Prepositional Phrases

- noun attach: I bought the shirt with pockets
- verb attach: I washed the shirt with soap
- As in the case of other attachment decisions in parsing: it depends on the meaning of the entire sentence – needs world knowledge, etc.
- Maybe there is a simpler solution: we can attempt to solve it using heuristics or associations between words
Structure Based Ambiguity Resolution

- Right association: a constituent (NP or PP) tends to attach to another constituent immediately to its right (Kimball 1973)
- Minimal attachment: a constituent tends to attach to an existing non-terminal using the fewest additional syntactic nodes (Frazier 1978)
- These two principles make opposite predictions for prepositional phrase attachment
- Consider the grammar:

\[
\begin{align*}
VP & \rightarrow V \ NP \ PP \\
NP & \rightarrow NP \ PP
\end{align*}
\]

(1)

(2)

for input: \(I [VP \ saw [NP \ the \ man . . . [PP \ with \ the \ telescope ] , \]
RA predicts that the PP attaches to the NP, i.e. use rule (2), and MA predicts V attachment, i.e. use rule (1)

Structure Based Ambiguity Resolution

- Garden-paths look structural:  
  \(The\ emergency\ crews\ hate\ most\ is\ domestic\ violence\)
- Neither MA or RA account for more than 55% of the cases in real text
- Psycholinguistic experiments using eyetracking show that humans resolve ambiguities as soon as possible in the left to right sequence using the words to disambiguate
- Garden-paths are caused by a combination of lexical and structural effects: 
  \(The\ flowers\ delivered\ for\ the\ patient\ arrived\)
Learning Prepositional Phrase Attachment: Annotated Data

<table>
<thead>
<tr>
<th>v</th>
<th>n1</th>
<th>p</th>
<th>n2</th>
<th>Attachment</th>
</tr>
</thead>
<tbody>
<tr>
<td>join</td>
<td>board</td>
<td>as</td>
<td>director</td>
<td>V</td>
</tr>
<tr>
<td>is</td>
<td>chairman</td>
<td>of</td>
<td>N.V.</td>
<td>N</td>
</tr>
<tr>
<td>using</td>
<td>crocidolite</td>
<td>in</td>
<td>filters</td>
<td>V</td>
</tr>
<tr>
<td>bring</td>
<td>attention</td>
<td>to</td>
<td>problem</td>
<td>V</td>
</tr>
<tr>
<td>is</td>
<td>asbestos</td>
<td>in</td>
<td>products</td>
<td>N</td>
</tr>
<tr>
<td>making</td>
<td>paper</td>
<td>for</td>
<td>filters</td>
<td>N</td>
</tr>
<tr>
<td>including</td>
<td>three</td>
<td>with</td>
<td>cancer</td>
<td>N</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

Prepositional Phrase Attachment

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always noun attachment</td>
<td>59.0</td>
</tr>
<tr>
<td>Most likely for each preposition</td>
<td>72.2</td>
</tr>
<tr>
<td>Average Human (4 head words only)</td>
<td>88.2</td>
</tr>
<tr>
<td>Average Human (whole sentence)</td>
<td>93.2</td>
</tr>
</tbody>
</table>
Let 1 represent noun attachment.

- We want to compute probability of noun attachment: $p(1 \mid v, n_1, p, n_2)$.

- Probability of verb attachment is $1 - p(1 \mid v, n_1, p, n_2)$.

Back-off Smoothing

1. If $f(v, n_1, p, n_2) > 0$ and $\hat{p} \neq 0.5$

   \[ \hat{p}(1 \mid v, n_1, p, n_2) = \frac{f(1, v, n_1, p, n_2)}{f(v, n_1, p, n_2)} \]

2. Else if $f(v, n_1, p) + f(v, p, n_2) + f(n_1, p, n_2) > 0$ and $\hat{p} \neq 0.5$

   \[ \hat{p}(1 \mid v, n_1, p, n_2) = \frac{f(1, v, n_1, p) + f(1, v, p, n_2) + f(1, n_1, p, n_2)}{f(v, n_1, p) + f(v, p, n_2) + f(n_1, p, n_2)} \]

