COS 484: Natural Language Processing

Sequence Models

Fall 2019
Why model sequences?

Part of Speech tagging

Named Entity recognition

Information Extraction
Overview

- Hidden markov models (HMM)
- Viterbi algorithm
- Maximum entropy markov models (MEMM)
What are POS tags

- Word classes or syntactic categories
  - Reveal useful information about a word (and its neighbors!)

The/DT cat/NN sat/VBD on/IN the/DT mat/NN

Princeton/NNP is/VBZ in/IN New/NNP Jersey/NNP

The/DT old/NN man/VB the/DT boat/NN
Parts of Speech

• Different words have different functions

• Closed class: fixed membership, function words
  
  • e.g. prepositions (in, on, of), determiners (the, a)

• Open class: New words get added frequently
  
  • e.g. nouns (Twitter, Facebook), verbs (google), adjectives, adverbs
Penn Tree Bank tagset

![Penn Tree Bank tagset table]

### Table: Penn Tree Bank Tagset

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Example 1</th>
<th>Tag</th>
<th>Description</th>
<th>Example 2</th>
<th>Tag</th>
<th>Description</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>coordinating conjunction</td>
<td>and, but, or</td>
<td>PDT</td>
<td>predeterminer</td>
<td>all, both</td>
<td>VBP</td>
<td>verb non-3sg</td>
<td>eat</td>
</tr>
<tr>
<td>CD</td>
<td>cardinal number</td>
<td>one, two</td>
<td>POS</td>
<td>possessive ending</td>
<td>'s</td>
<td>VBZ</td>
<td>verb 3sg pres</td>
<td>eats</td>
</tr>
<tr>
<td>DT</td>
<td>determiner</td>
<td>a, the</td>
<td>PRP</td>
<td>personal pronoun</td>
<td>l, you, he</td>
<td>WDT</td>
<td>wh-determin.</td>
<td>which, that</td>
</tr>
<tr>
<td>EX</td>
<td>existential ‘there’</td>
<td>there</td>
<td>PRP$</td>
<td>possess. pronoun</td>
<td>your, one’s</td>
<td>WP $</td>
<td>wh-pronoun</td>
<td>what, who</td>
</tr>
<tr>
<td>FW</td>
<td>foreign word</td>
<td>mea culpa</td>
<td>RB</td>
<td>adverb</td>
<td>quickly</td>
<td>WPS</td>
<td>wh-possess.</td>
<td>whose</td>
</tr>
<tr>
<td>IN</td>
<td>preposition/subordin-conj</td>
<td>of, in, by</td>
<td>RBR</td>
<td>comparative adverb</td>
<td>faster</td>
<td>WRB</td>
<td>wh-adverb</td>
<td>how, where</td>
</tr>
<tr>
<td>JJ</td>
<td>adjective</td>
<td>yellow</td>
<td>RBS</td>
<td>superlatv. adverb</td>
<td>fastest</td>
<td>$</td>
<td>dollar sign</td>
<td>$</td>
</tr>
<tr>
<td>JJR</td>
<td>comparative adj</td>
<td>bigger</td>
<td>RP</td>
<td>particle</td>
<td>up, off</td>
<td>#</td>
<td>pound sign</td>
<td>#</td>
</tr>
<tr>
<td>JJS</td>
<td>superlative adj</td>
<td>wildest</td>
<td>SYM</td>
<td>symbol</td>
<td>+, %, &amp;</td>
<td>''</td>
<td>left quote</td>
<td>‘ or “</td>
</tr>
<tr>
<td>LS</td>
<td>list item marker</td>
<td>1, 2, One</td>
<td>TO</td>
<td>“to”</td>
<td>to</td>
<td>’</td>
<td>right quote</td>
<td>’ or ”</td>
</tr>
<tr>
<td>MD</td>
<td>modal</td>
<td>can, should</td>
<td>UH</td>
<td>interjection</td>
<td>ah, oops</td>
<td>(</td>
<td>left paren</td>
<td>[, (, {, &lt;</td>
</tr>
<tr>
<td>NN</td>
<td>sing or mass noun</td>
<td>llama</td>
<td>VB</td>
<td>verb base form</td>
<td>eat</td>
<td>)</td>
<td>right paren</td>
<td>], ), }, &gt;</td>
</tr>
<tr>
<td>NNS</td>
<td>noun, plural</td>
<td>llamas</td>
<td>VBD</td>
<td>verb past tense</td>
<td>ate</td>
<td>,</td>
<td>comma</td>
<td>,</td>
</tr>
<tr>
<td>NNP</td>
<td>proper noun, sing.</td>
<td>IBM</td>
<td>VBG</td>
<td>verb gerund</td>
<td>eating</td>
<td>.</td>
<td>sent-end punc</td>
<td>! ?</td>
</tr>
<tr>
<td>NNPS</td>
<td>proper noun, plu.</td>
<td>Carolinas</td>
<td>VBN</td>
<td>verb past part.</td>
<td>eaten</td>
<td>:</td>
<td>sent-mid punc</td>
<td>; ... — -</td>
</tr>
</tbody>
</table>

