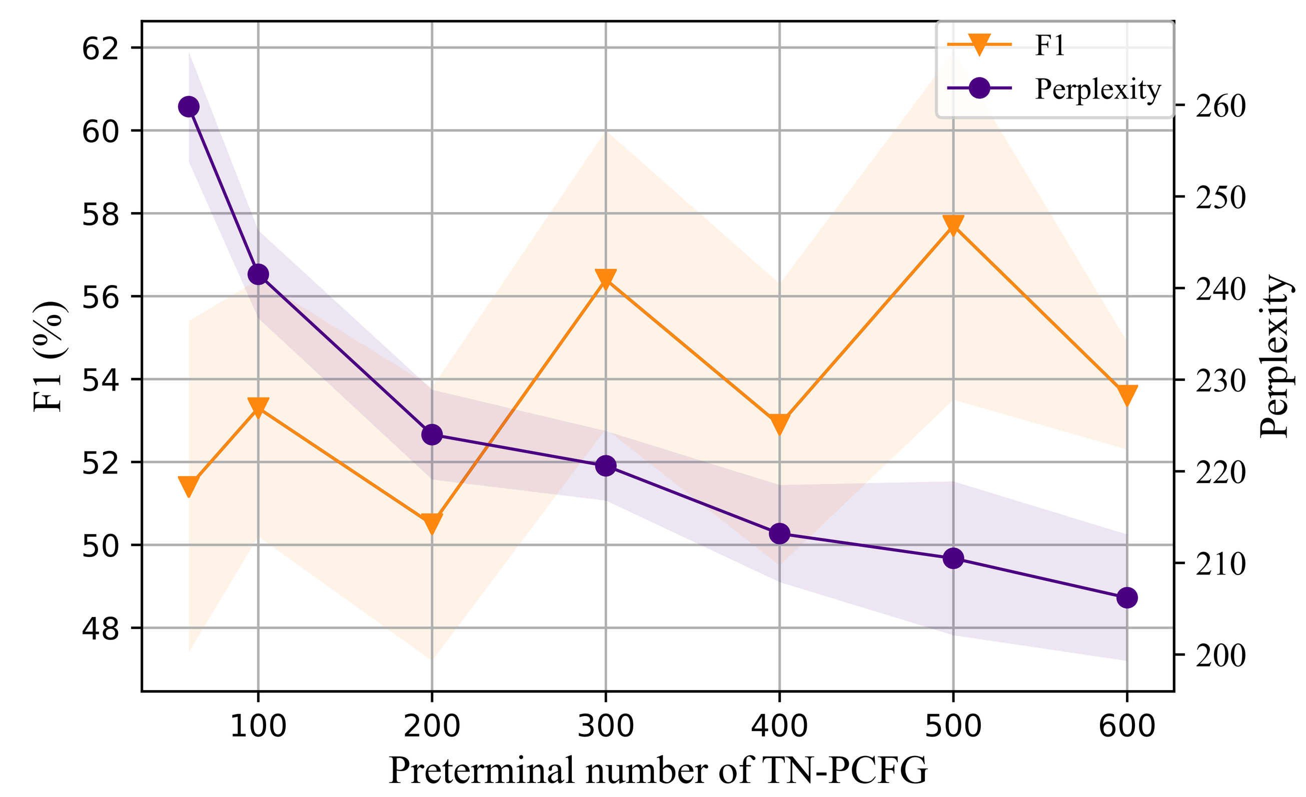


Figure 4. We set the ratio of nonterminals: perterminals as 1:2.

Multilingual experiments

Model	Chinese	Basque	German	French	Hebrew	Hungarian	Korean	Polish	Swedish	Mean
Left Branching†	7.2	17.9	10.0	5.7	8.5	13.3	18.5	10.9	8.4	11.2
Right Branching†	25.5	15.4	14.7	26.4	30.0	12.7	19.2	34.2	30.4	23.2
Random Trees†	15.2	19.5	13.9	16.2	19.7	14.1	22.2	21.4	16.4	17.6
N-PCFG w/ MBR	$26.3_{\pm 2.5}$	$35.1_{\pm 2.0}$	$42.3_{\pm 1.6}$	$\mathbf{45.0}_{\pm 2.0}$	$\mathbf{45.7}_{\pm 2.2}$	$43.5_{\pm 1.2}$	$28.4_{\pm 6.5}$	$43.2_{\pm 0.8}$	$17.0_{\pm 9.9}$	36.3
C-PCFG w/ MBR	$38.7_{\pm 6.6}$	$\mathbf{36.0}_{\pm 1.2}$	$43.5_{\pm 1.2}$	$\mathbf{45.0}_{\pm 1.1}$	$45.2_{\pm 0.5}$	$\mathbf{44.9}_{\pm 1.5}$	$30.5_{\pm 4.2}$	$43.8_{\pm 1.3}$	$33.0_{\pm 15.4}$	40.1
TN-PCFG $p = 500$	$\mathbf{39.2}_{\pm 5.0}$	$\mathbf{36.0}_{\pm 3.0}$	$\mathbf{47.1}_{\pm 1.7}$	$39.1_{\pm 4.1}$	$39.2_{\pm 10.7}$	$43.1_{\pm 1.1}$	$\mathbf{35.4}_{\pm 2.8}$	$\mathbf{48.6}_{\pm 3.1}$	$\mathbf{40.0}_{\pm 4.8}$	**40.9**