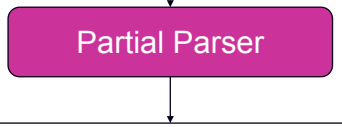
CS474 Natural Language Processing

- **Partial parsing / Chunking**
  - What is it?
  - Error-driven pruning of Treebank grammars
  - Comparison with TBL

Partial parsing

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.



When [$_{S}$ [$_{NP}$ it]] [$_{V}$'s] [$_{Obj}$ [$_{NP}$ time]] for [$_{NP}$ their biannual powwow] , [$_{NP}$ the nation] 's [$_{S}$ [$_{NP}$ manufacturing titans]] typically [$_{V}$ jet off] to [$_{NP}$ the sunny confines] of [$_{NP}$ resort towns] like [$_{NP}$ Boca Raton] and [$_{NP}$ 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] .

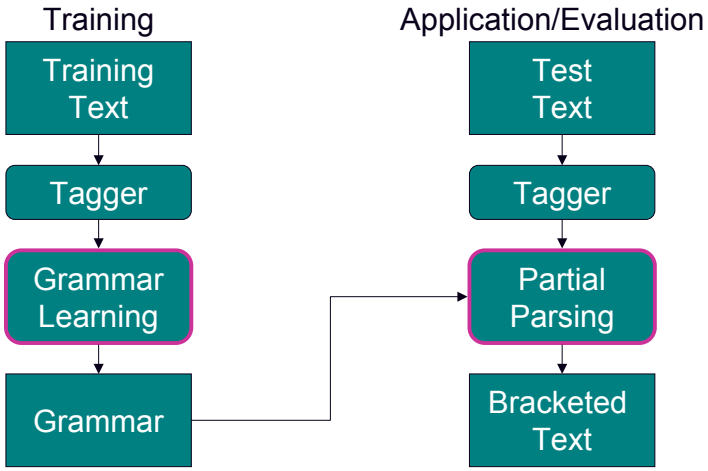
Inductive ML algorithm

- **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



Rule extraction

rule = sequence of part-of-speech tags

Training Text

When/Adv **it**/**Pron** 's/Pos **time**/**N** for/P **their**/**PosPron** **biannual**/**Adj** **powwow**/**N**, **the**/**Det** **nation**/**N** 's/Pos **manufacturing**/**V_g** **titans**/**N_{pl}** typically/Adv jet/V off/P to/P **the**/**Det** **sunny**/**Adj** **confines**/**N_{pl}** of/P **resort**/**N** **towns**/**N_{pl}** like/P **Boca**/**Name** **Raton**/**Name** and/Conj **Hot**/**Name** **Springs**/**Name**.

NP Grammar

Pron
N
PosPron Adj N
Det N
V_g N_{pl}
Det Adj N_{pl}
N N_{pl}
Name Name

Partial parsing bracketer

- Left-to-right
- Longest-match

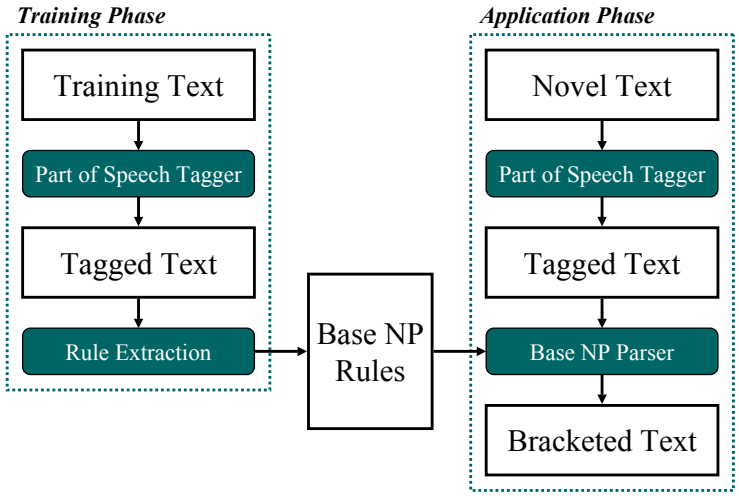
Det N? N?

Not/Adv **this**/**Det** **year**/**N**. The/Det National/Name Association/Name of/P Manufacturers/Name settled/V_p 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,…, w_n):
assign p-o-s tags t_1,…, t_n to words w_1,…, w_n
i = 1
while i ≤ n do
    {r_1,…, r_k} = Matches(w_i,…, w_n)
    r = longest(r_1,…, r_k)
    make new NP from w_i,…, $w_{i+|r|-1}$
    i = i + |r|

Overview of the method



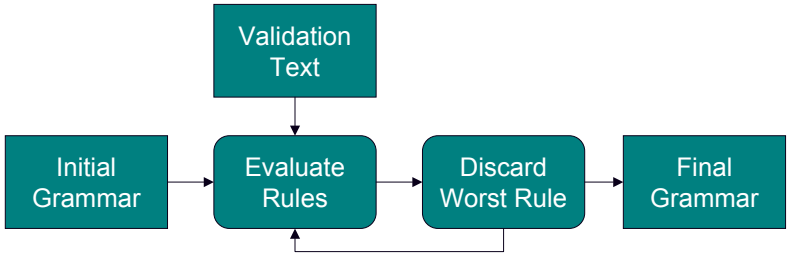
Poorly performing rules

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

…**manufacturing/V_g titans/N_{pl}**…

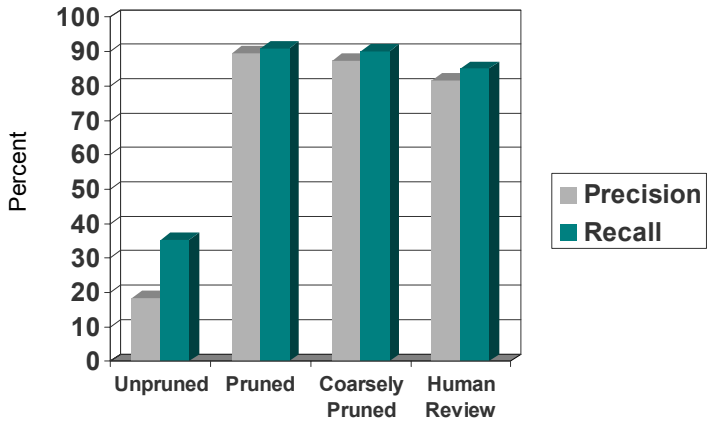
…the/Det executives/N_{pl} began/V_p **boarding/V_g buses/N_{pl}**…

Grammar pruning



- score(r) = correct(r) - errors(r)
- stop when worst score is positive

Results



Results vs. TBL on R&M corpus

	TBL results	Pierce & Cardie [98]	Difference
w/lexical templates	93.1P/93.5R		-3.7P/-2.6R
w/o lexical templates	90.5P/90.7R	89.4P/90.9R	-0.9P/+0.2R

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**
  - ~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)