[45 tags]

**Figure 8.1** Penn Treebank part-of-speech tags (including punctuation).

(Marcus et al., 1993)

Other corpora: Brown, WSJ, Switchboard
Part of Speech Tagging

- Disambiguation task: each word might have different senses/functions

- The/DT man/NN bought/VBD a/DT boat/NN

- The/DT old/NN man/VB the/DT boat/NN

<table>
<thead>
<tr>
<th>Types:</th>
<th>WSJ</th>
<th>Brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unambiguous</td>
<td>44,432</td>
<td>45,799</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>7,025</td>
<td>8,050</td>
</tr>
<tr>
<td>Tokens:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unambiguous</td>
<td>577,421</td>
<td>384,349</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>711,780</td>
<td>786,646</td>
</tr>
</tbody>
</table>

Figure 8.2 Tag ambiguity for word types in Brown and WSJ, using Treebank-3 (45-tag) tagging. Punctuation were treated as words, and words were kept in their original case.
Part of Speech Tagging

• Disambiguation task: each word might have different senses/functions

• The/DT man/NN bought/VBD a/DT boat/NN

• The/DT old/NN man/VB the/DT boat/NN

earnings growth took a back/JJ seat
a small building in the back/NN
a clear majority of senators back/VBP the bill
Dave began to back/VB toward the door
enable the country to buy back/RP about debt
I was twenty-one back/RB then

Some words have many functions!
A simple baseline

- Many words might be easy to disambiguate

- **Most frequent class:** Assign each token (word) to the class it occurred most in the training set. (e.g. man/NN)

- Accurately tags **92.34%** of word tokens on Wall Street Journal (WSJ)!

- State of the art ~ 97%

- Average English sentence ~ 14 words

  - Sentence level accuracies: $0.92^{14} = 31\%$ vs $0.97^{14} = 65\%$

- POS tagging not solved yet!
Hidden Markov Models
Some observations

• The function (or POS) of a word depends on its context

  • The/DT old/NN man/VB the/DT boat/NN

  • The/DT old/JJ man/NN bought/VBD the/DT boat/NN

• Certain POS combinations are extremely unlikely

  • <JJ, DT> or <DT, IN>

• Better to make decisions on entire sequences instead of individual words (Sequence modeling!)
Markov chains

- Model probabilities of sequences of variables

- Each state can take one of K values (\{1, 2, ..., K\} for simplicity)

- Markov assumption: \( P(s_t | s_{<t}) \approx P(s_t | s_{t-1}) \)

Where have we seen this before?
The cat sat on the mat.
Markov chains

![Markov chain diagram](image)

The/?? cat/?? sat/?? on/?? the/?? mat/??

