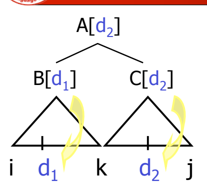
# Seven Lectures on Statistical Parsing

Christopher Manning
LSA Linguistic Institute 2007
LSA 354
Lecture 5

# A bit more on lexicalized parsing

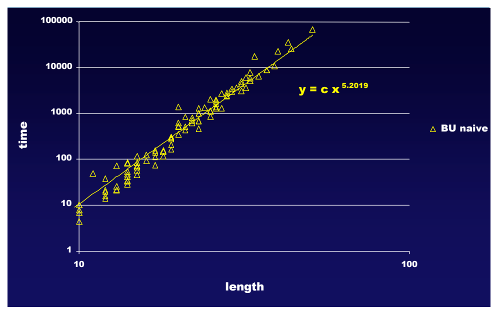
## Complexity of lexicalized PCFG parsing




Time charged :
- i, k, j $\Rightarrow$ $n^3$
- A, B, C $\Rightarrow$ $g^3$
- Naively, $g$ becomes huge
- $d_1$, $d_2$ $\Rightarrow$ $n^2$

Running time is $O(g^3 \times n^5)$ !!

## Complexity of exhaustive lexicalized PCFG parsing




## Complexity of lexicalized PCFG parsing

- Work such as Collins (1997) and Charniak (1997) *is* $O(n^5)$ – but uses heuristic search to be fast in practice
- Eisner and Satta (2000, etc.) have explored various ways to parse more restricted classes of bilexical grammars in $O(n^4)$ or $O(n^3)$ time
  - Neat algorithmic stuff!!!
  - See example later from dependency parsing

## Refining the node expansion probabilities

- Charniak (1997) expands each phrase structure tree in a single step.
- This is good for capturing dependencies between child nodes
- But it is bad because of data sparseness
- A pure dependency, one child at a time, model is worse
- But one can do better by in between models, such as generating the children as a Markov process on both sides of the head (Collins 1997; Charniak 2000)
  - Cf. the accurate unlexicalized parsing discussion

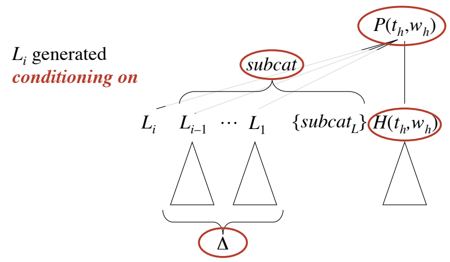
## Collins (1997, 1999); Bikel (2004)

- Collins (1999): also a generative model
- Underlying lexicalized PCFG has rules of form

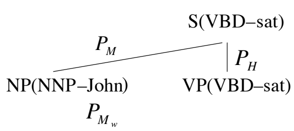
$$P \rightarrow L_j L_{j-1} \cdots L_1 H R_1 \cdots R_{k-1} R_k$$

- A more elaborate set of grammar transforms and factorizations to deal with data sparseness and interesting linguistic properties
- So, generate each child in turn: given *P* has been generated, generate *H*, then generate modifying nonterminals from head-adjacent outward with some limited conditioning

## Overview of Collins' Model




## Modifying nonterminals generated in two steps




## Smoothing for head words of modifying nonterminals

| Back-off level | $P_{M_w}(w_{M_i} \mid \ldots)$ |
|---|---|
| 0 | $M_i, t_{M_i}$, coord, punc, $P, H, w_h, t_h, \Delta_M, subcat_{side}$ |
| 1 | $M_i, t_{M_i}$, coord, punc, $P, H, t_h, \Delta_M, subcat_{side}$ |
| 2 | $t_{M_i}$ |

- Other parameter classes have similar or more elaborate backoff schemes

## Collins model … and linguistics

- Collins had 3 generative models: Models 1 to 3
- Especially as you work up from Model 1 to 3, significant linguistic modeling is present:
  - Distance measure: favors close attachments
    - Model is sensitive to punctuation
  - Distinguish base NP from full NP with post-modifiers
  - Coordination feature
  - Mark gapped subjects
  - Model of subcategorization; arguments vs. adjuncts
  - Slash feature/gap threading treatment of displaced constituents
    - Didn't really get clear gains from this.

