When it’s time for their biannual powwow, the nation’s manufacturing titans typically jet off to the sunny confines of resort towns like Boca Raton and Hot Springs.

Why partial parsing?

- Fast
- Supports a number of large-scale NLP tasks
  - Information Extraction
  - Phrase identification for Information Retrieval
  - Question Answering

Base noun phrases

Non-recursive noun phrases (smallest NPs)

When it's time for their biannual powwow, the nation's manufacturing titans typically jet off to the sunny confines of resort towns like Boca Raton and Hot Springs.
Inductive ML algorithm

- **Simple**
  base NP = any string having the same part-of-speech tag sequence as a base NP from the training corpus

- **Combines components of existing techniques**
  - Charniak (1996)
  - Brill (1995)

- **Achieves surprisingly high accuracies**

Partial parsing framework

```
Training

Training Text
Tagger
Grammar Learning
Grammar

Test Text
Tagger
Partial Parsing
Bracketed Text
```

Rule extraction

rule = sequence of part-of-speech tags

<table>
<thead>
<tr>
<th>Training Text</th>
<th>NP Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>When/Adv it/Pron 's/Pos time/N for/P their/PosPron biannual/Adj powwow/N, the/Det nation/N 's/Pos manufacturing/Vg titans/Npl, typically/Adv jet/V off/P to/P the/Det sunny/Adj confines/Npl of/P resort/N towns/Npl like/P Boca/Name Raton/Name and/Conj Hot/Name Springs/Name.</td>
<td>Pron N PosPron Adj N Det N Vg/Npl Det Adj Npl N Npl Name Name</td>
</tr>
</tbody>
</table>

Partial parsing bracketer

- **Left-to-right**
- **Longest-match**

```
Det N? N?
```

```
Not/Adv this/Det year/N. The/Det National/Name Association/Name of/P Manufacturers/Name settled/Vp on/P the/Det city/N of/P Indianapolis/Name for/P its/PosPron fall/N board/N meeting/N.
```
Parser (bracketer)

Bracket($w_1, \ldots, w_n$):
assign p-o-s tags $t_1, \ldots, t_n$ to words $w_1, \ldots, w_n$
$$i = 1$$
while $i \leq n$ do
  $$\{r_1, \ldots, r_k\} = \text{Matches}(w_i, \ldots, w_n)$$
  $$r = \text{longest}(r_1, \ldots, r_k)$$
  make new NP from $w_i, \ldots, w_{i+|r|-1}$
  $$i = i + |r|$$

Overview of the method

Part of Speech Tagger

Rule Extraction

Part of Speech Tagger

Tagged Text

Training Text

Base NP Parser

Tagged Text

Novel Text

Grammar pruning

Initial Grammar

Evaluate Rules

Discard Worst Rule

Final Grammar

Training Phase

Application Phase

Poorly performing rules

• Sources of bad rules
  – errors in training data
  – errors in part-of-speech tagging
  – irregular & ambiguous constructs

…manufacturing/$V_g$ titans/$N_p$…
…the/Det executives/$N_p$ began/$V_p$ boarding/$V_g$ buses/$N_p$…

– score($r$) = correct($r$) - errors($r$)
– stop when worst score is positive
Advantages of the approach

- **Good performance**
- **Simple**
  - Easy to understand, implement
  - Produces intelligible grammar rules
  - Easy to update for new text genre
- **Efficient**
  - Fastest bracketing procedure

- **State of the art circa 2003**
  - ~94% P/R for NP, VP, PP chunks
  - Using ensembles of SVM’s (Kudo & Matsumoto, 2000) and Winnow as employed in Zhang et al. (2001)

### Results vs. TBL on R&M corpus

<table>
<thead>
<tr>
<th></th>
<th>TBL results</th>
<th>Pierce &amp; Cardie [98]</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/lexical templates</td>
<td>93.1P/93.5R</td>
<td></td>
<td>-3.7P/-2.6R</td>
</tr>
<tr>
<td>w/o lexical templates</td>
<td>90.5P/90.7R</td>
<td>89.4P/90.9R</td>
<td>-0.9P/+0.2R</td>
</tr>
</tbody>
</table>