- We don’t observe POS tags in corpora
Hidden Markov Model (HMM)

- We don’t observe POS tags in corpora
- But we do observe the words!
- HMM allows us to *jointly reason* over both hidden and observed events.
Components of an HMM

1. Set of states $S = \{1, 2, ..., K\}$ and observations $O$

2. Initial state probability distribution $\pi(s_1)$

3. Transition probabilities $P(s_{t+1} \mid s_t)$

4. Emission probabilities $P(o_t \mid s_t)$
1. Markov assumption:

\[ P(s_{t+1} \mid s_1, \ldots, s_t) = P(s_{t+1} \mid s_t) \]

2. Output independence:

\[ P(o_t \mid s_1, \ldots, s_t) = P(o_t \mid s_t) \]

Which is a stronger assumption?
Sequence likelihood

$p(s, o) = p(s_1, s_2, ... s_n, o_1, o_2, ... o_n)$
Sequence likelihood

\[ P(s, o) = P(s_1, s_2, \ldots, s_n, o_1, o_2, \ldots, o_n) \]

\[ = \prod_{i=1}^n (s_i) P(o_i | s_i) \prod_{i=2}^n P(s_i, o_i | s_{i-1}) \]
\[ P(S, O) = P(s_1, s_2, \ldots, s_n, o_1, o_2, \ldots, o_n) \]

\[ = \prod_{i=1}^n P(s_i | s_{i-1}) \prod_{i=2}^n P(o_i | s_{i-1}) \]

\[ = \pi(s_1) P(o_1 | s_1) \prod_{i=2}^n P(s_i | s_{i-1}) P(o_i | s_i) \]
Learning

Training set:
1 Pierre NNP Vinken NNP , / , 61 CD years NNS old JJ , / , will MD join VB the DT board NN as IN a DT nonexecutive JJ director NN Nov NNP 29 CD . /.
2 Mr. NNP Vinken NNP is VBZ chairman NN of IN Elsevier NNP N V NNP , / , the DT Dutch NNP publishing VBG group NN . /
3 Rudolph NNP Agnew NNP , / , 55 CD years NNS old JJ and CC chairman NN of IN Consolidated NNP Gold NNP Fields NNP PLC NNP , / , was VBD named VBN a DT nonexecutive JJ director NN of IN this DT British JJ industrial JJ conglomerate NN . /

... 38,219 It PRP is VBZ also RB pulling VBG 20 CD people NNS out IN of IN Puerto NNP Rico NNP , / , who WP were VBD helping VBG Hurricane NNP Hugo NNP victims NNS , / , and CC sending VBG them PRP to TO San NNP Francisco NNP instead RB . /.
Learning

Training set:

1 Pierre Vinken, 61 years old, join the board as a non-executive director on Nov. 29.

2 Mr. Vinken is chairman of N.V., the Dutch publisher..

3 Rudolph Agnew, 55 years old, was named a non-executive director of this British industrial conglomerate.

... 38,219 It is also pulling of Puerto Rico, who Hurricane Hugo victims, them to San Francisco.

- Maximum likelihood estimate:

\[ P(s_i | s_j) = \frac{C(s_j, s_i)}{C(s_j)} \]

\[ P(o | s) = \frac{C(s, o)}{C(s)} \]
Example: POS tagging

the/?? cat/?? sat/?? on/?? the/?? mat/??

\[
\pi(DT) = 0.8 \quad s_{t+1}
\]

<table>
<thead>
<tr>
<th></th>
<th>DT</th>
<th>NN</th>
<th>IN</th>
<th>VBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>0.5</td>
<td>0.8</td>
<td>0.05</td>
<td>0.1</td>
</tr>
<tr>
<td>NN</td>
<td>0.05</td>
<td>0.2</td>
<td>0.15</td>
<td>0.6</td>
</tr>
<tr>
<td>IN</td>
<td>0.5</td>
<td>0.2</td>
<td>0.05</td>
<td>0.25</td>
</tr>
<tr>
<td>VBD</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
</tr>
</tbody>
</table>

\[
o_t
\]

<table>
<thead>
<tr>
<th></th>
<th>the</th>
<th>cat</th>
<th>sat</th>
<th>on</th>
<th>mat</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NN</td>
<td>0.01</td>
<td>0.2</td>
<td>0.01</td>
<td>0.01</td>
<td>0.2</td>
</tr>
<tr>
<td>IN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>0</td>
</tr>
<tr>
<td>VBD</td>
<td>0</td>
<td>0.01</td>
<td>0.1</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Example: POS tagging

the/?? cat/?? sat/?? on/?? the/?? mat/??

