ACS Introduction to NLP

Lecture 7: Estimation for Lexicalised PCFGs

UNIVERSITY OF CAMBRIDGE

Stephen Clark

Natural Language and Information Processing (NLIP) Group

sc609@cam.ac.uk

- Probabilistic Context Free Grammars provide a ready-made solution to the statistical parsing problem
- However, it is important to realise that **parameters do not have to be associated with the rules of a context free grammar**
  - we can choose to break up the tree in any way we like
- But extracting a PCFG from the Penn Treebank and parsing with it provides a useful baseline
  - a PCFG parser obtains roughly 70-75% Parseval scores

Parameterisation of a Parse Tree

- Collins describes the following two criteria for a good parameterisation:
  - **Discriminative power**: the parameters should include the contextual information required for the disambiguation process (PCFGs fail in this regard)
  - **Compactness**: the model should have as few parameters as possible (while still retaining adequate discriminative power)

Generative Models for Parse Trees

- **Representation**
  - the set of part-of-speech tags
  - whether to pass lexical heads up the tree (lexicalisation)
  - whether to replace words with their morphological stems
- **Decomposition**
  - the order in which to generate the tree
  - the order of *decisions*, d_i, made in generating the tree
  - these decisions do not have to correspond to parsing decisions
- **Independence assumptions**
  - group decision sequences into equivalence classes, Φ

$$P(T,S) = \prod_{i=1}^{n} P(d_i|\Phi(d_1 \ldots d_{i-1}))$$

Successive Parameterisations

- Simple PCFG
- PCFG + dependencies
- Dependencies + direction
- Dependencies + direction + relations
- Dependencies + direction + relations + subcategorisation
- Dependencies + direction + relations + subcategorisation + distance
- Dependencies + direction + relations + subcategorisation + distance + parts-of-speech

- Each rule in a PCFG has the following form:

$$P(h) \rightarrow L_n(l_n) \ldots L_1(l_1) H(h) R_1(r_1) \ldots R_m(r_m)$$

P is the parent; H is the head-child; L_i and R_i are left and right modifiers (n or m may be zero)

- The probability of a rule can be written (exactly) using the chain rule:

$$\begin{aligned} p(L_n(l_n) \ldots L_1(l_1) H(h) R_1(r_1) \ldots R_m(r_m) | P(h)) = \\ p(H | P(h)) \times \\ \prod_{i=1}^{n} p(L_i(l_i) | L_1(l_1) \ldots L_{i-1}(l_{i-1}), P(h), H) \times \\ \prod_{j=1}^{m} p(R_j(r_j) | L_1(l_1) \ldots L_n(l_n), R_1(r_1) \ldots R_n(r_n), P(h), H) \end{aligned}$$

Independence Assumptions

- For Model 1, assume the modifiers are generated independently of each other:

$$p_l(L_i(l_i)|L_1(l_1)\ldots L_{i-1}(l_{i-1}), P(h), H) = p_l(L_i(l_i)|P(h), H)$$

$$p_r(R_j(r_j)|L_1(l_1)\ldots L_n(l_n), R_1(r_1)\ldots R_n(r_n), P(h), H) = p_r(R_j(r_j)|P(h), H)$$

- Example rule: S(bought) $\rightarrow$ NP(week) NP(IBM) VP(bought)

p_h(VP$|$S,bought) $\times$ p_l(NP(IBM)$|$S,VP,bought) $\times$ p_l(NP(week)$|$S,VP,bought)
$\times$ p_l(STOP$|$S,VP,bought) $\times$ p_r(STOP$|$S,VP,bought)

Subcategorisation Parameters

- A better model would distinguish optional arguments (adjuncts) from required arguments (complements)
- In *Last week IBM bought Lotus*, *Last week* is an optional argument
- Here the verb subcategorises for an NP subject to the left and an NP object to the right
  - subjects are often omitted from subcat frames for English (because every verb has a subject in English) but we'll keep them in the model

Subcategorisation Parameters

- Probability of the rule S(bought) $\rightarrow$ NP(week) NP-C(IBM) VP(bought):

$$p_h(\text{VP}|\text{S,bought}) \times p_{lc}(\{\text{NP-C}\}|\text{S,VP,bought}) \times p_{rc}(\{\}|\text{S,VP,bought}) \times$$
$$p_l(\text{NP-C(IBM)}|\text{S,VP,bought},\{\text{NP-C}\}) \times p_l(\text{NP(week)}|\text{S,VP,bought},\{\}) \times$$
$$p_l(\text{STOP}|\text{S,VP,bought},\{\}) \times p_r(\text{STOP}|\text{S,VP,bought},\{\})$$

