Neural Bi-Lexicalized PCFG Induction

Songlin Yang, Yanpeng Zhao, Kewei Tu

June 24, 2021

ShanghaiTech University, University of Edinburgh

Lexicalized PCFGs

PCFGs:

$$
\begin{aligned}
&S \to A && A \in \mathcal{N} \\
&A \to BC && A \in \mathcal{N}, \quad B, C \in \mathcal{N} \cup \mathcal{P} \\
&T \to w && T \in \mathcal{P}, w \in \Sigma
\end{aligned}
$$

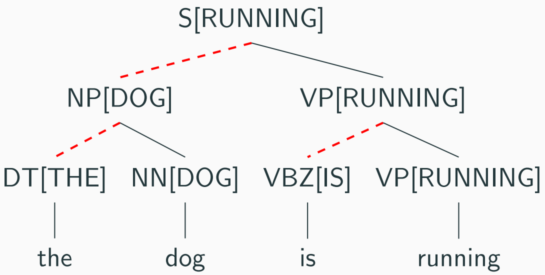
Lexicalized PCFGs:

$$
\begin{aligned}
&S \to A[w_p] && A \in \mathcal{N} \\
&A[w_p] \to B[w_p]C[w_q], && A \in \mathcal{N}; B, C \in \mathcal{N} \cup \mathcal{P} \\
&A[w_p] \to C[w_q]B[w_p], && A \in \mathcal{N}; B, C \in \mathcal{N} \cup \mathcal{P} \\
&T[w_p] \to w_p, && T \in \mathcal{P}
\end{aligned}
$$

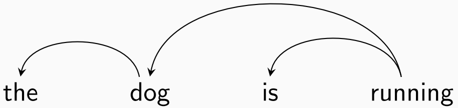
Lexicalized PCFGs extend PCFGs by associating a word, i.e., the lexical head, with each grammar symbol.

Terminal words are generated in the binary rules instead of the unary rules, so the unary rules in lexicalized PCFGs are deterministic.

Lexicalized phrase-structure tree



Binary constituency tree and projective dependency tree can be generated together by lexicalized PCFGs. Dashed line indicates dependency arcs.



Lexicalized PCFG Induction

Goal: learn the grammar rule probabilities of a lexicalized PCFG from corpus alone.

Learning objective: marginal sentence log-likelihood, which can be estimated by the inside algorithm.

Problem

Lexicalized PCFGs suffer from representation and learning/inference complexities.

- representation: too many learnable parameters. $\mathcal{O}\left(m^3|\Sigma|^2\right)$
- learning/inference: relatively slow. $\mathcal{O}(l^4 m^3)$

where

- m: nonterminal number
- l: sentence length
- $|\Sigma|$: vocabulary size

Previous solution

(Zhu et al, 2020) present neural lexicalized PCFG, combining the idea of

factorizing the binary rule probability and neural pamameterization

Previous solution

(Zhu et al, 2020) present neural lexicalized PCFG, combining the idea of

factorizing the binary rule probability and neural pamameterization

$$
\begin{aligned}
& p\left(A\left[w_p\right] \rightarrow B\left[w_p\right] C\left[w_q\right]\right) \\
& = p\left(B, \curvearrowleft, C \mid A, w_p\right) p\left(w_q \mid C\right) \\
& = p\left(B, \curvearrowleft \mid A, w_p\right) \\
& p(C \mid A, B, \curvearrowleft, w_p) p(w_q \mid C)
\end{aligned}
$$

Previous solution

(Zhu et al, 2020) present neural lexicalized PCFG, combining the idea of

factorizing the binary rule probability and neural pamameterization

$$\uparrow$$

$$p\left(A\left[w_p\right] \rightarrow B\left[w_p\right] C\left[w_q\right]\right)$$
$$= p\left(B, \curvearrowleft, C \mid A, w_p\right) p\left(w_q \mid C\right)$$
$$= p\left(B, \curvearrowleft \mid A, w_p\right)$$
$$p(C \mid A, B, \curvearrowleft, w_p) p(w_q \mid C)$$

☺ manage to decrease the inference complexity.
☹ given C, child word w_q is independent from the parent word w_p, thus bilexical dependencies are ignored.

Previous solution

(Zhu et al, 2020) present neural lexicalized PCFG, combining the idea of

factorizing the binary rule probability ↑ and neural pamameterization ↑

$$p\left(A\left[w_p\right] \rightarrow B\left[w_p\right] C\left[w_q\right]\right)$$
$$= p\left(B, \curvearrowleft, C \mid A, w_p\right) p\left(w_q \mid C\right)$$
$$= p\left(B, \curvearrowleft \mid A, w_p\right)$$
$$p(C \mid A, B, \curvearrowleft, w_p) p(w_q \mid C)$$

☺ manage to decrease the inference complexity.
☹ given C, child word w_q is independent from the parent word w_p, thus bilexical dependencies are ignored.

