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.

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

\[ \text{rule} = \text{sequence of part-of-speech tags} \]

Partial parsing bracketer

- Left-to-right
- Longest-match
Parser (bracketer)

Bracket(w₁,…, wₙ):
assign p-o-s tags t₁,…, tₙ to words w₁,…, wₙ
i = 1
while i ≤ n do
  {r₁,…, rₖ} = Matches(wᵢ,…, wₙ)
  r = longest(r₁,…, rₖ)
  make new NP from wᵢ,…, wᵢ+|r|-1
  i = i + |r|

Overview of the method

Part of Speech Tagger
Rule Extraction
Part of Speech Tagger
Tagged Text
Base NP Parser
Tagged Text
Novel Text
Bracketed Text

Poorly performing rules

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

Grammar pruning

– 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**
  - ~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)