\[ \pi(DT) = 0.8 \]

\[ s_{t+1} \]

\[ o_t \]

<table>
<thead>
<tr>
<th></th>
<th>DT</th>
<th>NN</th>
<th>IN</th>
<th>VBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>0.5</td>
<td>0.8</td>
<td>0.05</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DT</th>
<th>NN</th>
<th>IN</th>
<th>VBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>0.05</td>
<td>0.2</td>
<td>0.15</td>
<td>0.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DT</th>
<th>NN</th>
<th>IN</th>
<th>VBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>0.5</td>
<td>0.2</td>
<td>0.05</td>
<td>0.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DT</th>
<th>NN</th>
<th>IN</th>
<th>VBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBD</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
</tr>
</tbody>
</table>

\[ P(\text{the}/DT, \text{cat}/NN, \text{sat}/VBD, \text{on}/IN, \text{the}/DT, \text{mat}/NN) \]

\[ = 1.84 \times 10^{-5} \]
Decoding with HMMs

- **Task**: Find the most probable sequence of states \( \langle s_1, s_2, \ldots, s_n \rangle \) given the observations \( \langle o_1, o_2, \ldots, o_n \rangle \)

\[
\hat{S} = \underset{S}{\text{argmax}} \ P(S|O) = \underset{S}{\text{argmax}} \ \frac{P(S) P(O|S)}{P(O)} \quad [\text{Bayes}]
\]
Decoding with HMMs

- **Task:** Find the most probable sequence of states \( \langle s_1, s_2, \ldots, s_n \rangle \) given the observations \( \langle o_1, o_2, \ldots, o_n \rangle \)

\[
\hat{S} = \mathop{\text{argmax}}_{S} P(S | O) = \mathop{\text{argmax}}_{S} \frac{P(O | S) P(S)}{P(O)}
\]

[Bayes]
Decoding with HMMs

- **Task:** Find the most probable sequence of states \( \langle s_1, s_2, \ldots, s_n \rangle \) given the observations \( \langle o_1, o_2, \ldots, o_n \rangle \)

\[
\hat{S} = \arg\max_s \prod_{i=1}^n P(s_i | s_{i-1}) P(o_i | s_i)
\]
Greedy decoding

\[
S = \arg\max_S p(s) p(o_1|s) \\
= \arg\max_S \prod_{i=1}^{n} p(s_i|s_{i-1}) p(o_i|s_i)
\]

\[
\arg\max_S \prod_{i=1}^{n} p(s_i|s_{i-1}) p(o_i|s_i) = 'DT'
\]
Greedy decoding

\[ S = \arg\max_S p(s) p(o|s) \]

\[ = \arg\max_S \prod_{i=1}^{n} p(s_i|s_{i-1}) p(o_i|s_i) \]

Transition

Emission
Greedy decoding

\[ \forall t, \hat{s}_{t+1} = \underset{s}{\text{argmax}} \, p(s | \hat{s}_t) \, p(o_{t+1} | s) \]

- Not guaranteed to be optimal!
- Local decisions
Viterbi decoding

- Use dynamic programming!

- Probability lattice, $M[T, K]$

  - $T$: Number of time steps

  - $K$: Number of states

- $M[i, j]$: Most probable sequence of states ending with state $j$ at time $i$
Viterbi decoding

\[ M[1,DT] = \pi(DT) \ P(\text{the} \mid DT) \]

\[ M[1,NN] = \pi(NN) \ P(\text{the} \mid NN) \]

\[ M[1,VBD] = \pi(VBD) \ P(\text{the} \mid VBD) \]

\[ M[1,IN] = \pi(IN) \ P(\text{the} \mid IN) \]
Viterbi decoding

\[ M[2, DT] = \max_k M[1, k] \ P(DT | k) \ P(\text{cat} | DT) \]

\[ M[2, NN] = \max_k M[1, k] \ P(\text{NN} | k) \ P(\text{cat} | NN) \]

\[ M[2, VBD] = \max_k M[1, k] \ P(\text{VBD} | k) \ P(\text{cat} | VBD) \]

\[ M[2, IN] = \max_k M[1, k] \ P(IN | k) \ P(\text{cat} | IN) \]
Viterbi decoding

\[ M[i, j] = \max_k M[i-1,k] \ P(s_j | s_k) \ P(o_i | s_j) \quad 1 \leq k \leq K \quad 1 \leq i \leq n \]