☺ decrease the total learnable parameters; facilitate informed smoothing; boost the unsupervised parsing performance.

Desiderata

Can we avoid making additional independence assumptions (i.e., bilexicalized dependencies are properly modeled.) and meanwhile decrease representation and learning/inference complexities?

Latent-variable based factorization

$$p\,(B, C, W_q, D \mid A, W_p) = \sum_{H} p\,(H \mid A, W_p)\, p(B \mid H) p(C, D \mid H) p\,(W_q \mid H)$$

According to d-separation, when A and w_p are given, B, C, w_q, and D are interdependent due to the existence of H, so this parameterization does not make any independence assumption beyond the original binary rule.

Latent-variable based factorization

$$p(B, C, W_q, D \mid A, W_p) = \sum_H p(H \mid A, W_p)\, p(B \mid H) p(C, D \mid H) p(W_q \mid H)$$

According to d-separation, when A and w_p are given, B, C, w_q, and D are interdependent due to the existence of H, so this parameterization does not make any independence assumption beyond the original binary rule.

- Similar to the CP decomposition (a.k.a. tensor rank decomposition).

Latent-variable based factorization

$$p\,(B, C, W_q, D \mid A, W_p) =$$
$$\sum_{H} p\,(H \mid A, W_p)\, p(B \mid H) p(C, D \mid H) p\,(W_q \mid H)$$

According to d-separation, when A and w_p are given, B, C, w_q, and D are interdependent due to the existence of H, so this parameterization does not make any independence assumption beyond the original binary rule.

- Similar to the CP decomposition (a.k.a. tensor rank decomposition).
- The domain size of H (i.e. $|H|$) can be regarded as the tensor rank.

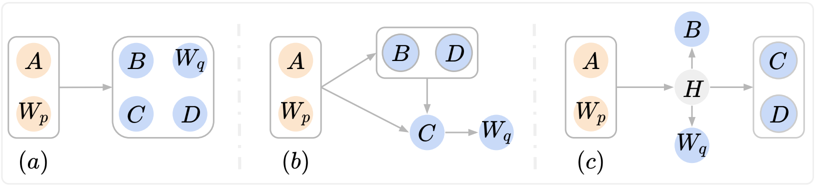
Latent-variable based factorization

$$p(B, C, W_q, D \mid A, W_p) = \sum_H p(H \mid A, W_p)\, p(B \mid H) p(C, D \mid H) p(W_q \mid H)$$

According to d-separation, when A and w_p are given, B, C, w_q, and D are interdependent due to the existence of H, so this parameterization does not make any independence assumption beyond the original binary rule.

- Similar to the CP decomposition (a.k.a. tensor rank decomposition).
- The domain size of H (i.e. $|H|$) can be regarded as the tensor rank.
- The time complexity of the inside algorithm can then be reduced by using the refold-unfold transformation (Eisner and Blatz, 2007) $\rightarrow O\left(l^4|H| + l^2 m|H|\right)$.

Comparison

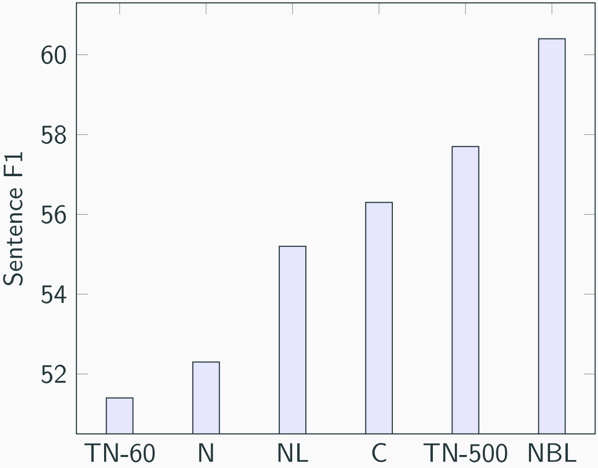


(a): original lexicalized PCFGs.

(b): (Zhu et al, 2020).