## Bilexical statistics: Is use of maximal context of $P_{M_w}$ useful?

- Collins (1999): "Most importantly, the model has parameters corresponding to dependencies between pairs of headwords."
- Gildea (2001) reproduced Collins' Model 1 (like regular model, but no subcats)
  - Removing maximal back-off level from $P_{M_w}$ resulted in only 0.5% reduction in F-measure
  - Gildea's experiment somewhat unconvincing to the extent that his model's performance was lower than Collins' reported results

## Choice of heads

- If not bilexical statistics, then surely choice of heads is important to parser performance…
- Chiang and Bikel (2002): parsers performed decently even when all head rules were of form "if parent is X, choose left/rightmost child"
- Parsing engine in Collins Model 2–emulation mode: LR 88.55% and LP 88.80% on §00 (sent. len. ≤40 words)
  - compared to LR 89.9%, LP 90.1%

## Use of maximal context of $P_{M_w}$

[Bikel 2004]

| | LR | LP | CBs | 0 CBs | ≤2 CBs |
|---|---|---|---|---|---|
| Full model | 89.9 | 90.1 | 0.78 | 68.8 | 89.2 |
| No bigrams | 89.5 | 90.0 | 0.80 | 68.0 | 88.8 |

Performance on §00 of Penn Treebank on sentences of length ≤40 words

## Use of maximal context of $P_{M_w}$

| Back-off level | Number of accesses | Percentage |
|---|---|---|
| 0 | 3,257,309 | 1.49 |
| 1 | 24,294,084 | 11.0 |
| 2 | 191,527,387 | 87.4 |
| Total | 219,078,780 | 100.0 |

Number of times parsing engine was able to deliver a probability for the various back-off levels of the mod-word generation model, $P_{M_w}$, when testing on §00 having trained on §§02–21

## Bilexical statistics *are* used often

[Bikel 2004]

- The 1.49% use of bilexical dependencies suggests they don't play much of a role in parsing
- But the parser pursues many (very) incorrect theories
- So, instead of asking how often the decoder can use bigram probability *on average*, ask how often *while pursuing its top-scoring theory*
- Answering question by having parser *constrain-parse* its own output
  - train as normal on §§02–21
  - parse §00
  - feed parse trees as *constraints*
- Percentage of time parser made use of bigram statistics shot up to **28.8%**
- So, used often, but use barely affect overall parsing accuracy
- Exploratory Data Analysis suggests explanation
  - distributions that include head words are usually sufficiently similar to those that do not as to make almost no difference in terms of accuracy

## Charniak (2000) NAACL: A Maximum-Entropy-Inspired Parser

- There was nothing maximum entropy about it. It was a cleverly smoothed generative model
- Smoothes estimates by smoothing ratio of conditional terms (which are a bit like maxent features):

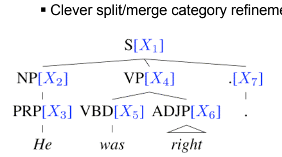
$$\frac{P(t \mid l, l_p, t_p, l_g)}{P(t \mid l, l_p, t_p)}$$

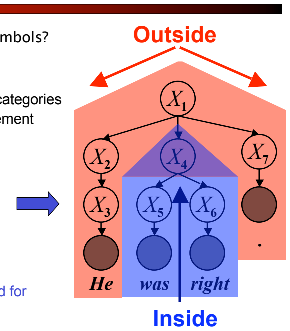
- Biggest improvement is actually that generative model predicts head tag first and then does P(*w*|*t*,…)
  - Like Collins (1999)
- Markovizes rules similarly to Collins (1999)
- Gets 90.1% LP/LR F score on sentences ≤ 40 wds

## Petrov and Klein (2006): Learning Latent Annotations

Can you automatically find good symbols?

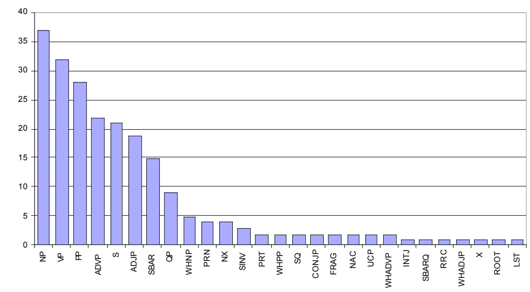
- Brackets are known
- Base categories are known
- Induce subcategoriesInduce subcategories
- Clever split/merge category refinement




EM algorithm, like Forward-Backward for HMMs, but constrained by tree.