- Easy!

$$\hat{P}(\textit{RHS}|\textit{LHS}) = \frac{f(\textit{LHS} \rightarrow \textit{RHS})}{f(\textit{LHS})}$$

where $f(\textit{LHS} \rightarrow \textit{RHS})$ is the number of times *LHS* rewrites as the *RHS* in a treebank, and $f(\textit{LHS})$ is the total number of times *LHS* is rewritten as anything

- These relative frequency estimates can be justified as maximum likelihood estimates:

$$\hat{P} = \arg\max_{P} \prod_{i=1}^{n} \prod_{j=1}^{m} P(\textit{RHS}_j^i|\textit{LHS}_j^i)$$

where $\textit{LHS}_j^i \rightarrow \textit{RHS}_j^i$ is the jth rule application in the ith training example (Collins has a proof of this)

Smoothing for Lexicalised PCFGs

- The grammar Collins uses is (roughly speaking) a lexicalised PCFG (only roughly speaking because of the Markov process generating the subcat frames)
- Lexicalised PCFGs can be thought of as PCFGs with much larger sets of non-terminal symbols (the standard non-terminals embellished with lexical items)
- So relative frequency estimation isn't going to work (many combinations of LHS's and RHS's won't appear in the data)

Backoff and Interpolation

- Backoff levels for $p_h(H|P,w,t)$ where H is the head category, P is the parent, w is the head word associated with the head category, and t is the pos tag of the head word
  - $p_h(H|P,w,t)$
  - $p_h(H|P,t)$
  - $p_h(H|P)$
- Use a linear combination of these (linear interpolation):

$$\tilde{p}_h(H|P,w,t) = \lambda_1 \hat{p}_h(H|P,w,t) + \lambda_2 \hat{p}_h(H|P,t) + \lambda_3 \hat{p}_h(H|P)$$

$$\lambda_i \geq 0, \Sigma_i \lambda_i = 1$$

- A neat way to set the values of the λs based on the *diversity*:

$$\lambda_i = \frac{f_i}{f_i + 5u_i}$$

where f_i is the number of times we've seen the denominator from the relative frequency estimate and u_i is the number of unique outcomes in the distribution (see p.185 of Collins' thesis); and 5 is set empirically

More Backoff and Interpolation

- $p_L(L_i(lw_i, lt_i)|P, H, w, t, LC)$

where $L_i(lw_i, lt_i)$ is a left complement consisting of non-terminal L_i, word lw_i, and pos tag lt_i; P is the parent category; H is the category of the head; w is the head word; t is the pos tag of the head word, and LC is the left subcat frame

$$p_L(L_i(lw_i, lt_i)|P, H, w, t, LC) = p_L(L_i(lt_i)|P, H, w, t, LC)$$
$$\times p_L(lw_i|L_i, lt_i, P, H, w, t, LC)$$

More Backoff and Interpolation

- $p_L(L_i(l_i)|P, H, w, t, LC)$

where $L_i(l_i)$ is a left complement, P is the parent category, H is the category of the head, w is the head word, t is the pos tag of the head word, and LC is the left subcat frame

- $p_L(L_i(l_i)|P, H, w, t, LC)$
- $p_L(L_i(l_i)|P, H, t, LC)$
- $p_L(L_i(l_i)|P, H, LC)$

$$p_L(L_i(l_i)|P, H, t, LC) = \lambda_1 p_L(L_i(l_i)|P, H, w, t, LC) +$$
$$\lambda_2 p_L(L_i(l_i)|P, H, t, LC) + \lambda_3 p_L(L_i(l_i)|P, H, LC)$$

Distance

- All Collins' models have "distance" parameters in them which improve the results
- I've ignored these parameters only because they clutter the equations further and adding them as extra parameters is not complicated

Results

- Model 1 achieves 87.5/87.7 LP/LR on WSJ section 23 according to the Parseval measures
- Model 2 achieves 88.1/88.3 LP/LR
- Current best scores on this task are around 92