Incremental Coarse-to-Fine Parsing

Marten van Schijndel
Department of Linguistics
The Ohio State University

April 20, 2012
Incremental Motivation

Understanding ... One Step at a Time

- Cognitive motivations
  - Operates on incomplete information (Cloze testing)

- Engineering motivations
  - Can make use of information about recent content/structure (coreference, pragmatics)
  - Unsegmented input
  - $O(n)$ Streaming task
Coarse-to-Fine Motivation

What is it?
Coarse-to-Fine Motivation

What is it?

- A way of improving parse speed/accuracy through pruning the search space.
Coarse-to-Fine Motivation

What is it?

- A way of improving parse speed/accuracy through pruning the search space.
- It has massively sped up parsers in the recent past
  - [Petrov and Klein, 2007] 50x
CTF Theory

How does it work?
CTF Theory

How does it work?

Parse in phases
[Charniak et al., 2006]
Some ways of implementing Coarse-to-Fine:

- Do it by hand [III and Kaplan, 1993, Charniak et al., 2006] or machine [Petrov and Klein, 2007]
- Single or Multi-layered
- If we assume the Berkeley Parser paradigm:
  - Trainer derives split-merge grammar files
  - Initialization phase creates a predictive chain back to coarse grammar
Sequence Model Parsing

Marten van Schijndel  Incremental Coarse-to-Fine Parsing
Sequence Model Training

Split-Merge Berkeley Grammar Trainer
[Petrov et al., 2006]

- Input: Boring tagged sentences
  \((S (ADVP \text{happily}) (NP-\text{SUBJ} \text{John})\ldots)\)
- EM classification performed over a given number of split-merge cycles
- Output: Sleek new PCFG
  \((S^g_{10} \rightarrow \text{ADVP}^g_{21} \text{NP}^g_{4} 1.462527E-18) \text{ WOW!}\)
Sequence Model Training

Split-Merge Berkeley Grammar Trainer
[Petrov et al., 2006]

- Input: Boring tagged sentences
  \( (S (ADVP \text{happily}) (NP-\text{SUBJ} John). . . ) \)
- EM classification performed over a given number of split-merge cycles
- Output: Sleek new PCFG
  \( (S^g_{10} \rightarrow ADVP^g_{21} NP^g_4 1.462527E-18) \) WOW!

Profit:
- Accuracy
Sequence Model Training

Split-Merge Berkeley Grammar Trainer
[Petrov et al., 2006]

- Input: Boring tagged sentences
  \( (S \text{(ADVP happily)} \text{(NP-SUBJ John). . . }) \)
- EM classification performed over a given number of split-merge cycles
- Output: Sleek new PCFG
  \( (S^g_{10} \rightarrow \text{ADVP}^g_{21} \text{NP}^g_{4} 1.462527E-18) \) WOW!

Profit:
- Accuracy

Cost:
- Training time
- Increased size of grammar
Sequence Model Training

Sequence Model Conversion

- Input: Sleek newly obtained PCFG
  \((S ^{g_{10}} \rightarrow ADVP ^{g_{21}} NP ^{g_{4}} 1.462527E-18)\)
- Generate virtual trees to give probabilities of component productions
- Output: Phase-, depth-specific grammar
  \((B 2 S ^{g_{10}} ADVP ^{g_{21}} \rightarrow NP ^{g_{4}} 2.348767E-20)\)
Sequence Model Training

Sequence Model Conversion

- Input: Sleek newly obtained PCFG
  \((S_{g_{10}} \rightarrow \text{ADVP}_{g_{21}} \text{NP}_{g_{4}} 1.462527E-18)\)
- Generate virtual trees to give probabilities of component productions
- Output: Phase-, depth-specific grammar
  \((B_2 S_{g_{10}} \text{ADVP}_{g_{21}} \rightarrow \text{NP}_{g_{4}} 2.348767E-20)\)

Profit:

- Information about upcoming categories
- Information about embedding depth
Sequence Model Training

Sequence Model Conversion

- **Input**: Sleek newly obtained PCFG
  
  \[(S^g_{10} \rightarrow \text{ADVP}^g_{21} \text{NP}^g_{4} 1.462527E-18)\]

- **Generate virtual trees to give probabilities of component productions**

- **Output**: Phase-, depth-specific grammar
  
  \[(B 2 S^g_{10} \text{ADVP}^g_{21} \rightarrow \text{NP}^g_{4} 2.348767E-20)\]

**Profit:**

- Information about upcoming categories
- Information about embedding depth

**Cost:**

- Training time
- Increased size of grammar
Mix it all up

How does this work?

