CMPT-882: Statistical Learning of Natural Language

Lecture #14

Anoop Sarkar
anoop@cs.sfu.ca
http://www.sfu.ca/~anoop

the company's clinical trials of both its animal and human-based insulins indicated no difference in the level of hypoglycemia between users of either product
Supervised Models for Parsing: History-based models

- Parsing can be framed as a supervised learning task

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

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

- The best parse is then \( \arg \max_{T \in \mathcal{T}(S)} P(T, S) \)
History-based models and PCFGs

- History-based approaches maps \((T, S)\) into a decision sequence \(d_1, \ldots, d_n\)

- Probability of tree \(T\) for sentence \(S\) is:

\[
P(T, S) = \prod_{i=1 \ldots n} P(d_i \mid \phi(d_1, \ldots, d_{i-1}))
\]

- \(\phi\) 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 $\prod_{i=1}^{n} P(\beta_i \mid \gamma_i)$ for a derivation with $n$ rule applications
Generative models and PCFGs

\[ T_{\text{best}} = \arg \max_T P(T \mid S) \]

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

\[ = \arg \max_T P(T, S) \]

\[ = \prod_{i=1}^{n} P(RHS_i \mid LHS_i) \]
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}
Statistical Parsing and PCFGs

The company's trials indicated no difference in the level of hypoglycemia between users of either product.
Bilexical CFG: (Collins 1997)
Bilexical CFG: \( VP\{\text{indicate}\} \rightarrow VB\{+H:\text{indicate}\} \text{ NP\{difference\} PP\{in\}} \)
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) for Japanese
Statistical Parsing Results using Lexicalized PCFGs

<table>
<thead>
<tr>
<th>System</th>
<th>≤ 40wds</th>
<th>≤ 40wds</th>
<th>≤ 100wds</th>
<th>≤ 100wds</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Magerman 95)</td>
<td>84.9</td>
<td>84.6</td>
<td>84.3</td>
<td>84.0</td>
</tr>
<tr>
<td>(Collins 99)</td>
<td>88.5</td>
<td>88.7</td>
<td>88.1</td>
<td>88.3</td>
</tr>
<tr>
<td>(Charniak 97)</td>
<td>87.5</td>
<td>87.4</td>
<td>86.7</td>
<td>86.6</td>
</tr>
<tr>
<td>(Ratnaparkhi 97)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Charniak 99)</td>
<td>90.1</td>
<td>90.1</td>
<td>89.6</td>
<td>89.5</td>
</tr>
<tr>
<td>(Collins 00)</td>
<td>90.1</td>
<td>90.4</td>
<td>89.6</td>
<td>89.9</td>
</tr>
<tr>
<td>Voting (HB99)</td>
<td>92.09</td>
<td>89.18</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

The store bought IBM that last week

Diagram of the sentence structure.
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 (≤ 40 words)
  Latest results: ≈ 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)