Probabilistic Parsing

FSNLP, chapter 12

Christopher Manning and Hinrich Schütze
© 1999–2002

Modern Statistical Parsers

- A greatly increased ability to do accurate, robust, broad coverage parsing (Charniak 1997; Collins 1997; Ratnaparkhi 1997b; Charniak 2000)
- Achieved by converting parsing into a classification task and using statistical/machine learning methods
- Statistical methods (fairly) accurately resolve structural and real world ambiguities
- Much faster: rather than being cubic in the sentence length or worse, for modern statistical parsers parsing time is made linear (by using beam search)
- Provide probabilistic language models that can be integrated with speech recognition systems.

Supervised ML parsing

- Crucial resource has been treebanks of parses, especially the Penn Treebank (Marcus et al. 1993)
- From these train classifiers:
  - Mainly probabilistic models, but also:
  - Conventional decision trees
  - Decision lists/transformation-based learning
- Possible only when extensive resources exist
- Somewhat uninteresting from Cog. Sci. viewpoint – which would prefer bootstrapping from minimal supervision

A Penn Treebank tree (POS tags not shown)

```
( (S (NP-SBJ The move)
     (VP followed
         (NP (NP a round)
             (PP of
                 (NP (NP similar increases)
                     (PP by
                         (NP other lenders))
                     (PP against
                         (NP Arizona real estate loans)))))
         ,
         (S-ADV (NP-SBJ *)
                (VP reflecting
                    (NP (NP a continuing decline)
                        (PP-LOC in
                                (NP that market))))))
     .))
```


Probabilistic models for parsing

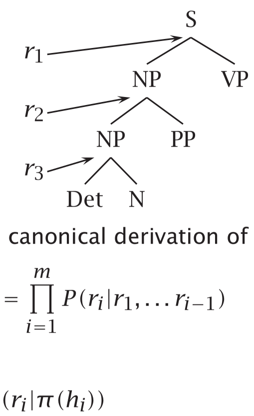
- **Conditional/Parsing model:** We estimate directly the probability of parses of a sentence

  $\hat{t} = \arg\max_t P(t|s, G) \quad \text{where} \quad \sum_t P(t|s, G) = 1$
- We don't learn from the distribution of sentences we see (but nor do we assume some distribution for them)
  - (Magerman 1995; Collins 1996)
- **Generative/Joint/Language model:**

  $$\sum_{\{t:\ \text{yield}(t) \in \mathcal{L}\}} P(t) = 1$$
- Most likely tree

  $\hat{t} = \arg\max_t P(t|s) = \arg\max_t \frac{P(t,s)}{P(s)} = \arg\max_t P(t, s)$
  - (Collins 1997; Charniak 1997, 2000)

Generative/Derivational model = Chain rule



$$P(t) = \sum_{\{d:\ d \text{ is a derivation of } t\}} P(d)$$

Or: $P(t) = P(d)$ where d is the canonical derivation of t

$$d = P(S \xrightarrow{r_1} \alpha_1 \xrightarrow{r_2} \ldots \xrightarrow{r_m} \alpha_m = s) = \prod_{i=1}^{m} P(r_i | r_1, \ldots r_{i-1})$$

- History-based grammars

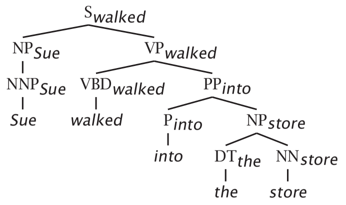
  $$P(d) = \prod_{i=1}^{m} P(r_i | \pi(h_i))$$

Enriching a PCFG

- A naive PCFG with traditional nonterminals (NP, PP, etc.) works quite poorly due to the independence assumptions it embodies (Charniak 1996)
- Fix: encode more information into the nonterminal space
  - □ Structure sensitivity (Manning and Carpenter 1997; Johnson 1998b)
    - ▸ Expansion of nodes depends a lot on their position in the tree (independent of lexical content)
    - ▸ E.g., enrich nodes by also recording their parents: ^{S}NP is different to ^{VP}NP

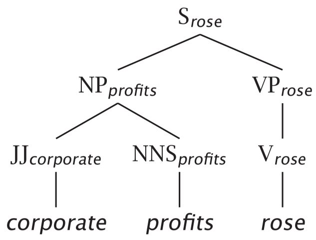
Enriching a PCFG (2)

- □ (Head) Lexicalization (Collins 1997; Charniak 1997)
  - ▸ The head word of a phrase gives a good representation of the phrase's structure and meaning
  - ▸ Puts the properties of words back into a PCFG

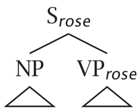


Parsing via classification decisions: Charniak (1997)

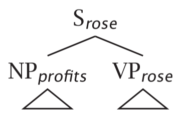
- A very simple, conservative model of lexicalized PCFG
- Probabilistic conditioning is "top-down" (but actual computation is bottom-up)

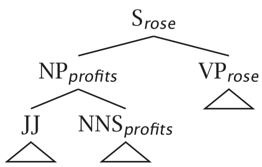


Charniak (1997) example



a. h = *profits*; c = NP
b. ph = *rose*; pc = S
c. $P(h|ph, c, pc)$
d. $P(r|h, c, pc)$