**Backward:** Pick \( \max_k M[n,k] \) and backtrack
Viterbi decoding

\[ M[i, j] = \max_k M[i - 1, k] P(s_j | s_k) P(o_i | s_j) \quad 1 \leq k \leq K \quad 1 \leq i \leq n \]

**Backward:** Pick \( \max_k M[n, k] \) and backtrack

The cat sat on

\[ M[i, j] = \max_k M[i - 1, k] P(s_j | s_k) P(o_i | s_j) \quad 1 \leq k \leq K \quad 1 \leq i \leq n \]
Beam Search

- If $K$ (number of states) is too large, Viterbi is too expensive!
Beam Search

• If K (number of states) is too large, Viterbi is too expensive!

Many paths have very low likelihood!
Beam Search

• If $K$ (number of states) is too large, Viterbi is too expensive!

• Keep a fixed number of hypotheses at each point
  
  • Beam width, $\beta$
Beam Search

• Keep a fixed number of hypotheses at each point

$\beta = 2$

DT

score = $-4.1$

NN

score = $-9.8$

VBD

score = $-6.7$

IN

score = $-10.1$

The
Beam Search

- Keep a fixed number of hypotheses at each point

\[ \beta = 2 \]

\[ \text{score} = -16.5 \]
\[ \text{score} = -6.5 \]
\[ \text{score} = -13.0 \]
\[ \text{score} = -22.1 \]

Step 1: Expand all partial sequences in current beam
Beam Search

• Keep a fixed number of hypotheses at each point

\[ \beta = 2 \]

\[ \text{score} = -16.5 \]
\[ \text{score} = -6.5 \]
\[ \text{score} = -13.0 \]
\[ \text{score} = -22.1 \]

Step 2: Prune set back to top \( \beta \) sequences
Beam Search

• Keep a fixed number of hypotheses at each point

\[ \beta = 2 \]

Pick \( \max_k M[n, k] \) from within beam and backtrack
Beam Search

• If $K$ (number of states) is too large, Viterbi is too expensive!

• Keep a fixed number of hypotheses at each point
  • Beam width, $\beta$

• Trade-off computation for (some) accuracy

Time complexity?
Beyond bigrams

- Real-world HMM taggers have more relaxed assumptions

- Trigram HMM: \( P(s_{t+1} \mid s_1, s_2, \ldots, s_t) \approx P(s_{t+1} \mid s_{t-1}, s_t) \)
Maximum Entropy Markov Models
Generative vs Discriminative

- HMM is a *generative* model

- Can we model $P(s_1, \ldots, s_n \mid o_1, \ldots, o_n)$ directly?

**Generative**

**Naive Bayes:**

$$P(c)P(d \mid c)$$

**HMM:**

$$P(s_1, \ldots, s_n)P(o_1, \ldots, o_n \mid s_1, \ldots, s_n)$$

**Discriminative**

**Logistic Regression:**

$$P(c \mid d)$$

**MEMM:**

$$P(s_1, \ldots, s_n \mid o_1, \ldots, o_n)$$
MEMM

HMM

MEMM

• Compute the posterior directly:

\[ \hat{S} = \arg \max_S P(S \mid O) = \arg \max_S \prod_i P(s_i \mid o_i, s_{i-1}) \]

• Use features: 

\[ P(s_i \mid o_i, s_{i-1}) \propto \exp(w \cdot f(s_i, o_i, s_{i-1})) \]
In general, we can use all observations and all previous states:

\[
\hat{S} = \arg \max_S P(S \mid O) = \arg \max_S \prod_i P(s_i \mid o_n, o_{i-1}, \ldots, o_1, s_{i-1}, \ldots, s_1)
\]

\[
P(s_i \mid s_{i-1}, \ldots, s_1, O) \propto \exp(w \cdot f(s_i, s_{i-1}, \ldots, s_1, O))
\]
Features in an MEMM