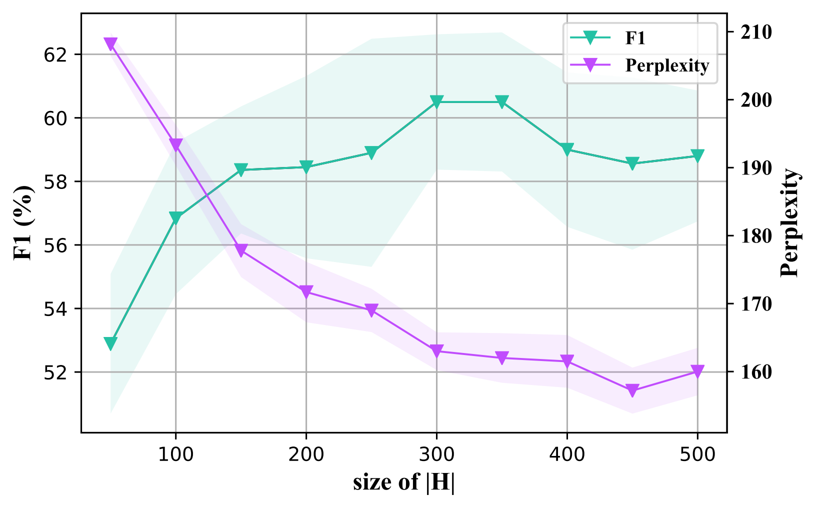
(c): ours.

Comparison on WSJ test set



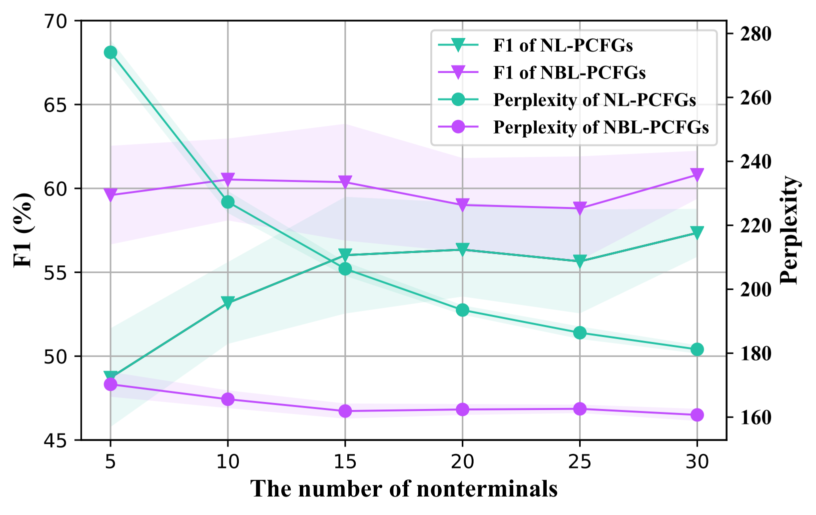
N: [1]; C: [1]; NL: [3]; TN: [2]; NBL: ours.

Influence of the domain size of H



The domain size $|H|$ is analogous to the tensor rank and thus influences the expressiveness of the model.

Influence of the number of nonterminals



NBL-PCFGs are less sensitive to the number of nonterminals as we explicitly model bilexical dependencies.

Influence of different variable bindings

	F1	UDAS	UUAS	Perplexity
D-C	**60.4**	39.1	56.1	**161.9**
D-alone	57.2	32.8	54.1	164.8
D-w_q	47.7	**45.7**	**58.6**	176.8
D-B	47.8	36.9	54.0	169.6

Table 3: Binding the head direction D with different variables.

- D-alone: D is generated alone.
- D-w_q: D is generated with w_q.
- D-B: D is generated with head-child B.
- D-C: D is generated with non-head-child C.

The way to bind head direction has a great impact on the parsing performance.

Summary

- We have presented NBL-PCFGs, which combine tensor rank decomposition and refold-unfold transformation technique to decrease the representation, learning and inference complexities and meanwhile model bilexical dependencies.
- Experiments on WSJ show the effectiveness of modeling bilexical dependencies in increasing unsupervised parsing performance and decreasing perplexities.

Code

Our code is publicly available at:

https://github.com/sustcsonglin/TN-PCFG

Questions?

References i

Y. Kim, C. Dyer, and A. Rush.
Compound probabilistic context-free grammars for grammar induction.
In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy, July 2019. Association for Computational Linguistics.

S. Yang, Y. Zhao, and K. Tu.
PCFGs can do better: Inducing probabilistic context-free grammars with many symbols.
In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1487–1498, Online, June 2021. Association for Computational Linguistics.

H. Zhu, Y. Bisk, and G. Neubig.
The return of lexical dependencies: Neural lexicalized PCFGs.
Transactions of the Association for Computational Linguistics, 8:647–661, 2020.