Statistical NLP
Spring 2008

Lecture 8: Word Classes
Dan Klein – UC Berkeley

What’s Next for POS Tagging

- Better features!
- We could fix this by linking capitalized words to their lowercase versions
- Solution: maximum entropy sequence models
- Reality check:
  - Taggers are already pretty good on WSJ journal text…
  - What the world needs is taggers that work on other text!
  - Also: same techniques used for other sequence models (NER, etc)

Decoding

- Decoding maxent taggers:
  - Just like decoding HMMs
  - Viterbi, beam search, posterior decoding
- Viterbi algorithm (HMMs):
  \[ \delta_j(s) = \arg \max_{s'} P(s|s') P(w_i|s) \delta_{j-1}(s') \]
- Viterbi algorithm (Maxent):
  \[ \delta_j(s) = \arg \max_{s'} P(s|s', w, i) \delta_{j-1}(s') \]

Feature Templates

- Important distinction:
  - Features: \(<w_0=\text{future}, t_0=\text{JJ}>\)
  - Feature templates: \(<w_0, t_0>\)
- In maxent taggers:
  - Can now add edge feature templates:
    - \(<t_{i-1}, t_i, t_0>\)
    - Also, mixed feature templates:
      - \(<t_i, w_i, t_j>\)

Maxent Taggers

- MEMMs: use local discriminative models
  \[ P(t|w) = \prod_i \frac{1}{Z} \exp \left( w^T f(t_i, t_{i-1}, t_{i-2}, w, i) \right) \]
  - Train up \( P(t|w, t_{i-1}, t_{i-2}, w) \) as a normal maxent problem, then use to score sequences
  - Referred to as a maxent tagger [Ratnaparkhi 96]
  - Beam search effective! (Why?)
  - What’s the advantage of beam size 1?

HMM Trellis

Intrinsic flaws remained undetected.
They left as soon as he arrived.
TBL Tagger

- [Brill 95] presents a transformation-based tagger
  - Label the training set with most frequent tags
    
    \[
    \text{TBL} \quad \text{MD} \quad \text{VBD} \quad \text{VBD}.
    \]
    
    \(\text{The can was rusted.}\)
  - Add transformation rules which reduce training mistakes
    
    - MD → NN: DT __
    - VBD → VBN: VBD __.
  - Stop when no transformations do sufficient good
  - Does this remind anyone of anything?

- Probably the most widely used tagger (esp. outside NLP)
- … but not the most accurate: 96.6% / 82.0 %

EngCG Tagger

- English constraint grammar tagger
  - [Tapanainen and Voutilainen 94]
  - Something else you should know about
  - "Don’t guess if you know" (general point about modeling more structure!)
  - Tag set doesn’t make all of the hard distinctions as the standard tag set (e.g. JJ/NN)
  - They get stellar accuracies: 98.5% on their tag set
  - Linguistic representation matters…
  - … but it’s easier to win when you make up the rules

CRFs

- Make a maxent model over entire taggings
  - MEMM
    
    \[
    P(t|w) = \prod_i \frac{1}{Z_i(t)} \exp \left( \lambda^\top f(t_i, t_{i-1}, w, i) \right)
    \]
  - CRF
    
    \[
    P(t|w) = \frac{1}{Z(w)} \exp \left( \lambda^\top f(t, w) \right) = \frac{1}{Z(w)} \exp \left( \lambda^\top \sum_i f(t_i, t_{i-1}, w, i) \right) = \frac{1}{Z(w)} \prod_i \phi_i(t_i, t_{i-1})
    \]

CRF Taggers

- Newer, higher-powered discriminative sequence models
  - CRFs (also voted perceptrons, M3Ns)
  - Do not decompose training into independent local regions
  - Can be deathly slow to train – require repeated inference on training set
  - Differences tend not to be too important for POS tagging
  - Differences more substantial on other sequence tasks
  - However: one issue worth knowing about in local models
    - "Label bias" and other explaining away effects
    - Maxent taggers’ local scores can be near one without having both good "transitions" and "emissions"
    - This means that often evidence doesn’t flow properly
    - Why isn’t this a bigger deal for POS tagging?
    - Also: in decoding, condition on predicted, not gold, histories