Charniak (1997) linear interpolation/shrinkage

$$\begin{aligned}\hat{P}(h|ph,c,pc) &= \lambda_1(e)P_{\mathrm{MLE}}(h|ph,c,pc)\\ &\quad+\lambda_2(e)P_{\mathrm{MLE}}(h|C(ph),c,pc)\\ &\quad+\lambda_3(e)P_{\mathrm{MLE}}(h|c,pc)+\lambda_4(e)P_{\mathrm{MLE}}(h|c)\end{aligned}$$

- $\lambda_i(e)$ is here a function of how much one would expect to see a certain occurrence, given the amount of training data, word counts, etc.
- $C(ph)$ is semantic class of parent headword
- Techniques like these for dealing with data sparseness are vital to successful model construction

Charniak (1997) shrinkage example

	P(prft|rose, NP, S)	P(corp|prft, JJ, NP)
$P(h\|ph, c, pc)$	0	0.245
$P(h\|C(ph), c, pc)$	0.00352	0.0150
$P(h\|c, pc)$	0.000627	0.00533
$P(h\|c)$	0.000557	0.00418

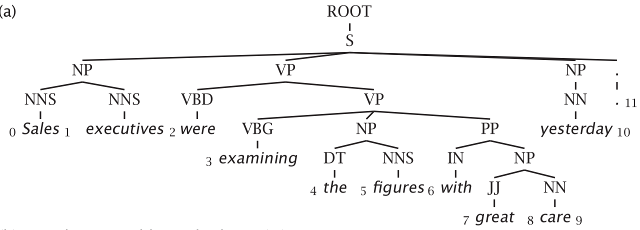
- Allows utilization of rich highly conditioned estimates, but smoothes when sufficient data is unavailable
- One can't just use MLEs: one commonly sees previously unseen events, which would have probability 0.

Probabilistic parsing

- 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)

Evaluation

(a)



(b) Brackets in gold standard tree (a.):
S-(0:11), **NP-(0:2)**, VP-(2:9), VP-(3:9), **NP-(4:6)**, PP-(6-9), NP-(7,9), *NP-(9:10)

(c) Brackets in candidate parse:
S-(0:11), **NP-(0:2)**, VP-(2:10), VP-(3:10), NP-(4:10), **NP-(4:6)**, PP-(6-10), NP-(7,10)

(d)

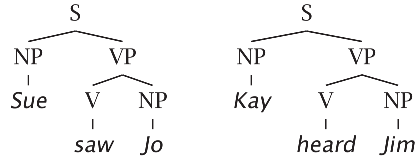
Precision:	3/8 = 37.5%	Crossing Brackets:	0
Recall:	3/8 = 37.5%	Crossing Accuracy:	100%
Labeled Precision:	3/8 = 37.5%	Tagging Accuracy:	10/11 = 90.9%
Labeled Recall:	3/8 = 37.5%		

Parser results

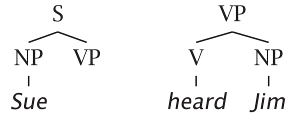
- Parsers are normally evaluated on the relation between *individual postulated nodes* and ones in the gold standard tree (Penn Treebank, section 23)
- Normally people make systems balanced for precision/recall
- Normally evaluate on sentences of 40 words or less
- Magerman (1995): about 85% labeled precision and recall
- Charniak (2000) gets 90.1% labeled precision and recall
- Good performance. Steady progress in error reduction
- At some point size of and errors in treebank must become the limiting factor
  - (Some thought that was in 1997, when several systems were getting 87.x%, but apparently not.)

Bod's (1998) Data Oriented Parsing (DOP)

- A memory-based approach using linguistic experiences
- Suppose we have a corpus of two sentences:



- To parse the new sentence *Sue heard Jim*, we put together tree fragments that we have already seen:

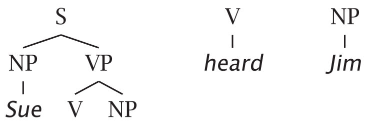


Bod's (1998) Data Oriented Parsing (DOP)

- Probability of fragment F expanding mother M:

$$P(F|M) = \#(F)/\#(M)$$

- We assume independence across fragments
- So, we just multiply to get a derivation probability
  - E.g., 8 fragments with VP as mother, 1 *heard Jim*, so P([VP [V *heard*] [NP *Jim*]]) = 1/8.
- I gave one DOP derivation of a tree. There are many:



- Need to sum over derivations to find tree probability

Data Oriented Parsing results (Bod 2000)

- The model has a lot of really bad problems:
  - The estimator is biased and inconsistent
  - Big trees are greatly over-represented in estimates
    - # of fragments is exponential in the tree size
  - There is little explicit modeling of linguistic structure, as in the head percolation of lexicalized PCFGs
- But DOP learns nonhead dependencies, topical correlations and idioms, that no other approach treats.
- Because of this, it works well:
  - 89.5% LP, 89.3% LR – better than Charniak (1997), Collins (1997); a smidgen behind Charniak (2000)