Neural Bi-Lexicalized PCFG Induction

Songlin Yang, Yanpeng Zhao, Kawei Tu
June 24, 2021
ShanghaiTech University, University of Edinburgh
Lexicalized PCFGs

PCFGs:

\[
S \rightarrow A \\
A \rightarrow BC \\
T \rightarrow w
\]

\[A, B, C \in \mathcal{N} \cup \mathcal{P} \]

\[T \in \mathcal{P}, w \in \Sigma\]

Lexicalized PCFGs:

\[
S \rightarrow A[w_p] \\
A[w_p] \rightarrow B[w_p]C[w_q], \quad A \in \mathcal{N}; \quad B, C \in \mathcal{N} \cup \mathcal{P} \\
A[w_p] \rightarrow C[w_q]B[w_p], \quad A \in \mathcal{N}; \quad B, C \in \mathcal{N} \cup \mathcal{P} \\
T[w_p] \rightarrow w_p, \quad T \in \mathcal{P}
\]

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.
Binary constituency tree and projective dependency tree can be generated together by lexicalized PCFGs. Dashed line indicates dependency arcs.
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.
Lexicalized PCFGs suffer from representation and learning/inference complexities.

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

where

- $m$: nonterminal number
- $l$: sentence length
- $|\Sigma|$: vocabulary size
(Zhu et al, 2020) present neural lexicalized PCFG, combining the idea of
factorizing the binary rule probability and neural parameterization
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.
(Zhu et al, 2020) present neural lexicalized PCFG, combining the idea of factorizing the binary rule probability and neural parameterization.

\[
p(A[w_p] \rightarrow B[w_p] C[w_q]) = p(B, \odot, C | A, w_p) p(w_q | C) = p(B, \odot | A, w_p) \quad p(C | A, B, \odot, w_p) p(w_q | C)
\]
(Zhu et al, 2020) present neural lexicalized PCFG, combining the idea of factorizing the binary rule probability and neural parameterization:

\[
p(A[w_p] \rightarrow B[w_p] C[w_q])
= p(B, \bowtie, C | A, w_p) p(w_q | C)
= p(B, \bowtie | A, w_p)
\]

\[
p(C | A, B, \bowtie, w_p)p(w_q | C)
\]

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

\[
p (A [w_p] \rightarrow B [w_p] C [w_q]) = p (B, \circledast, C | A, w_p) p (w_q | C) = p (B, \circledast | A, w_p) p(C | A, B, \circledast, w_p) p(w_q | C)
\]

- 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.
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

\[ p(B, C, W_q, D | A, W_p) = \]
\[ \sum_H p(H | A, W_p) p(B | H) p(C, D | H) p(W_q | H) \]

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)
  \[ \rightarrow O(l^4|H| + l^2 m|H|). \]
Comparison

(a): original lexicalized PCFGs.
(b): (Zhu et al, 2020).
(c): ours.
Comparison on WSJ test set

Sentence F1

<table>
<thead>
<tr>
<th></th>
<th>N: [1]; C: [1]; NL: [3]; TN: [2]; NBL: ours.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TN-60</td>
<td>52</td>
</tr>
<tr>
<td>N</td>
<td>54</td>
</tr>
<tr>
<td>NL</td>
<td>56</td>
</tr>
<tr>
<td>C</td>
<td>58</td>
</tr>
<tr>
<td>TN-500</td>
<td>60</td>
</tr>
<tr>
<td>NBL</td>
<td>60</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>UDAS</th>
<th>UUAS</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D-C$</td>
<td>60.4</td>
<td>39.1</td>
<td>56.1</td>
<td>161.9</td>
</tr>
<tr>
<td>$D$-alone</td>
<td>57.2</td>
<td>32.8</td>
<td>54.1</td>
<td>164.8</td>
</tr>
<tr>
<td>$D-w_q$</td>
<td>47.7</td>
<td>45.7</td>
<td>58.6</td>
<td>176.8</td>
</tr>
<tr>
<td>$D-B$</td>
<td>47.8</td>
<td>36.9</td>
<td>54.0</td>
<td>169.6</td>
</tr>
</tbody>
</table>

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.
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.
Our code is publicly available at:

https://github.com/sustcsonglin/TN-PCFG
Questions?
Y. Kim, C. Dyer, and A. Rush.  
**Compound probabilistic context-free grammars for grammar induction.**  

S. Yang, Y. Zhao, and K. Tu.  
**PCFGs can do better: Inducing probabilistic context-free grammars with many symbols.**  
H. Zhu, Y. Bisk, and G. Neubig.

The return of lexical dependencies: Neural lexicalized PCFGs.