- Approximate Inference

Variable Descriptions

- $q_t^d$ represents an element of working memory/incomplete constituent
- These are decomposed into $a_t^d$ and $b_t^d$
- $x_t$ is the observation at time $t$
- $p_t$ is the preterminal that expands into that observation
- $f_t$ is the final state obtained by integrating a new observation into the parse (expansion state)
Mix it all up

\[ q_t^d \]
Mix it all up

\[ q^d_t \]

\[
\begin{array}{c}
\begin{array}{c}
q_1^1 \\
q_2^1 \\
q_3^1 \\
q_1^2 \\
q_2^2 \\
q_3^2 \\
q_1^3 \\
q_2^3 \\
q_3^3 \\
q_1^4 \\
q_2^4 \\
q_3^4 \\
p_1 \\
p_2 \\
x_1 \\
x_2 \\
f_2 \\
f_3 \\
\end{array}
\end{array}
\]

\[ q_{12}^1 = NP/NN \]
\[ q_{13}^1 = S/VP \]
\[ q_{21}^2 = NP \]
\[ q_{31}^2 = NP \]
\[ q_{22}^3 = NP \]
\[ q_{32}^3 = NP \]

= the
= fund

Marten van Schijndel
Incremental Coarse-to-Fine Parsing
Mix it all up

[\[ q^d_{gt} \]]

\[
\begin{align*}
q_1^1 & \rightarrow g_{11}^1 \rightarrow g_{21}^1 \rightarrow q_2^1 \\
q_1^2 & \rightarrow g_{11}^2 \rightarrow g_{21}^2 \rightarrow q_2^2 \\
q_1^3 & \rightarrow g_{11}^3 \rightarrow g_{21}^3 \rightarrow q_2^3 \\
q_1^4 & \rightarrow g_{11}^4 \rightarrow g_{21}^4 \rightarrow q_2^4 \\
p_{01} & \rightarrow X_1 \\
p_{11} & \\
p_{21} & \\
p_{02} & \\
p_{12} & \\
p_{22} & \rightarrow X_2 \\
f_2 & \rightarrow \text{the} \\
f_3 & \rightarrow \text{fund}
\end{align*}
\]

= NP_0/NN_0

= NP_0/NN_2

= NP_0/NN_4

= S_0/VP_0

= DT

= NN_0

= NN_2

= NN_4

= NP_0
How does it work?

Theory/Equation time
Most likely sequence

\[ \hat{q}_{1:D}^{1:T} \overset{\text{def}}{=} \arg\max_{q_{1:D}^{1:T}} \prod_{t=1}^{T} P_{\theta_Q}(q_{t}^{1:D} | q_{t-1}^{1:D}, p_{t-1}) \cdot P_{\theta_{P,d'}}(p_{t} | b_{t}^{d'}) \cdot P_{\theta_X}(x_{t} | p_{t}) \] (1)

where \( d' \) is the lowest non-empty \( q_{t}^{d} \)
How does it work?

Theory/Equation time
Right-Corner: Single expansion, Single reduction
E-R+, E-R-, E+R+, E+R-

\[ \theta_Q \]

\[ P_{\theta_Q}(q_t^{1..D} | q_{t-1}^{1..D} p_{t-1}) \]

\[ \overset{\text{def}}{=} P_{\theta_F}(0' | b_{t-1}^{d'} p_{t-1}) \cdot P_{\theta_{A,d'}}(\text{'}- | b_{t-1}^{d'-1} a_{t-1}^{d'}) \cdot \left[ a_{t-1}^{d'-1} = a_{t-1}^{d'-1} \right] \cdot P_{\theta_{B,d'-1}}(b_{t-1}^{d'-1} | b_{t-1}^{d'-1} a_{t-1}^{d'}) \]

\[ \cdot \left[ q_t^{1..d'-2} = q_{t-1}^{1..d'-2} \right] \cdot \left[ q_{t}^{d'+..D} = \text{'}- \right] \]

\[ + P_{\theta_F}(0' | b_{t-1}^{d'} p_{t-1}) \cdot P_{\theta_{A,d'}}(a_{t}^{d'} | b_{t-1}^{d'-1} a_{t-1}^{d'}) \cdot P_{\theta_{B,d'}}(b_{t}^{d'} | a_{t}^{d'} a_{t-1}^{d'+1}) \]

