Intro to Statistical Parsing

Presenter: Benyah Shaparenko
CS 674, 3/14/2005

Statistical Parsing

- Parsing + Statistics
- Choose "best" parse

First Paper

- Eugene Charniak, "Statistical Parsing with a Context-free Grammar and Word Statistics"
- AAAI-1997

Charniak 1997 Parser

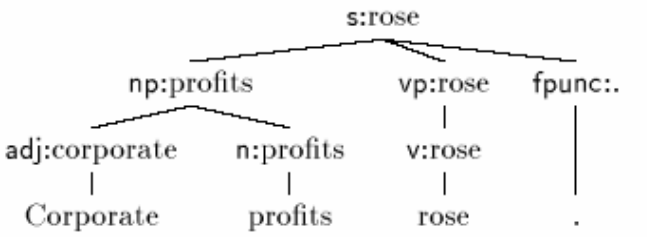
- Penn Treebank
  - Grammar, probabilities
- Uses probabilities for a good parse
  - Exhaustive search impractical
- Smoothing

Probabilistic Model

$$\mathbf{P}(s) = \arg\max_{\pi} \frac{p(\pi, s)}{p(s)} = \arg\max_{\pi} p(\pi, s)$$

- *s*: sentence, π: parse
- Probabilities calculated bottom-up
- System not guaranteed to find the best (highest probability) parse

An Example Parse



Probabilities of Constituents

- Constituent: e.g. np, vp, s
- 3 Steps (Top-down)
  - 1. Calculate probability of head
  - 2. Calculate probability of constituent given the head
  - 3. Recurse down in parse tree

Step 1: Head Probability

$$\begin{aligned} p(s \mid h,t,l) &= \lambda_1(e)\hat{p}(s \mid h,t,l) \\ &\quad +\lambda_2(e)\hat{p}(s \mid \mathbf{c}_h,t,l) \\ &\quad +\lambda_3(e)\hat{p}(s \mid t,l) + \lambda_4(e)\hat{p}(s \mid t) \end{aligned} \quad (2)$$

- s: head
- t: type of s
- h: head of parent of s
- l: type of parent of s

Step 1: Example

$$p(s \mid h,t,l) = \lambda_1(e)\hat{p}(s \mid h,t,l) + \lambda_2(e)\hat{p}(s \mid \mathbf{c}_h,t,l) + \lambda_3(e)\hat{p}(s \mid t,l) + \lambda_4(e)\hat{p}(s \mid t) \quad (2)$$

- p(profits | rose, np, s) is based on…
  - p(profits | rose, np, s) = 0
  - p(profits | class of rose, np, s) = .00352223
  - p(profits | np, s) = .0006274
  - p(profits | np) = .000556527

Step 2: Pr(Constituent | Head)

$$p(r \mid h,t,l) = \lambda_1(e)\hat{p}(r \mid h,t,l) + \lambda_2(e)\hat{p}(r \mid h,t) + \lambda_3(e)\hat{p}(r \mid \mathbf{c}_h,t) + \lambda_4(e)\hat{p}(r \mid t,l) + \lambda_5(e)\hat{p}(r \mid t) \quad (3)$$

- r: rule used for rewriting the constituent
- h, t, l as before
  - Head, type, parent type

Step 2: Example

- r is (np -> adj plural-n) in "corporate profits"
- p(r | profits, np, s) is based on…
  - p(r | profits, np, s) = .1707
  - p(r | profits, np) = .1875
  - p(r | class of profits, np) = .1192
  - p(r | np, s) = .0176
  - p(r | np) = .0255

Algorithm

- Get grammar and stats from treebank
- Use only constituents likely to be in probable parses
  - Based upon p(r | t) distribution
- Find best parse of probable parses

Experimental Setup

- Training: Sections 2-21 of Penn Treebank corpus (1 million words)
- Testing: Section 23 (50K words)
- Preliminary testing/tuning: Section 24

Systems Tested

- PCFG: only use probability p(r | t)
- Minimal: use observed ṗ(r | h, t, l)
- No classes: all except ṗ's with c_h
- Basic: all equations as described earlier
- Full: Basic + 30M words used for unsupervised learning