Figure 8.13 An MEMM for part-of-speech tagging showing the ability to condition on more features.

\[
\begin{align*}
\langle t_i, w_{i-2} \rangle, & \langle t_i, w_{i-1} \rangle, \langle t_i, w_i \rangle, \langle t_i, w_{i+1} \rangle, \langle t_i, w_{i+2} \rangle \\
\langle t_i, t_{i-1} \rangle, & \langle t_i, t_{i-2}, t_{i-1} \rangle, \\
\langle t_i, t_{i-1}, w_i \rangle, & \langle t_i, w_{i-1}, w_i \rangle \langle t_i, w_i, w_{i+1} \rangle,
\end{align*}
\]

- \( t_i = \) VB and \( w_{i-2} = \) Janet
- \( t_i = \) VB and \( w_{i-1} = \) will
- \( t_i = \) VB and \( w_i = \) back
- \( t_i = \) VB and \( w_{i+1} = \) the
- \( t_i = \) VB and \( w_{i+2} = \) bill
- \( t_i = \) VB and \( t_{i-1} = \) MD
- \( t_i = \) VB and \( t_{i-1} = \) MD and \( t_{i-2} = \) NNP
- \( t_i = \) VB and \( w_i = \) back and \( w_{i+1} = \) the

Feature templates
MEMMs: Decoding

\[ \hat{S} = \arg \max_S P(S \mid O) = \arg \max_S \prod_i P(s_i \mid o_i, s_{i-1}) \]

(assume features only on previous time step and current obs)

- Greedy decoding:

\[ \hat{s}_i = \arg \max_s P(s \mid \text{The}) \]

\[ = \text{DT} \]
MEMMs: Decoding

\[ \hat{S} = \arg \max_S P(S \mid O) = \arg \max_S \prod_i P(s_i \mid o_i, s_{i-1}) \]

- Greedy decoding:

\[ \hat{S}_2 = \arg \max_S P(S \mid \text{cat}, \text{DT}) = \text{NN} \]
MEMMs: Decoding

\[ \hat{S} = \arg \max_s P(S \mid O) = \arg \max_s \Pi_i P(s_i \mid o_i, s_{i-1}) \]

• Greedy decoding:

\[ \forall t, \quad \hat{s}_{t+1} = \arg \max_s P(s \mid o_{t+1}, s_t) \]
MEMMs: Decoding

\[ \hat{S} = \arg \max_S P(S \mid O) = \arg \max_S \prod_i P(s_i \mid o_i, s_{i-1}) \]

- Greedy decoding
- Viterbi decoding:

\[ M[i, j] = \max_k M[i - 1, k] P(s_j \mid o_i, s_k) \quad 1 \leq k \leq K \quad 1 \leq i \leq n \]
MEMM: Learning

• Gradient descent: similar to logistic regression!

\[ P(s_i | s_1, \ldots, s_{i-1}, O) \propto \exp(w \cdot f(s_1, \ldots, s_i, O)) \]

• Given: pairs of \((S, O)\) where each \(S = \langle s_1, s_2, \ldots, s_n \rangle\)

Loss for one sequence, \(L = - \sum_i \log P(s_i | s_1, \ldots, s_{i-1}, O)\)

• Compute gradients with respect to weights \(w\) and update
Both HMM and MEMM assume left-to-right processing

Why can this be undesirable?
Bidirectionality

The old man the boat

P(JJ | DT) \(\boxed{P(\text{old} | JJ)}\) P(NN | JJ) \(\boxed{P(\text{man} | NN)}\) P(DT | NN)

P(NN | DT) \(\boxed{P(\text{old} | NN)}\) P(VB | NN) \(\boxed{P(\text{man} | VB)}\) P(DT | VB)

Observation bias
Observation bias
Conditional Random Field (advanced)

- Compute log-linear functions over cliques
- Lesser independence assumptions
- Ex: $P(s_t | \text{everything else}) \propto \exp(w \cdot f(s_{t-1}, s_t, s_{t+1}, O))$