A Bayesian Model of Syntax-Directed Tree to String Grammar Induction
Trevor Cohn and Phil Blunsom

Presented by: Jeff Flanigan
The Problem

• Lots of heuristics for grammar extraction
• Word alignments could be better
• EM for extracting rules fails
  – See “Why Generative Phrase Models Underperform Surface Heuristics” DeNero et al. 2006 (VERY good read!)
The Problem

- Lots of heuristics for grammar extraction
- Word alignments could be better
- EM for extracting rules fails
  - See “Why Generative Phrase Models Underperform Surface Heuristics” DeNero et al. 2006 (VERY good read!)

Learn rules directly

Use Bayesian Methods
The Idea

Learn alignments between target trees and source spans directly

Each node in tree has a latent variable (the alignment)
Nodes can be unaligned
The Idea

One rule extracted for each aligned node
Alignment span ⇔ Rule

Use Bayesian Nonparametrics to Prevent Degeneracy
Bayesian Learning

\[ p(\theta \mid D, \alpha) = \frac{p(D \mid \theta)p(\theta \mid \alpha)}{p(D \mid \alpha)} \propto p(D \mid \theta)p(\theta \mid \alpha) \]

In Bayesian grammar induction, \( \theta \) is distribution over grammars (usually write \( G \) instead of \( \theta \)). Learning a grammar is a “draw” from \( G \).
Bayesian Learning

\[ p(\theta | D, \alpha) = \frac{p(D | \theta)p(\theta | \alpha)}{p(D | \alpha)} \propto p(D | \theta)p(\theta | \alpha) \]

In Bayesian grammar induction, \( \theta \) is distribution over grammars (usually write G instead of \( \theta \))
Learning a grammar is a “draw” from G

Use Gibbs Sampling to get a draw (i.e. grammar)
Gibbs Sampling

- The GS algorithm:
  1. Suppose the graphical model contains variables $x_1, \ldots, x_n$
  2. Initialize starting values for $x_1, \ldots, x_n$
  3. Do until convergence:
     1. Pick an ordering of the $n$ variables (can be fixed or random)
     2. For each variable $x_i$ in order:
        1. Sample $x$ from $P(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, i.e. the conditional distribution of $x_i$ given the current values of all other variables
        2. Update $x_i \leftarrow x$

- When we update $x_i$, we immediately use its new value for sampling other variables $x_j$
Bayesian Learning

Need: prior over the space of grammars

\[ p(\theta \mid D, \alpha) = \frac{p(D \mid \theta)p(\theta \mid \alpha)}{p(D \mid \alpha)} \propto p(D \mid \theta)p(\theta \mid \alpha) \]

Use Gibbs Sampling
Prior over Grammars

Constituent $c$
Rewrite $c$ using rule $r$ with probability:

$$r | c \sim G_c$$

$G_c$ is a distribution over grammars, drawn from a Dirichlet Process:

$$G_c | \alpha_c, P_0 \sim DP(\alpha_c, P_0(\cdot | c))$$
Dirichlet Process

Draws from $\text{DP}(\alpha, H(\Omega))$ are distributions (i.e. $\geq 0$, sum to 1)

$H$ is the base distribution over $\Omega$

$\alpha$ is the concentration parameter – determines sparsity

Base distribution
A single draw from the DP

In our case $\Omega =$ space of rules

Picture source: http://www.cs.cmu.edu/~epxing/Class/10708/lecture/lecture24-DP.pdf
Base Distribution $H(\Omega)$

Base distribution probability of rewriting $c$ with RHS $r$

$$P_0(r|c) = P_0(e,w|c) = P(e|c) P(w|e)$$

$e =$ elementry tree

$w =$ words in source

$P(e|c)$

Expand $c$ recursively
Number of child nodes $\sim \text{Geom}(p_{\text{child}})$
Pre-terminals have one child
Draw non-terminals and terminals uniformly from $N$ and $T$

$P(w|e)$

Number of terminals $\sim \text{Geom}(p_{\text{term}})$
Draw source terminals (i.e. source phrases) uniformly from possible phrases
Arrange variables, source phrases randomly

Example: $\langle (\text{NP} \text{ NP}_1 (\text{PP} (\text{IN} \text{ of}) \text{ NP}_2)), \text{ 2 的 1} \rangle$
The Gibbs Sampler

Visit sentences/nodes randomly, and resample a rule

Resample using $P(r|\text{everything else})$
The Gibbs Sampler

Dirichlet Process Prior

\[ r|c \sim G_c \]
\[ G_c|\alpha_c, P_0 \sim \text{DP}(\alpha_c, P_0(\cdot|c)) \]
The Gibbs Sampler

Dirichlet Process Prior

\[ r \mid c \sim G_c \]
\[ G_c \mid \alpha_c, P_0 \sim \text{DP}(\alpha_c, P_0(\cdot \mid c)) \]

Integrate out \( G_c \) (see Neal 2000 “Markov Chain Sampling Methods for Dirichlet Process Mixture Models”)
The Gibbs Sampler

Dirichlet Process Prior

\[ r | c \sim G_c \]
\[ G_c | \alpha_c, P_0 \sim DP(\alpha_c, P_0(\cdot | c)) \]