## Number of phrasal subcategories




## POS tag splits, commonest words: effectively a class-based model

- Proper Nouns (NNP):

| | | | |
|---|---|---|---|
| NNP-14 | Oct. | Nov. | Sept. |
| NNP-12 | John | Robert | James |
| NNP-2 | J. | E. | L. |
| NNP-1 | Bush | Noriega | Peters |
| NNP-15 | New | San | Wall |
| NNP-3 | York | Francisco | Street |

- Personal pronouns (PRP):

| | | | |
|---|---|---|---|
| PRP-0 | It | He | I |
| PRP-1 | it | he | they |
| PRP-2 | it | them | him |

## The Latest Parsing Results…

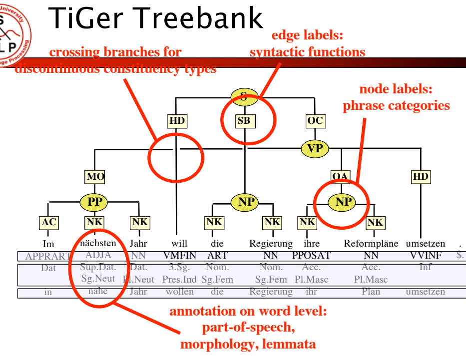
| *Parser* | *F1 ≤ 40 words* | *F1 all words* |
|---|---|---|
| Klein & Manning unlexicalized 2003 | 86.3 | 85.7 |
| Matsuzaki et al. simple EM latent states 2005 | 86.7 | 86.1 |
| Charniak generative ("maxent inspired") 2000 | 90.1 | 89.5 |
| Petrov and Klein NAACL 2007 | 90.6 | 90.1 |
| Charniak & Johnson discriminative reranker 2005 | 92.0 | 91.4 |

## Treebanks and linguistic theory

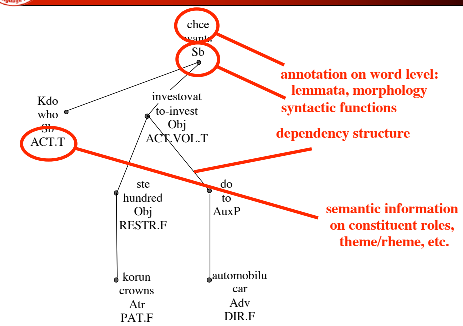
## Treebanks

- Treebank and parsing experimentation has been dominated by the Penn Treebank
- If you parse other languages, parser performance generally heads, downhill, even if you normalize for treebank size:
  - WSJ small 82.5%
  - Chinese TB 75.2%
- What do we make of this? We're changing several variables at once.
  - "Is it harder to parse Chinese, or the Chinese Treebank?" [Levy and Manning 2003]
- What is the basis of the structure of the Penn English Treebank anyway?

## TiGer Treebank



## Prague Dependency Bank




## Treebanks

- A good treebank has an extensive manual for each stage of annotation
  - Consistency is often at least as important as being right
  - But is what's in these treebanks right?
- Tokenization:
  - *has n't*  *I 'll*
- Hyphenated terms
  - *cancer-causing/JJ asbestos/NN*
  - *the back-on-terra-firma/JJ toast/NN*
  - *the/DT nerd-and-geek/JJ club/NN*

## Treebank: POS

- Some POS tagging errors reflect not only human inconsistency but problems in the definition of POS tags, suggestive of clines/blends
- Example: *near*
- In Middle English, an adjective
- Today is it an adjective or a preposition?
  - *The near side of the moon*
  - *We were near the station*
- Not just a word with multiple parts of speech! Evidence of blending:
  - *I was nearer the bus stop than the train*

## Criteria for Part Of Speech

- In some cases functional/notional tagging dominates structure in Penn Treebank, even against explicit instructions to the contrary:
- *worth:* 114 instances
  - 10 tagged IN (8 placed in ADJP!)
  - 65 tagged JJ (48 in ADJP, 13 in PP, 4 NN/NP errors)
  - 39 tagged NN (2 IN/JJ errors)
- Linguist hat on: I tend to agree with IN choice (when not a noun):
  - tagging accuracy only 41% for *worth!*

## Where a rule was followed: "marginal prepositions"

- Fowler (1926): "there is a continual change going on by which certain participles or adjectives acquire the character of prepositions or adverbs, no longer needing the prop of a noun to cling to"
- It is easy to have no tagging ambiguity in such cases:
- Penn Treebank (Santorini 1991):
  - "Putative prepositions ending in *-ed* or *-ing* should be tagged as past participles (VBN) or gerunds (VBG), respectively, not as prepositions (IN).
    - *According/VBG to reliable sources*
    - *Concerning/VBG your request of last week*"

## Preposition IN ⇄ Verb (VBG)

- But this makes no real sense
- Rather we see "a development caught in the act" (Fowler 1926)
- *They moved slowly, toward the main gate, following the wall*
- *Repeat the instructions following the asterisk*
- *This continued most of the week following that ill-starred trip to church*
- *He bled profusely following circumcision*
- *Following a telephone call, a little earlier, Winter had said …*
- IN: *during* [cf. *endure*], *pending*, *notwithstanding*
- ??: *concerning*, *excepting*, *considering*, …