How to build a POS tagger?

- Sources of information:
  - POS tags of surrounding words: syntactic context
  - The word itself
  - Features!
    - Word-internal information
    - External lexicons
    - Features from surrounding words

Diagram:

- HMM
- CRF

Classifier
• **Sequence labeling as structured prediction**

\[ \hat{y}_{1:M} = \arg\max_{y_{1:M} \in \mathcal{Y}(w_{1:M})} \theta^\top f(w_{1:M}, y_{1:M}) \]

• **Hidden Markov model**
  - Fully generative, simple sequence model
  - Supports many operations
    - \( P(w) \): Likelihood (generative model)
      - Forward algorithm
    - \( \arg\max_{y} P(y | w) \): Predicted sequence (“decoding”)
      - Viterbi algorithm
    - \( P(y_m | w) \): Predicted tag marginals
      - Forward-Backward algorithm

• The HMM is a type of log-linear model
HMM as log-linear

- HMM as a joint log-linear model

\[
P(y, w) = \prod_t P(y_t | y_{t-1})P(w_t | y_t)
\]

\[
P(y, w) = \exp(\theta^T f(y, w))
\]

\[
f(y, w) = \sum_t f(y_{t-1}, y_t, w_t)
\]

Local features only!
(Allows efficient inference)

\[
e.g. \{ (N,V), (V,\text{dog}) \}
\]

- The conditional is also log-linear: like we saw before, scoring just “outputs”:

\[
P(y | w) \propto \exp(\theta^T f(y, w))
\]
(Log?-)linear Viterbi

\[ \hat{y} = \arg\max_y \theta^\top f(w, y) \]
\[ f(w, y) = \sum_{m=1}^M f(w, y_m, y_{m-1}, m). \]

\[
\max_y \theta^\top f(w, y) = \max_k v_M(k) \\
\text{Score of best sequence ending in } k \\
\hat{v}_m(k) \triangleq \max_{y_{1:m-1}} \theta^\top f(w, k, y_{m-1}, m) + \sum_{n=1}^{m-1} \theta^\top f(w, y_n, y_{n-1}, n) \\
= \max_{y_{m-1}} \theta^\top f(w, k, y_{m-1}, m) \\
+ \max_{y_{1:m-2}} \theta^\top f(w, y_{m-1}, y_{m-2}) + \sum_{n=1}^{m-2} \theta^\top f(w, y_n, y_{n-1}, n) \\
\hat{v}_{m-1}(y_{m-1})
\]

(a) Weights for emission features.

<table>
<thead>
<tr>
<th></th>
<th>they</th>
<th>can</th>
<th>fish</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>V</td>
<td>-10</td>
<td>-1</td>
<td>-3</td>
</tr>
</tbody>
</table>

(b) Weights for transition features.

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>V</th>
<th>£</th>
</tr>
</thead>
<tbody>
<tr>
<td>◊</td>
<td>-1</td>
<td>-2</td>
<td>-\infty</td>
</tr>
<tr>
<td>N</td>
<td>-3</td>
<td>-1</td>
<td>-12</td>
</tr>
<tr>
<td>V</td>
<td>-1</td>
<td>-3</td>
<td>-1</td>
</tr>
</tbody>
</table>
Forward-Backward

• Purpose: compute
  • Tag marginals $p(y_t | w)$
  • Pair marginals $p(y_{t-1}, y_t | w)$

• Why?
  • Min Bayes Risk decoding
    • For each $t$, choose: $\arg\max_k p(y_t=k | w)$
  • E-step for EM learning of unsupervised HMM
  • Feature expectations for supervised CRF
Learning a CRF

- Gradient descent on the neg. log-likelihood
  - Log-linear gradient: sum over all possible predicted structures
- Non-probabilistic losses: compare gold structure to only one predicted structure
  - Structured perceptron algorithm
  - Structured SVM (hinge loss)
Learning a CRF

$$\log p_\theta(y \mid w) = \theta^T f(y, w) - \log \sum_{y'} \exp(\theta^T f(y, w))$$

$$\frac{\partial \log p_\theta(...)}{\partial \theta_j} = f_j(y, w) - \sum_{y'} p_\theta(y' \mid w) f_j(y', w)$$

- Apply local decomposition
Learning a CRF

\[
\log p_\theta(y \mid w) = \theta^T f(y, w) - \log \sum_{y'} \exp(\theta^T f(y, w))
\]

\[
\frac{\partial \log p_\theta(...)}{\partial \theta_j} = f_j(y, w) - \sum_{y'} p_\theta(y' \mid w) f_j(y', w)
\]

- Apply local decomposition

\[
= \left( \sum_t f_j(y_{t-1}, y_t, w_t) \right) - \sum_{y'} p_\theta(y' \mid w) \sum_t f_j(y'_{t-1}, y'_t, w_t)
\]
Learning a CRF

\[
\log p_\theta(y \mid w) = \theta^T f(y, w) - \log \sum_{y'} \exp(\theta^T f(y, w))
\]

\[
\frac{\partial \log p_\theta(\ldots)}{\partial \theta_j} = f_j(y, w) - \sum_{y'} p_\theta(y' \mid w) f_j(y', w)
\]

• Apply local decomposition

\[
= \left( \sum_t f_j(y_{t-1}, y_t, w_t) \right) - \sum_{y'} p_\theta(y' \mid w) \sum_t f_j(y'_{t-1}, y'_t, w_t)
\]

\[
= \sum_t \left( f_j(y_{t-1}, y_t, w_t) - \sum_{y'_{t-1}, y'_t} p_\theta(y'_{t-1}, y'_t \mid w) f_j(y'_{t-1}, y'_t, w_t) \right)
\]
Learning a CRF

$$\log p_\theta(y \mid w) = \theta^T f(y, w) - \log \sum_{y'} \exp(\theta^T f(y, w))$$

$$\frac{\partial \log p_\theta(...)}{\partial \theta_j} = f_j(y, w) - \sum_{y'} p_\theta(y' \mid w) f_j(y', w)$$

• Apply local decomposition

$$= \left( \sum_t f_j(y_{t-1}, y_t, w_t) \right) - \sum_{y'} p_\theta(y' \mid w) \sum_t f_j(y'_{t-1}, y'_t, w_t)$$

Real feature value

$$= \sum_t \left( f_j(y_{t-1}, y_t, w_t) - \sum_{y'_{t-1}, y'_t} p_\theta(y'_{t-1}, y'_t \mid w) f_j(y'_{t-1}, y'_t, w_t) \right)$$

Expected feature value

Tag marginals (to compute: forward-backward)