Integrate out \( G_c \) (see Neal 2000 “Markov Chain Sampling Methods for Dirichlet Process Mixture Models”)

Gibbs Sampler is simple!

\[
p(r_i | r^{-i}, c, \alpha_c, P_0) = \frac{n_{r_i}^{-i} + \alpha_c P_0(r_i | c)}{n_c^{-i} + \alpha_c}
\]

\[ n_{r_i}^{-i} = \# \text{ times rule } r_i \text{ used everywhere else} \]
\[ n_c^{-i} = \# \text{ times } c \text{ used everywhere else} \]
The Gibbs Sampler

Dirichlet Process Prior

\[ r|c \sim G_c \]
\[ G_c|\alpha_c, P_0 \sim DP(\alpha_c, P_0(\cdot|c)) \]

Integrate out \( G_c \) (see Neal 2000 “Markov Chain Sampling Methods for Dirichlet Process Mixture Models”)

Gibbs Sampler is simple!

\[ p(r_i|r^{-i}, c, \alpha_c, P_0) = \frac{n_{r_i}^{-i} + \alpha_c P_0(r_i|c)}{n_{c}^{-i} + \alpha_c} \]

\( n_{r_i}^{-i} = \# \) times rule \( r_i \) used everywhere else

\( n_{c}^{-i} = \# \) times \( c \) used everywhere else

Small \( \alpha \) means do what’s popular

“Rich get richer”
Resampling Operators

• 1\textsuperscript{st} EXPAND
  – Resample the alignment subject to constrains
  – Can’t go outside closest aligned parent
  – Must include descendants
  – Can’t overlap siblings
Resampling Operators

- **2\textsuperscript{nd} SWAP**
  - Swap alignments of two nodes
  - Only allowed for nodes with unaligned descendants
Summary

\[ p(\theta | D, \alpha) = \frac{p(D | \theta) p(\theta | \alpha)}{p(D | \alpha)} \propto p(D | \theta) p(\theta | \alpha) \]

Resample using \( p(r | \text{everything else}) \)
Training Set

- FBIS + 100k Sinorama

<table>
<thead>
<tr>
<th></th>
<th>English → Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sentences</strong></td>
<td>300k</td>
</tr>
<tr>
<td><strong>Words or Segments</strong></td>
<td>11.0M 8.6M</td>
</tr>
<tr>
<td><strong>Avg. Sent. Length</strong></td>
<td>36 28</td>
</tr>
<tr>
<td><strong>Longest Sent.</strong></td>
<td>80 80</td>
</tr>
</tbody>
</table>
Training

• $\alpha = 10^6$
• $p_{\text{child}} = p_{\text{expand}} = p_{\text{term}} = .5$
• 300 iterations of Gibbs sampler
• $\frac{1}{2}$ hour per iteration on single core 2.3 Ghz
Extracted Grammar

- Number of rules
- Maximum tree depth
- Variables
- Source terminals
- Target terminals
Extracted Grammar

Top 10 Rules not in GHKM

1. ((TOP (S NP1 VP2)), 1 2 3)
2. ((S (VP (TO to) VP1)), 1)
3. ((NP NP1 (PP (IN of) NP2)), 2 1)
4. ((PP (IN in) NP1), 在 1)
5. ((NP NP1 (PP (IN of) NP2)), 1 2)
6. ((S (VP TO1 VP2)), 1 2)
7. ((VP (VBZ is) NP1), 是 1)
8. ((NP (NP (DT the) NN1) (PP (IN of) NP2)), 2 1)

Top 10 Rules not in New Model

1. ((PP (IN at) (NP DT1 (NNS levels))), 1 级)
2. ((NP NP1 ,2 NP3 (, ) CC4 NP5), 1 2 3 4 5)
3. ((NP NP1 ,2 NP3 ,4 NP5 (, ) CC and) NP6, 1 2 3 4 5, 6)
4. ((S S11 (NP (PRP They)) VP2,3), 1 2 3)
5. ((S PP1 ,2 NP3 VP4 ,5), 1 2 3 4 5 6)
6. ((S PP1 ,2 NP3 VP4 ,5), 1 中 2 3 4 5)
7. ((NP (NP Foreign) (NNP Ministry) NN2 (NNP Zhu) (NNP Bangzao)), 外交部 1 朱邦造)
8. ((S S11 S2), 1 2)
9. ((S S11 (NP (PRP We)) VP2 ,3), 1 2 3)
10. ((NP (DT the) (NNS people) POS1)), 人民 1)

Table 5: Top five rules which include the possessive particle and at least one variable.
BLEU Scores

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU score</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHKM</td>
<td>26.0</td>
</tr>
<tr>
<td>Our model</td>
<td>26.6</td>
</tr>
</tbody>
</table>

Table 6: Translation results on the NIST test set MT03 for sentences of length ≤ 20.
\[ p(\theta \mid D, \alpha) = \frac{p(D \mid \theta)p(\theta \mid \alpha)}{p(D \mid \alpha)} \propto p(D \mid \theta)p(\theta \mid \alpha) \]