Connectionist-Inspired Incremental PCFG Parsing

Marten van Schijndel\textsuperscript{a}, Andy Exley\textsuperscript{b}, William Schuler\textsuperscript{a}
\textsuperscript{a}Dept Linguistics, The Ohio State University
\textsuperscript{b}Dept Computer Science and Engineering, University of Minnesota

June 7, 2012
Introduction

Goals and Motivation

Create a cognitively-motivated parser

- [Schuler, 2009] outlines a cognitively-motivated parser, which requires book-keeping nodes built in to work with PCFGs (engineering fix).
- We’d like to be able to strip out elements included solely for engineering.
Background

Why PCFGs? [Jurafsky, 1996]

- Simple
- Widespread use, community understanding
- Easily integrated with other technologies
- Latent variable training procedures easily obtained [Petrov et al., 2006]
- Tractable recognition $O(n^3)$

Problems with CKY

- Not incremental $O(n^3)$
- In certain applications, word/phrase breaks not certain (ASR, MT, etc)
Background

Why Incremental?

- Operates on incomplete information
- Can make use of information about recent content/structure
- $O(n)$
- Streaming task

Must operate on a beam to efficiently stream
The Setup

S
  NP
    DT the
    NN fund
  VP
    VB bought
    NP
date
  RB
today

S/VP
  NP
    DT two
    JJ regional
    NN banks

S/VP
  NP
    DT the
    NN fund
date
  RB
today

V/NN
  NN
date

The Setup

Marten van Schijndel, Andy Exley, William Schuler
Connectionist-Inspired Incremental PCFG Parsing
Corresponding structure seen in C-R axis of DL-PFC (proximal to Broca’s) [Petrides, 1987, Botvinick, 2007]
Cognitive Motivation

- Can define graph-theory **connected components** (sub-graphs) of a semantic dependency graph (of ‘concepts’ [Kintsch, 1988] or discourse referents)
- F-node = create new independent connected component linked via an episodic trace [Sederberg et al., 2008] to previous connected component
- Connected components act as ‘chunks’ [Miller, 1956]
Design Motivations

Schuler (2009) based on:

- HHMM [Murphy and Paskin, 2001] but too general (next slide)
- 4 layers [Cowan, 2001]

Serial recall chunking [Miller, 1956] seems to be different from language chunking or chunking with distractions [Cowan, 2001].

[Schuler et al., 2010] found 4 layers yielded >99.9% coverage of WSJ.
Single Expansion, Single Reduction

NP | VP
---|---
the fund | bought regional banks

NP | VP
---|---
two NN | regional

VP/NN | VP/NP
---|---
the fund | the two bought

VP/NN | VP
---|---
regional banks | bought today
The Model
Tree Training

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

- Input: TB-annotated sentences
  \( (S (ADVP \text{happily}) (NP-SUBJ \text{John})\ldots ) \)
Tree Training

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

- Input: TB-annotated sentences
  \((S (ADVP happily) (NP-SUBJ John)\ldots)\)
- EM classification performed over a given number of split-merge cycles
- Output: Subcat-Annotated PCFG
  \((S^{g}_{10} \rightarrow ADVP^{g}_{21} NP^{g}_{4} 1.462527E-18)\)

Profit:
- More specialized and informative PCFG

Cost:
- Training time
- Increased size of grammar
Testing Methodology
Internal Testing
  ▶ Timing Comparisons [Hidden State Factoring]
External Testing
  ▶ Roark (2001) Parser [Incremental]
# Accuracy Results

<table>
<thead>
<tr>
<th>System</th>
<th>R</th>
<th>P</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schuler et al. 2008/2010</td>
<td>83.4</td>
<td>83.7</td>
<td>83.5</td>
</tr>
<tr>
<td>Roark 2001</td>
<td>86.6</td>
<td>86.5</td>
<td>86.5</td>
</tr>
<tr>
<td>Schuler 2009* (2000)</td>
<td>87.9</td>
<td>87.8</td>
<td>87.8</td>
</tr>
<tr>
<td>van Schijndel et al (250)</td>
<td>85.6</td>
<td>87.1</td>
<td>86.3</td>
</tr>
<tr>
<td>van Schijndel et al (500)</td>
<td>86.8</td>
<td>87.4</td>
<td>87.1</td>
</tr>
<tr>
<td>van Schijndel et al (1000)</td>
<td>87.4</td>
<td>87.6</td>
<td>87.5</td>
</tr>
<tr>
<td>van Schijndel et al (2000)</td>
<td>87.9</td>
<td>87.8</td>
<td>87.8</td>
</tr>
<tr>
<td>van Schijndel et al (5000)</td>
<td>87.9</td>
<td>87.8</td>
<td>87.8</td>
</tr>
<tr>
<td>Petrov Klein (Binary)</td>
<td>88.1</td>
<td>87.8</td>
<td>88.0</td>
</tr>
<tr>
<td>Petrov Klein (+Unary)</td>
<td>88.3</td>
<td>88.6</td>
<td>88.5</td>
</tr>
</tbody>
</table>