3. Else if $f(v, p) + f(n_1, p) + f(p, n_2) > 0$

   \[ \hat{p}(1 \mid v, n_1, p, n_2) = \frac{f(1, v, p) + f(1, n_1, p) + f(1, p, n_2)}{f(v, p) + f(n_1, p) + f(p, n_2)} \]

4. Else if $f(p) > 0$

   \[ \hat{p}(1 \mid v, n_1, p, n_2) = \frac{f(1, p)}{f(p)} \]

5. Else $\hat{p}(1 \mid v, n_1, p, n_2) = 1.0$
Prepositional Phrase Attachment: (Collins and Brooks 1995)

- **Results**: 84.5% accuracy
  with the use of some limited word classes for dates, numbers, etc.
- Using complex word classes taken from WordNet (which we shall be looking at later in this course) increases accuracy to 88% (Stetina and Nagao 1998)
- We can improve on parsing performance with Probabilistic CFGs by using the insights taken from PP attachment.
- Modify the PCFG model to be sensitive to words and other context-sensitive features of the input.
- And generalizing to other kinds of attachment problems, like coordination or deciding which constituent is an argument of a verb.

Some other studies

- **Toutanova, Manning, and Ng, 2004**: use sophisticated smoothing model for PP attachment
  86.18% with words & stems; with word classes: 87.54%
- **Merlo, Crocker and Berthouzoz, 1997**: test on multiple PPs, generalize disambiguation of 1 PP to 2-3 PPs
  14 structures possible for 3PPs assuming a single verb: all 14 are attested in the Treebank
  same model as CB95; but generalized to dealing with upto 3PPs
  1PP: 84.3%  2PP: 69.6%  3PP: 43.6%
  **Note that this is still not the real problem faced in parsing natural language**
Probability Models

- $p(x, y)$: $x =$ input, $y =$ labels
- Pick best prob distribution $p(x, y)$ to fit the data
- Max likelihood of the data according to the prob model equivalent to minimizing entropy

Max likelihood of the data according to the prob model

- Equivalent to picking best parameter values $\theta$ such that the data gets highest likelihood:

$$\max_\theta p(\theta \mid \text{data}) = \max_\theta p(\theta) \cdot p(\text{data} \mid \theta)$$
Advantages of probability models

- parameters can be estimated automatically, while scores have to twiddled by hand
- parameters can be estimated from supervised or unsupervised data
- probabilities can be used to quantify confidence in a particular state and used to compare against other probabilities in a strictly comparable setting
- modularity: $p(\text{semantics}) \cdot p(\text{syntax} \mid \text{semantics}) \cdot p(\text{morphology} \mid \text{syntax}) \cdot p(\text{phonology} \mid \text{morphology}) \cdot p(\text{sounds} \mid \text{phonology})$

Naive Bayes Classifier

- $x$ is the input that can be represented as $d$ independent features $f_j$, $1 \leq j \leq d$
- $y$ is the output classification
- $P(y \mid x) = \frac{P(y) \cdot P(x \mid y)}{P(x)}$
- $P(x \mid y) = \prod_{j=1}^{d} P(f_j \mid y)$
- $P(y \mid x) = P(y) \cdot \prod_{j=1}^{d} P(f_j \mid y)$
Using Naive Bayes for Document Classification

- Spam text: Learn how to make $38.99 into a money making machine that pays ... $7,000 / month!
- Distinguish spam text from regular email text
- Find useful features to make this distinction

Using Naive Bayes

- Useful features
  1. contains turn $AMOUNT into
  2. contains $AMOUNT
  3. contains Learn how to
  4. contains exclamation mark at end of sentence
Using Naive Bayes

- How many times do these features occur?
  1. contains: turn $AMOUNT$ into
     in spam text: 50
     in normal email: 2
     i.e. 25x more likely in spam
  2. contains: $AMOUNT$
     in spam text: 90
     in normal email: 10
     i.e. 9x more likely in spam

- How likely is it for both features to occur at the same time in a spam message?
  1. contains: turn $AMOUNT$ into
  2. contains: $AMOUNT$

- Assume we have a new feature, contains: turn $AMOUNT$ into and $AMOUNT$

- The model predicts that the event that both features occur simultaneously has probability $\frac{140}{152} = 0.92$

- But Naive Bayes assumes that these features are independent and should occur with probability:
  $0.92 \cdot 0.9 = 0.864$