CRFs

- Like any maxent model, derivative is:
  
  \[
  \frac{\partial L(\lambda)}{\partial \lambda} = \sum_x f(x^t) - \sum_i P(t_i|w) \delta_i(x)
  \]
  
  So all we need is to be able to compute the expectation each feature, for example the number of times the label pair DT-NN occurs, or the number of times NN-interest occurs in a sentence
  
  - How many times does, say, DT-NN occur at position 10? The ratio of the scores of trajectories with that configuration to the score of all
  - This requires exactly the same forward-backward score ratios as for EM, but using the local potentials \(\phi\) instead of the local probabilities
Domain Effects

- Accuracies degrade outside of domain
  - Up to triple error rate
  - Usually make the most errors on the things you care about in the domain (e.g. protein names)

- Open questions
  - How to effectively exploit unlabeled data from a new domain (what could we gain?)
  - How to best incorporate domain lexica in a principled way (e.g. UMLS specialist lexicon, ontologies)

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

Unsupervised Tagging? (cont.)

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?

Unsupervised Tagging?
EM for HMMs: Quantities

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

EM for HMMs: Process

- From these quantities, we can re-estimate transitions:
  \[
  \text{count}(s \rightarrow s') = \frac{\sum \alpha_i(s) P(s' | s) P(w_i | s) \beta_{i+1}(s')}{P(w)}
  \]
- And emissions:
  \[
  \text{count}(w, s) = \frac{\sum \alpha_i(s) \beta_{i+1}(s)}{P(w)}
  \]
- 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>Correct tags (% words) after</td>
<td>97.0</td>
<td>97.0</td>
<td>97.4</td>
<td>97.2</td>
<td>96.6</td>
<td>96.9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>86.6</td>
<td>85.8</td>
<td>95.8</td>
<td>96.3</td>
<td>96.4</td>
<td>96.4</td>
</tr>
<tr>
<td>2</td>
<td>81.8</td>
<td>83.0</td>
<td>95.7</td>
<td>96.1</td>
<td>96.0</td>
<td>96.4</td>
<td>96.4</td>
</tr>
<tr>
<td>3</td>
<td>83.0</td>
<td>93.0</td>
<td>95.4</td>
<td>95.8</td>
<td>96.1</td>
<td>96.2</td>
<td>96.2</td>
</tr>
<tr>
<td>4</td>
<td>84.0</td>
<td>93.0</td>
<td>95.8</td>
<td>95.4</td>
<td>95.5</td>
<td>94.9</td>
<td>96.3</td>
</tr>
<tr>
<td>5</td>
<td>84.8</td>
<td>92.9</td>
<td>95.3</td>
<td>95.4</td>
<td>95.8</td>
<td>95.8</td>
<td>95.8</td>
</tr>
<tr>
<td>6</td>
<td>85.3</td>
<td>92.8</td>
<td>94.9</td>
<td>95.7</td>
<td>95.5</td>
<td>95.8</td>
<td>95.6</td>
</tr>
<tr>
<td>7</td>
<td>85.8</td>
<td>92.8</td>
<td>94.7</td>
<td>95.1</td>
<td>95.7</td>
<td>95.7</td>
<td>95.5</td>
</tr>
<tr>
<td>8</td>
<td>86.1</td>
<td>92.7</td>
<td>94.8</td>
<td>95.0</td>
<td>95.2</td>
<td>95.4</td>
<td>95.4</td>
</tr>
<tr>
<td>9</td>
<td>86.3</td>
<td>92.6</td>
<td>94.9</td>
<td>95.1</td>
<td>95.3</td>
<td>95.3</td>
<td>95.3</td>
</tr>
<tr>
<td>10</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]

[Finch and Chater 92, Shuetze 93, many others]
Nearest Neighbors

Dendrograms

A Probabilistic Version?

$$P(S, C) = \prod_i P(c_i)P(w_i | c_i)P(w_{i+1}, w_{i+2} | c_i)$$

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)