\[ \cdot \left[ q_t^{1..d'-1} = q_{t-1}^{1..d'-1} \right] \cdot \left[ q_{t}^{d'+1..D} = \text{'}- \right] \]

\[ + P_{\theta_F}(1' | b_{t-1}^{d'} p_{t-1}) \cdot P_{\theta_{A,d'}}(\text{'}- | b_{t-1}^{d'} p_{t-1}) \cdot \left[ a_{t}^{d'} = a_{t}^{d'} \right] \cdot P_{\theta_{B,d'}}(b_{t}^{d'} | b_{t-1}^{d'} p_{t-1}) \]

\[ \cdot \left[ q_t^{1..d'-1} = q_{t-1}^{1..d'-1} \right] \cdot \left[ q_{t}^{d'+1..D} = \text{'}- \right] \]

\[ + P_{\theta_F}(1' | b_{t-1}^{d'} p_{t-1}) \cdot P_{\theta_{A,d'}}(a_{t}^{d'+1} | b_{t-1}^{d'} p_{t-1}) \cdot P_{\theta_{B,d'}}(b_{t}^{d'+1} | a_{t}^{d'+1} p_{t-1}) \]

\[ \cdot \left[ q_t^{1..d'} = q_{t-1}^{1..d'} \right] \cdot \left[ q_{t}^{d'+2..D} = \text{'}- \right] \]

(2)
How does it work?

Theory/Equation time

\[ \theta_{F,d,g} \]

\[
P_{\theta_{F,d,g}}(f_G | b_G, p_G) \stackrel{\text{def}}{=} \begin{cases} 
P_{\theta_{F,d}}(f_G | b_G, p_G) & \text{if } g = 0 \\
1 & \text{else}
\end{cases}
\] (3)

\[ \theta_{A,d,g} \]

\[
P_{\theta_{A,d,g}}(a_G | b_G, f_G, \pi(a_g)) \stackrel{\text{def}}{=} \frac{\max_{a_G | a_g \prec a_G} P_{\theta_{A,d,g}}(a_G | b_G, f_G, \pi(a_G))}{\max_{a'_G | \pi(a_g) \prec a'_G} P_{\theta_{A,d,g}}(a'_G | b_G, f_G, \pi(a'_G))}
\] (4)

\[
= \frac{\max_{a_G | a_g \prec a_G} P_{\theta_{A,d}}(a_G | b_G, f_G)}{\max_{a'_G | \pi(a_g) \prec a'_G} P_{\theta_{A,d}}(a'_G | b_G, f_G)}
\] (5)
How does it work?

Theory/Equation time

\( \theta_{B,d,g} \)

a) Active Transition

\[
P_{\theta_{B,d,g}}(b_g \mid a_g, f_G, \pi(b_g)) \overset{\text{def}}{=} \frac{\max_{b_G, a_G} b_g \prec b_G, a_g \prec a_G P_{\theta_{B,d,g}}(b_G \mid a_G, f_G, \pi(b_G))}{\max_{b_G', a_G} \pi(b_g) \prec b_G', a_g \prec a_G P_{\theta_{B,d,g}}(b_G' \mid a_G, f_G, \pi(b_G'))}
\]

\[= \frac{\max_{b_G, a_G} b_g \prec b_G, a_g \prec a_G P_{\theta_{B,d}}(b_G \mid a_G, f_G)}{\max_{b_G', a_G} \pi(b_g) \prec b_G', a_g \prec a_G P_{\theta_{B,d}}(b_G' \mid a_G, f_G)}
\]

(6)

b) Awaited Transition

\[
P_{\theta_{B,d,g}}(b_g \mid b_G', f_G, \pi(b_g)) \overset{\text{def}}{=} \frac{\max_{b_G} b_g \prec b_G P_{\theta_{B,d,g}}(b_G \mid b_G', f_G, \pi(b_G))}{\max_{b_G'} \pi(b_g) \prec b_G', b_g \prec b_G P_{\theta_{B,d,g}}(b_G' \mid b_G', f_G, \pi(b_G'))}
\]

\[= \frac{\max_{b_G} b_g \prec b_G P_{\theta_{B,d}}(b_G \mid b_G', f_G)}{\max_{b_G'} \pi(b_g) \prec b_G', b_g \prec b_G P_{\theta_{B,d}}(b_G' \mid b_G', f_G)}
\]

(8)
How does it work?