Metrics

- Labeled recall (LR): #right / #possible
- Labeled precision (LP): #right / #marked
- LR2/LP2 are LR/LP ignoring punctuation, and collapsing ADVP and PRT
- Crossing brackets CB: constituents violating correct boundaries
- CB0: no crossing brackets
- CB2: no more than 2 crossing brackets

Results

	LR	LR2	LP	LP2	CB	0CB	2CB
			≤ 40 words (2245 sentences)				
PCFG	71.2	71.7	75.3	75.8	2.03	39.5	68.1
Minimal	82.9	83.4	83.6	84.1	1.40	53.2	79.0
No Cls	86.2	86.8	85.8	86.4	1.14	59.9	83.4
Basic	86.3	86.8	86.6	87.1	1.09	60.7	84.0
Full	86.9	87.5	86.8	87.4	1.00	62.1	86.1
			≤ 100 words (2416 sentences)				
PCFG	70.1	70.6	74.3	74.8	2.37	37.2	64.5
Minimal	82.0	82.5	82.6	83.1	1.68	50.6	75.7
No Cls	85.4	86.0	84.9	85.5	1.37	57.2	80.6
Basic	85.5	86.0	85.6	86.2	1.32	57.8	81.1
Full	86.1	86.7	86.0	86.6	1.20	59.5	83.2

Comparison w/ Previous Work

	LR2	LP2	CB	0 CB	2CB
	$\leq$ 40 words (2245 sentences)				
Magerman	84.6	84.9	1.26	56.6	81.4
Collins	85.8	86.3	1.14	59.9	83.6
Charniak	87.5	87.4	1.0	62.1	86.1
	$\leq$ 100 words (2416 sentences)				
Magerman	84.0	84.3	1.46	54.0	78.8
Collins	85.3	85.7	1.32	57.2	80.8
Charniak	86.7	86.6	1.20	59.5	83.2

Comparison Explanations

- Non-factors
  - p(s, π) vs. p(π | s)
  - POS tags
  - Formal/explicit grammar?
  - Non-occurrence of grammar rules in data
- Factors: statistics and smoothing
  - Decision tree vs. smoothing equations
  - Word counts vs. classes
  - Unsupervised data: ~.5% LR2

Second Paper

- Michael Collins, "Three Generative, Lexicalised Models for Statistical Parsing"
- ACL/EACL-1997

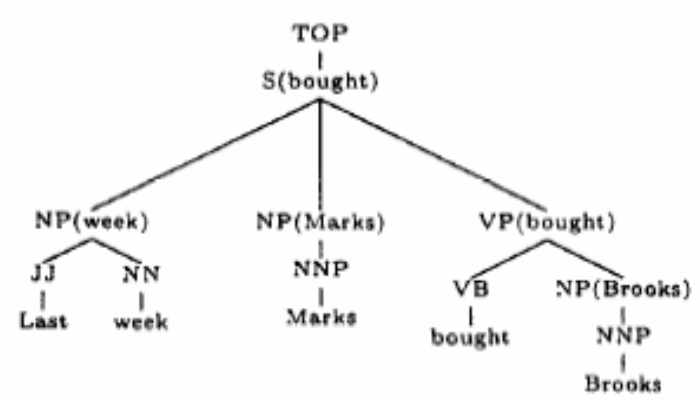
Three Generative Models

- Model 1: Collins 1996, but generative
  - Generative: the same T maximizes both p(S, T) and p(T | S)
    - Note: S is Charniak's s, T is Charniak's π
- Model 2: adds probabilistic complement/adjunct distinction
- Model 3: adds probabilistic wh-movement

Notation

- PCFG: rewrite rules w/ probabilities
- Lexicalized PCFG
  - P(h) -> $L_n(l_n)...L_1(l_1)H(h)R_1(r_1)...R_m(r_m)$
  - X(x) with nonterminal X and <word, POS tag> pair x
  - P: parent nonterminal
  - H: head child of P, with h being head word
  - L_i and R_i are modifiers of H