*Without grammar trainer, Schuler 2009 (2000) F-Score = 75.06.*
Table: Speed comparison using a beam-width of 500 elements

<table>
<thead>
<tr>
<th>System</th>
<th>Sec/Sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schuler 2009</td>
<td>74</td>
</tr>
<tr>
<td>Current Model</td>
<td>12</td>
</tr>
</tbody>
</table>
Future Work

- Incremental Dependency Parsing (including Unbounded)
- Incremental Semantic Role Labelling
- Interactive associative memory access
- Coreference resolution
Questions?

Thanks!
More slides!
<table>
<thead>
<tr>
<th>System</th>
<th>R</th>
<th>P</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schuler et al. 2008/2010</td>
<td>83.4</td>
<td>83.7</td>
<td>83.5</td>
</tr>
<tr>
<td>Roark 2001</td>
<td>86.6</td>
<td>86.5</td>
<td>86.5</td>
</tr>
<tr>
<td>Schuler 2009 (2000)</td>
<td>87.9</td>
<td>87.8</td>
<td>87.8</td>
</tr>
<tr>
<td>van Schijndel et al (50)</td>
<td>75.9</td>
<td>84.6</td>
<td>80.0</td>
</tr>
<tr>
<td>van Schijndel et al (100)</td>
<td>81.7</td>
<td>85.6</td>
<td>83.6</td>
</tr>
<tr>
<td>van Schijndel et al (250)</td>
<td>85.6</td>
<td>87.1</td>
<td>86.3</td>
</tr>
<tr>
<td>van Schijndel et al (500)</td>
<td>86.8</td>
<td>87.4</td>
<td>87.1</td>
</tr>
<tr>
<td>van Schijndel et al (1000)</td>
<td>87.4</td>
<td>87.6</td>
<td>87.5</td>
</tr>
<tr>
<td>van Schijndel et al (1500)</td>
<td>87.6</td>
<td>87.7</td>
<td>87.7</td>
</tr>
<tr>
<td>van Schijndel et al (2000)</td>
<td>87.9</td>
<td>87.8</td>
<td>87.8</td>
</tr>
<tr>
<td>van Schijndel et al (5000)</td>
<td>87.9</td>
<td>87.8</td>
<td>87.8</td>
</tr>
<tr>
<td>Petrov Klein (Binary)</td>
<td>88.1</td>
<td>87.8</td>
<td>88.0</td>
</tr>
<tr>
<td>Petrov Klein (+Unary)</td>
<td>88.3</td>
<td>88.6</td>
<td>88.5</td>
</tr>
</tbody>
</table>
How does it work?

Theory/Equation time

Most likely sequence

\[ \hat{q}_{1:T}^{1..D} \overset{\text{def}}{=} \arg\max_{q_{1:T}^{1..D}} \prod_{t=1}^{T} P_{\theta_Q}(q_t^{1..D} \mid q_{t-1}^{1..D} p_{t-1}) \cdot P_{\theta_{P,d'}}(p_t \mid b_t^{d'}) \cdot P_{\theta_X}(x_t \mid p_t) \] (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-

\( \theta_Q \)

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

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

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

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

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

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

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

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

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

(2)
The right-corner transform (tree)
The right-corner transform (grammar)

\[ c_\eta \rightarrow c_{\eta 0} \, c_{\eta 1} \in G' \]
\[ c_\eta / c_{\eta 1} \rightarrow c_{\eta 0} \in G'' \quad (1) \]

\[ c_{\eta \ell} \rightarrow c_{\eta \ell 0} \, c_{\eta \ell 1} \in G', \ c_\eta \in C' \]
\[ c_\eta / c_{\eta \ell 1} \rightarrow c_\eta / c_{\eta \ell} \, c_{\eta \ell 0} \in G'' \quad (2) \]

\[ c_{\eta \ell} \rightarrow x_{\eta \ell} \in G', \ c_\eta \in C' \]
\[ c_\eta \rightarrow c_\eta / c_{\eta \ell} \, c_{\eta \ell} \in G'' \quad (3) \]