Theory/Equation time

$$\theta_{P,d,g}$$

$$P_{\theta_{P,d,g}}(p_g \mid b_g, \pi(p_g)) \overset{\text{def}}{=} \frac{\max_{p_G,b_G \mid p_g \prec p_G, b_g \prec b_G} P_{\theta_{P,d,g}}(p_G \mid b_G, \pi(p_G))}{\max_{p'_G,b_G \mid \pi(p_g) \prec p'_G, b_g \prec b_G} P_{\theta_{P,d,g}}(p'_G \mid b_G, \pi(p'_G))}$$

$$= \frac{\max_{p_G,b_G \mid p_g \prec p_G, b_g \prec b_G} P_{\theta_{P,d}}(p_G \mid b_G)}{\max_{p'_G,b_G \mid \pi(p_g) \prec p'_G, b_g \prec b_G} P_{\theta_{P,d}}(p'_G \mid b_G)}$$

(10)

$$\theta_{X,g}$$

$$P_{\theta_{X,g}}(x \mid p_g) \overset{\text{def}}{=} \frac{\max_{p_G \mid p_g \prec p_G} P_{\theta_{X,g}}(x \mid p_G)}{\max_{p'_G \mid \pi(p_g) \prec p'_G} P_{\theta_{X,g}}(x \mid p'_G)}$$

$$= \frac{\max_{p_G \mid p_g \prec p_G} P_{\theta_X}(x \mid p_G)}{\max_{p'_G \mid \pi(p_g) \prec p'_G} P_{\theta_X}(x \mid p'_G)}$$

(12)
## Paydirt

### Timing Results

<table>
<thead>
<tr>
<th>System</th>
<th>CTF-FAWP</th>
<th>FAWP</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>5sm-2000</td>
<td>30.3</td>
<td>61.05</td>
<td>0.496</td>
</tr>
<tr>
<td>4sm-500</td>
<td>4.16</td>
<td>7.17</td>
<td>0.580</td>
</tr>
<tr>
<td>3sm-500</td>
<td>2.11</td>
<td>4.83</td>
<td>0.437</td>
</tr>
<tr>
<td>2sm-500</td>
<td>1.64</td>
<td>3.35</td>
<td>0.490</td>
</tr>
<tr>
<td>Ave</td>
<td></td>
<td></td>
<td>0.50</td>
</tr>
</tbody>
</table>

Timing results with varying sm. (sec/sent)
## CTF-FAWP Timing Results

<table>
<thead>
<tr>
<th>System</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3sm-2000</td>
<td>8.27</td>
</tr>
<tr>
<td>3sm-1000</td>
<td>4.26</td>
</tr>
<tr>
<td>3sm-500</td>
<td>2.00</td>
</tr>
<tr>
<td>3sm-250</td>
<td>0.87</td>
</tr>
<tr>
<td>3sm-100</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Timing results with varying beam-width. (sec/sent)
### CTF Accuracy Results

<table>
<thead>
<tr>
<th>System</th>
<th>Recall</th>
<th>Prec</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Petrov Klein (Reported, 10-best)</td>
<td>91.2</td>
<td>91.1</td>
<td>91.2</td>
</tr>
<tr>
<td>Petrov Klein (5sm, U+B, 1-best)</td>
<td>88.5</td>
<td>88.8</td>
<td>88.7</td>
</tr>
<tr>
<td>Petrov Klein (5sm, Binary, 1-best)</td>
<td>88.2</td>
<td>87.9</td>
<td>88.0</td>
</tr>
<tr>
<td>FAWP (5sm, b5000)</td>
<td>87.9</td>
<td>87.7</td>
<td>87.8</td>
</tr>
<tr>
<td>CTF-FAWP (5sm, b5000)</td>
<td>88.0</td>
<td>87.6</td>
<td>87.8</td>
</tr>
<tr>
<td>FAWP (5sm, b2000)</td>
<td>87.7</td>
<td>87.6</td>
<td>87.6</td>
</tr>
<tr>
<td>CTF-FAWP (5sm, b2000)</td>
<td>86.2</td>
<td>86.3</td>
<td>86.3</td>
</tr>
</tbody>
</table>

Accuracy of CTF on various incarnations of FAWP.
Future Work
Where to now?

- Condition on more variables (MaxEnt)
- Weight predictions based on proportion of total beam predictions; More NP predictions make NP a better guess.


The Model

Marten van Schijndel
Incremental Coarse-to-Fine Parsing