An Example Parse



Model 1

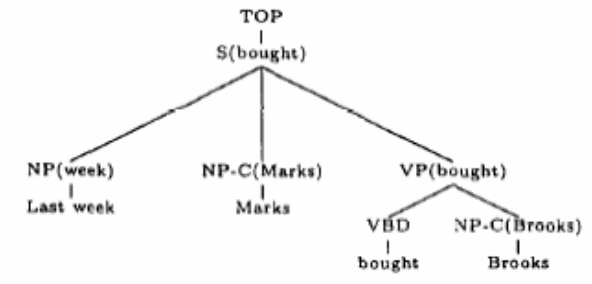
- Step 1: Generate head probability
  - $p_H(H \mid P, h)$
- Steps 2,3: Generate left,right probability
  - $p_L(L_i(l_i) \mid P, h, H)$
  - $p_R(R_i(r_i) \mid P, h, H)$
  - Uses 0-order Markov assumption
  - Can use distance, e.g.
    - $p_L(L_i(l_i) \mid P, h, H, \text{distance}_l(i-1))$

Model 1 Example

- Top rewrite rule in example parse
- Head prob.: $p_H(VP \mid S, \text{bought})$
- Left prob.: $p_L(NP(\text{marks}) \mid S, VP, \text{bought}) * p_L(...$
- Right prob.: $p_R(STOP \mid S, VP, \text{bought})$

Model 2: Subcategorization

- Model 1 + complement/adjunct division
- Example…



Model 2 Motivation

- Parsing info useful for marking complements
- May help accuracy
- Some rules for treebank data: must be NP, SBAR, S, etc under an S, cannot be ADV, VOC, etc

Model 2 Probabilities

- Head probability same: $p_H(H \mid P, h)$
- L/R subcat frames LC and RC w/ probs.
  - $p_{lc}(LC \mid P, h, H)$ and $p_{rc}(RC \mid P, h, H)$
  - Frames specify needed complements…
- L/R probabilities depend on LC/RC, e.g.
  - $p_L(L_i(l_i) \mid P, h, H, \text{distance}_l(i\text{-}1), LC)$

Model 2 Example

- p_L(NP(marks) | S, VP, bought)
  * p_L(NP(week) | S, VP, bought)
  * p_L(STOP | S, VP, bought) becomes…
- p_{lc}({NP-C} | S, VP, brought)
  * p_L(NP-C(marks) | S, VP, bought, {NP-C})
  * p_L(NP(week) | S, VP, bought, {})
  * p_L(STOP | S, VP, bought, {})

Model 3: Wh-Movement

- Refers to the effects of a wh-word, e.g. which, where, on a clause
- Parsing solution: using a +gap feature which must be matched with a TRACE
  - 1. +gap passed to head of phrase
  - 2. +gap passed to L/R modifiers or output as a TRACE

Model 3 Probabilities

- $p_G(G \mid P, h, H)$
  - G either Head, Left, or Right
- If G = Head, propagate +gap to head
- If G = Left/Right, add +gap to Left/Right subcat variable

Implementation Details

- Smoothing used
  - Backoff… pretty standard
- Unknown words
  - Words < 5 times replaced by "UNKNOWN"
- POS tags
  - Only use tags that appear in training data

Experimental Results

- Setup: same as Charniak
- Metrics: Same, except only "LR2" and "LP2," which are now called LR and LP

The Numbers…

MODEL	≤ 40 Words (2245 sentences)				
	LR	LP	CBs	0 CBs	≤ 2 CBs
(Magerman 95)	84.6%	84.9%	1.26	56.6%	81.4%
(Collins 96)	85.8%	86.3%	1.14	59.9%	83.6%
Model 1	87.4%	88.1%	0.96	65.7%	86.3%
Model 2	88.1%	88.6%	0.91	66.5%	86.9%
Model 3	88.1%	88.6%	0.91	66.4%	86.9%

≤ 100 Words (2416 sentences)				
LR	LP	CBs	0 CBs	≤ 2 CBs
84.0%	84.3%	1.46	54.0%	78.8%
85.3%	85.7%	1.32	57.2%	80.8%
86.8%	87.6%	1.11	63.1%	84.1%
87.5%	88.1%	1.07	63.9%	84.6%
87.5%	88.1%	1.07	63.9%	84.6%

Still Not the End of the Story…

- Collins (1998) applied techniques for "semantic tagging"
  - Management succession: outgoing manager, new manager, the position
- Charniak (2000) made an max entropy parser
  - Just over 90% LP / LR