CMPT-825
Natural Language Processing

Anoop Sarkar
http://www.cs.sfu.ca/~anoop
Supervised Models for Parsing: History-based models

- Parsing can be framed as a supervised learning task

- Induce function $f : S \rightarrow T$ given $S_i \in S$, pick best $T_i$ from $T(S)$

- Statistical parser builds model $P(T, S)$ for each $(T, S)$

- The best parse is then $\arg \max_{T \in T(S)} P(T, S)$
History-based approaches maps \((T, S)\) into a decision sequence \(d_1, \ldots, d_n\).

- Probability of tree \(T\) for sentence \(S\) is:
  \[
P(T, S) = \prod_{i=1}^{n} P(d_i \mid \phi(d_1, \ldots, d_{i-1}))
  \]

- \(\phi\) is a function that groups histories into equivalence classes.
History-based models and PCFGs

- PCFGs can be viewed as a history-based model using leftmost derivations

- A tree with rules \( \langle \gamma_i \rightarrow \beta_i \rangle \) is assigned a probability \( \prod_{i=1}^{n} P(\beta_i | \gamma_i) \) for a derivation with \( n \) rule applications
Generative models and PCFGs

$$T_{best} = \arg \max_T P(T \mid S)$$

$$= \arg \max_T \frac{P(T, S)}{P(S)}$$

$$= \arg \max_T P(T, S)$$

$$= \prod_{i=1 \ldots n} P(RHS_i \mid LHS_i)$$
Evaluation of Statistical Parsers: EVALB

Bracketing recall = \frac{\text{num of correct constituents}}{\text{num of constituents in the goldfile}}

Bracketing precision = \frac{\text{num of correct constituents}}{\text{num of constituents in the parsed file}}

Complete match = \% \text{ of sents where recall & precision are both 100\%}

Average crossing = \frac{\text{num of constituents crossing a goldfile constituent}}{\text{num of sents}}

No crossing = \% \text{ of sents which have 0 crossing brackets}

2 or less crossing = \% \text{ of sents which have } \leq 2 \text{ crossing brackets}
the company’s trials indicated no difference in the level of hypoglycemia between users of either product.
Bilexical CFG: (Collins 1997)

```
S
  ..
  |
  VP{indicated}
  |
  VB{indicated}    NP{difference}    PP{in}
  |
  indicated    difference    P    NP
  |
  ..
```
Bilexical CFG: $\text{VP}\{\text{indicate}\} \rightarrow \text{VB}\{+H:\text{indicate}\} \text{ NP}\{\text{difference}\} \text{ PP}\{\text{in}\}$

\[
P_h(\text{VB} \mid \text{VP, indicated}) \times P_l(\text{STOP} \mid \text{VP, VB, indicated}) \times P_r(\text{NP (difference)} \mid \text{VP, VB, indicated}) \times P_r(\text{PP (in)} \mid \text{VP, VB, indicated}) \times P_r(\text{STOP} \mid \text{VP, VB, indicated})
\]
Independence Assumptions

2.23% 0.06%

60.8% 0.7%
Independence Assumptions

• Also violated in cases of coordination.
  e.g. NP and NP; VP and VP

• Processing facts like attach low in general.

• Also, English parse trees are generally right branching due to SVO structure.

• Language specific features are used heavily in the statistical model for parsing: cf. (Haruno et al. 1999)
Bilexical CFG with probabilistic ‘features’ (Collins 1997)

NP{store} → [ NP{+H} SBAR{+gap} ]
SBAR{+gap} → [ WHNP S{+H}{+C}{+gap} ]
S{+gap} → [ NP{+C} SBAR{+H}{+gap} ]
VP{+gap} → [ VB{+H} TRACE{+C} NP ]
## Statistical Parsing Results using Lexicalized PCFGs

