November 6, 2003

Ch 12

Probabilistic and Lexicalized Parsing

Overview

- Review: PCFGs, using probabilities, learning probabilities, non-probabilistic CKY
- Probabilistic parsing with CKY
- Inside-Outside
- Problems with PCFGs
- Lexicalized PCFGs
- Other things to add to PCFGs
- Modeling human parsing

Review: PCFGs

- $G = (N, \Sigma, P, S, D)$
- N: A set of non-terminal symbols
- Σ: A set of terminal symbols (disjoint from N)
- P: A set of productions (or phrase structure rules)
  $A \longrightarrow \ \beta$ where $A \in N$ and $\beta \in (\Sigma \cup N)*$
- S: A desginated start symbol, selected from N.
- D: a function assigning probabilities to each rule in P.

Review: Using Probabilities

- Probability of a tree:

$$P(T) = \prod_{n \in T} p(r(n))$$

- The best parse: $\hat{T}(S) = \underset{T \in \tau(S)}{\operatorname{argmax}} P(T)$

Review: Finding probabilities

- Not known *a priori* like in the case of a fair die.
- Count occurences (relative frequencies) in a treebank.
- If no treebank is available, iteratively estimate with the inside-outside algorithm.

Review: CKY

- CKY (bottom-up, exhaustive)
- Two-dimensional array: #words $\times$ #words
- For each span, store all possible categories the grammar can license over that span.
- (In a separate array, store pointers back to the daughters.)

Probabilistic CKY

function CKY(*words*, *grammar*) **returns** most probable parse w/probability
    Create, clear π[*#words*,*#words*,*#non-terms*], *back*[*#words*,*#words*,*#non-terms*]
    for $i \leftarrow$ 1 to *#words*
        for $A \leftarrow$ 1 to *#non-terms*
            if ($A \rightarrow w_i$ is in *grammar*) then
                $\pi[i, i, A] \leftarrow P(A \rightarrow w_i)$
    for *span* $\leftarrow$ 2 to *#words*
        for *begin* $\leftarrow$ 1 to *#words* − *span* +1
            end $\leftarrow$ *begin* + *span* − 1
            for $m \leftarrow$ *begin* to *end* − 1
                for $A, B, C \leftarrow$ 1 to *#non-terms*
                    prob = π[*begin*,*m*,*B*] $\times$ π[$m+1$,*end*,C] $\times$ $P(A \rightarrow BC)$
                    if (*prob* $>$ π[*begin*,*end*,*A*]) then
                        π[*begin*,*end*,*A*] = *prob*
                        back[*begin*,*end*,*A*] = {*m*,*B*,*C*}
    return BUILD_TREE(*back*[1,*#words*,1]), π[1,*#words*,1]

Questions about Probabilistic CKY

- Top-down or bottom-up?
- What kind of object (data structure) is *chart* (likewise, *back*)?
- What kind of information is stored in each cell in *chart* (likewise, *back*)?
- Is this best-first or exhaustive?
- Is lexical ambiguity allowed? How?
- Where would you look to find the probability of the best parse?

Inside-Outside (EM for PCFGs)

- Start with a grammar, or just a set of non-terminals
- Assume that a good grammar is one that makes the corpus likely
- Assume that sentences in a corpus are independent (not!)
- Goal: Find probabilities for each rule that maximize the likelihood of the corpus
- Assign (perhaps randomly) some initial probability to each rule
- Parse a corpus with that grammar

Inside-Outside (EM for PCFGs)

- Assign new probabilities to each rule based on their occurrence in the corpus and weighted by the probability of each parse
- Iterate until a local maximum is reached (or at least approximated)
- (Variant of EM: Expectation Maximization)

(Manning & Schütze 1999)

Problems with Inside-Outside for learning PCFGs

- It's slow: For each sentence, each iteration of training is $O(m^3 n^3)$ where m = length of the sentence and n = the number of non-terminals in the grammar.
- Local maxima: the algorithm is very sensitive to the initialization of the parameters. (Charniak 1993)
- Satisfactory grammar learning requires $\sim$3x as many non-terms as are linguistically motivated. (Lari & Young 1990)
- No guarantee that the grammars learned ressemble the kinds of grammars that linguists write.

(Manning & Schütze 1999)

Problems with PCFGs

- Assumes the expansion of one non-terminal is independent of the expansion of any other (definition of ‘context-free’).
  - Preference for pronouns in subject position
- $\longrightarrow$ Data-Oriented Parsing (DOP) (e.g. Bod 1998)
- Lack of sensitivity to words
  - Not modeling subcategorization preferences
  - Or other lexical dendencies (cf. coordination)
- $\longrightarrow$ PHPSG, etc.
- $\longrightarrow$ Probabilistic lexicalized CFGs

Probabilistic lexicalized CFGs

- Each node encodes lex item at bottom of its head path.
- Model rule-head and head-head dependencies:

$$P(T) = \prod_{n \in T} p(r(n) \mid n, h(n)) \times p(h(n) \mid n, h(m(n)))$$

  - Given that the head is *dumped*, what is the probability of expanding this VP as V NP PP?
  - Given that the mother's head is *dumped*, what is the probability that the head of this NP is *sacks*?
- Estimating these probabilities requires smoothing and back-off techniques to deal with sparse data.

Other kinds of information to include

- Condition probability of rule on syntactic category of grandparent node
- Argument adjunct distinction
- Weighting lexical dependencies by proximity
- String-based context (three leftmost parts of speech)
- General strutural preferences

Evaluating parsers

- Create a “gold standard”
- C = # of correct constituents in candidate parse
- N = total # of constituents in candidate parse
- N_s = total # of constituents in gold standard parse
- Precision: C/N
- Recall: C/N_s
- Cross-brackets: number of occurrences of ((A B) C) for (A (B C))

More on Precision and Recall

- Precision and recall tend to conflict: maximizing one can be done at the cost of sacrificing the other.
- F-Score: balance of precision and recall:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

$\beta > 1$, precision is favored, $\beta < 1$, recall is favored.

Modeling Human Parsing

- Model attachment preferences
- Model garden-path effects:
  - Prune search space to eliminate parses below a certain probability threshhold.
  - In a garden-path, the correct parse gets pruned.
  - Do experiments with human speakers to detect garden paths of varying degrees of severity.
  - Explore which kinds of probabilistic information are required to model those results on a computer.

Summary

- Review: PCFGs, using probabilities, learning probabilities, non-probabilistic CKY
- Probabilistic parsing with CKY
- Inside-Outside
- Problems with PCFGs
- Lexicalized PCFGs
- Other things to add to PCFGs
- Modeling human parsing

Announcements

- No class Thursday
- Homework posted by Thursday, due 11/18
- Next time (Tuesday): Computational phonology