Unsupervised Tagging?

- AKA part-of-speech induction
- Task:
  - Raw sentences in
  - Tagged sentences out
- Obvious thing to do:
  - Start with a (mostly) uniform HMM
  - Run EM
  - Inspect results

EM for HMMs: Quantities

- Remember from last time:
  \[
  \alpha_i(s) = P(w_0 \ldots w_{i-1}, s_i) = \sum_{s_{i-1}} P(s_i|s_{i-1})P(w_{i-1}|s_{i-1})\alpha_{i-1}(s_{i-1}),
  \]
  \[
  \beta_i(s) = P(w_i \ldots w_n|s_i) = \sum_{s_{i+1}} P(s_{i+1}|s_i)P(w_i|s_i)\beta_{i+1}(s_{i+1}).
  \]
- Can calculate in \(O(s^2n)\) time (why?)

EM for HMMs: Process

- From these quantities, we can re-estimate transitions:
  \[
  \text{count}(s \rightarrow s') = \frac{\sum_i \alpha_i(s)P(s'|s)P(w_i|s)\beta_{i+1}(s')}{P(w)}
  \]
- And emissions:
  \[
  \text{count}(w, s) = \frac{\sum_i w_i=P(w) \alpha_i(s)\beta_{i+1}(s)}{P(w)}
  \]
- If you don’t get these formulas immediately, just think about hard EM instead, where we re-estimate from the Viterbi sequences

Merialdo: Setup

- Some (discouraging) experiments [Merialdo 94]

- Setup:
  - You know the set of allowable tags for each word
  - Fix \(k\) training examples to their true labels
    - Learn \(P(w|t)\) on these examples
    - Learn \(P(t|t-1, t-2)\) on these examples
  - On \(n\) examples, re-estimate with EM

- Note: we know allowed tags but not frequencies
Merialdo: Results

<table>
<thead>
<tr>
<th>Number of tagged sentences used for the initial model</th>
<th>0</th>
<th>100</th>
<th>2000</th>
<th>5000</th>
<th>10000</th>
<th>20000</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iter.</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Correct tags (%) after ML on TM words</td>
<td>77.0</td>
<td>93.0</td>
<td>95.4</td>
<td>96.2</td>
<td>96.6</td>
<td>96.9</td>
<td>97.0</td>
</tr>
<tr>
<td></td>
<td>81.3</td>
<td>92.6</td>
<td>95.8</td>
<td>96.3</td>
<td>96.6</td>
<td>96.7</td>
<td>96.8</td>
</tr>
<tr>
<td></td>
<td>81.8</td>
<td>93.0</td>
<td>95.7</td>
<td>96.1</td>
<td>96.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>83.1</td>
<td>93.1</td>
<td>95.4</td>
<td>95.9</td>
<td>96.2</td>
<td>97.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>84.2</td>
<td>93.0</td>
<td>95.2</td>
<td>95.8</td>
<td>96.0</td>
<td>96.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>84.8</td>
<td>92.9</td>
<td>95.1</td>
<td>95.6</td>
<td>95.8</td>
<td>95.8</td>
<td>95.8</td>
</tr>
<tr>
<td></td>
<td>85.3</td>
<td>92.8</td>
<td>94.9</td>
<td>95.2</td>
<td>95.6</td>
<td>95.6</td>
<td>95.7</td>
</tr>
<tr>
<td></td>
<td>85.8</td>
<td>92.8</td>
<td>94.7</td>
<td>95.1</td>
<td>95.3</td>
<td>95.5</td>
<td>95.5</td>
</tr>
<tr>
<td></td>
<td>86.1</td>
<td>92.7</td>
<td>94.6</td>
<td>95.0</td>
<td>95.2</td>
<td>95.4</td>
<td>95.4</td>
</tr>
<tr>
<td></td>
<td>86.3</td>
<td>92.6</td>
<td>94.5</td>
<td>94.9</td>
<td>95.1</td>
<td>95.3</td>
<td>95.3</td>
</tr>
<tr>
<td></td>
<td>86.6</td>
<td>92.6</td>
<td>94.4</td>
<td>94.8</td>
<td>95.0</td>
<td>95.2</td>
<td>95.2</td>
</tr>
</tbody>
</table>

Distributional Clustering

- Three main variants on the same idea:
  - Pairwise similarities and heuristic clustering
    - E.g. [Finch and Chater 92]
  - Produces dendrograms
  - Vector space methods
    - E.g. [Shuetze 93]
  - Models of ambiguity
  - Probabilistic methods
    - Various formulations, e.g. [Lee and Pereira 99]

Nearest Neighbors

- Various nearest neighbors techniques, e.g. [Lee and Pereira 99]
Vector Space Version

- [Shuetze 93] clusters words as points in $\mathbb{R}^n$
- Vectors too sparse, use SVD to reduce context counts

\[
\begin{align*}
W & \quad M \\
U & \quad \Sigma & \quad V
\end{align*}
\]

Cluster these 50-200 dim vectors instead.

A Probabilistic Version?

\[
P(S, C) = \prod_i P(c_i) P(w_i | c_i) P(w_{i-1}, w_{i+1} | c_i)
\]

- the president said that the downturn was over

What Else?

- Various newer ideas:
  - Context distributional clustering [Clark 00]
  - Morphology-driven models [Clark 03]
  - Contrastive estimation [Smith and Eisner 05]

- Also:
  - What about ambiguous words?
    - Using wider context signatures has been used for learning synonyms (what’s wrong with this approach?)
  - Can extend these ideas for grammar induction (later)