<table>
<thead>
<tr>
<th>System</th>
<th>( \leq 40\text{wds} )</th>
<th>( \leq 100\text{wds} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LP</td>
<td>LR</td>
</tr>
<tr>
<td>(Magerman 95)</td>
<td>84.9</td>
<td>84.6</td>
</tr>
<tr>
<td>(Collins 99)</td>
<td>88.5</td>
<td>88.7</td>
</tr>
<tr>
<td>(Charniak 97)</td>
<td>87.5</td>
<td>87.4</td>
</tr>
<tr>
<td>(Ratnaparkhi 97)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Charniak 99)</td>
<td>90.1</td>
<td>90.1</td>
</tr>
<tr>
<td>(Collins 00)</td>
<td>90.1</td>
<td>90.4</td>
</tr>
<tr>
<td>(LSJ 03)</td>
<td>90.2</td>
<td>90.5</td>
</tr>
<tr>
<td>Voting ( \S )22 (HB99)</td>
<td>92.09</td>
<td>89.18</td>
</tr>
</tbody>
</table>

Voting \( \S \)22 (HB99)

\[
\begin{array}{cccc}
\leq 40\text{wds} & \leq 40\text{wds} & \leq 100\text{wds} & \leq 100\text{wds} \\
LP & LR & LP & LR \\
\end{array}
\]
Tree Adjoining Grammars

- Locality and independence assumptions are captured elegantly.

- Simple and well-defined probability model.

- Parsing can be treated in two steps:
  1. Classification: structured labels (elementary trees) are assigned to each word in the sentence.
  2. Attachment: the elementary trees are connected to each other to form the parse.
Tree Adjoining Grammars: Different Modeling of Bilexical Dependencies
Probabilistic TAGs: Substitution

\[ \sum_{t'} P(t, \eta \rightarrow t') = 1 \]
Probabilistic TAGs: Adjunction

\[ \mathcal{P}(t, \eta \rightarrow NA) + \sum_{t'} \mathcal{P}(t, \eta \rightarrow t') = 1 \]
Tree Adjoining Grammars

- Simpler model for parsing.
  Performance (Chiang 2000): 86.9% LR 86.6% LP ($\leq$ 40 words)
  Latest results: $\approx$ 88% average P/R

- Parsing can be treated in two steps:
  1. Classification: structured labels (elementary trees) are assigned to each word in the sentence.
  2. Attachment: Apply substitution or adjunction to combine the elementary trees to form the parse.
Tree Adjoining Grammars

- Produces more than the phrase structure of each sentence.

- A more embellished parse in which phenomena such as predicate-argument structure, subcategorization and movement are given a probabilistic treatment.
Practical Issues: Beam Thresholding and Priors

- Probability of nonterminal $X$ spanning $j \ldots k$: $N[X, j, k]$

- Beam Thresholding compares $N[X, j, k]$ with every other $Y$ where $N[Y, j, k]$

- But what should be compared?

- Just the **inside probability**: $P(X \xrightarrow{*} t_j \ldots t_k)$? written as $\beta(X, j, k)$

- Perhaps $\beta(\text{FRAG}, 0, 3) > \beta(\text{NP}, 0, 3)$, but NPs are much more likely than FRAGs in general
Practical Issues: Beam Thresholding and Priors

- The correct estimate is the outside probability:

$$P(S \Rightarrow^* t_1 \ldots t_{j-1} X t_{k+1} \ldots t_n)$$

written as $\alpha(X, j, k)$

- Unfortunately, you can only compute $\alpha(X, j, k)$ efficiently after you finish parsing and reach $(S, 0, n)$
Practical Issues: Beam Thresholding and Priors

- To make things easier we multiply the prior probability $P(X)$ with the inside probability.

- In beam Thresholding we compare every new insertion of $X$ for span $j, k$ as follows:
  Compare $P(X) \cdot \beta(X, j, k)$ with every $Y P(Y) \cdot \beta(Y, j, k)$.

- Other more sophisticated methods are given in (Goodman 1997).