Parts of Speech (POGs)

Part of speech is a formal property of word-types that determines their acceptable uses in syntax.

Parts of speech (syntactic categories) can be regarded as classes of words. Examples:

- nouns
- verbs
- adjectives
- adverbs

POS does not define how a word participates in the semantic interpretation of a sentence (although not entirely independent).

A word-type can have more than one POS, but a word-token has exactly one, e.g.:

I can_{\text{Aux}} kick the can_{\text{N}}.
Tagging: Assigning Parts of Speech

POS Tagging is a first step towards

• classification (POS tag can be feature)
• finding meaning of word
• parsing a sentence
• partial parsing, e.g., noun-phrase detection
Sources of Knowledge about POS

<table>
<thead>
<tr>
<th>Input: The red ducks can run down steep banks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Det</td>
</tr>
<tr>
<td>—</td>
</tr>
<tr>
<td>—</td>
</tr>
<tr>
<td>—</td>
</tr>
<tr>
<td>—</td>
</tr>
<tr>
<td>True: Det</td>
</tr>
</tbody>
</table>

*Syntagmatic statistics* (horizontal): how likely is a sequence of tags?  
*Paradigmatic statistics* (vertical): how likely is a given word to have one tag vs. another?
Sources of Knowledge about POS

<table>
<thead>
<tr>
<th>Input:</th>
<th>The</th>
<th>red</th>
<th>ducks</th>
<th>can</th>
<th>run</th>
<th>down</th>
<th>steep</th>
<th>banks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Det</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>Adj</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>(Adj)</td>
<td>Adj</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>Noun</td>
<td>Noun</td>
<td>Noun</td>
<td>Noun</td>
<td>Noun</td>
<td>Noun</td>
<td>—</td>
<td>Noun</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>Verb</td>
<td>Verb</td>
<td>Verb</td>
<td>Verb</td>
<td>Verb</td>
<td>Verb</td>
<td>—</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>Prep</td>
<td>—</td>
</tr>
<tr>
<td>True:</td>
<td>Det</td>
<td>Adj</td>
<td>Noun</td>
<td>Verb</td>
<td>Verb</td>
<td>Prep</td>
<td>Adj</td>
<td>Noun</td>
</tr>
</tbody>
</table>

*Syntagmatic statistics* (horizontal): how likely is a sequence of tags?

*Paradigmatic statistics* (vertical): how likely is a given word to have one tag vs. another?

Syntagmatic looks useful, but isn’t: $\approx 77\%$ accuracy.
Sources of Knowledge about POS

Input: The red ducks can run down steep banks

Det — — — — — — — —
— Adj — — — — (Adj) Adj —
— Noun Noun Noun Noun Noun Noun — Noun
— — Verb Verb Verb Verb Verb Verb Verb
— — — — — — — Prep — —

True: Det Adj Noun Verb Verb Prep Adj Noun

*Syntagmatic statistics* (horizontal): how likely is a sequence of tags?

*Paradigmatic statistics* (vertical): how likely is a given word to have one tag vs. another?

Paradigmatic is very useful: as high as $\approx 90\%$ accuracy.

Use both: as high as $\approx 95\%$.

Warning: these are per-word accuracies.
How do we combine these sources of knowledge?

\[
\arg\max_{t_1 \ldots t_n} P(t_1 \ldots t_n \mid w_1 \ldots w_n)
\]

\[
= \arg\max_{t_1 \ldots t_n} \frac{P(w_1 \ldots w_n \mid t_1 \ldots t_n) P(t_1 \ldots t_n)}{P(w_1 \ldots w_n)}
\]

\[
= \arg\max_{t_1 \ldots t_n} P(w_1 \ldots w_n \mid t_1 \ldots t_n) P(t_1 \ldots t_n)
\]

\[
= \arg\max_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_1 \ldots t_n) P(t_1 \ldots t_n)
\]

\[
= \arg\max_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_n \mid t_{n-1}) \ldots P(t_2 \mid t_1) P(t_1)
\]

\[
= \arg\max_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_i \mid t_{i-1})
\]

 \[ [P(t_1 \mid t_0) \equiv P(t_1)] \]
With an HMM!

\[
\begin{align*}
\arg\max_{t_1 \ldots t_n} P(t_1 \ldots t_n \mid w_1 \ldots w_n) \\
\doteq \arg\max_{t_1 \ldots t_n} \frac{P(w_1 \ldots w_n \mid t_1 \ldots t_n)P(t_1 \ldots t_n)}{P(w_1 \ldots w_n)} \\
= \arg\max_{t_1 \ldots t_n} P(w_1 \ldots w_n \mid t_1 \ldots t_n)P(t_1 \ldots t_n) \\
\doteq \arg\max_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_1 \ldots t_n)P(t_1 \ldots t_n) \\
\doteq \arg\max_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i)P(t_1 \ldots t_n) \\
= \arg\max_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i)P(t_n \mid t_{n-1}) \ldots P(t_2 \mid t_1)P(t_1) \\
= \arg\max_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i)P(t_i \mid t_{i-1}) \\
\quad \quad \quad \quad \quad \quad [P(t_1 \mid t_0) \equiv P(t_1)]
\end{align*}
\]

Use tags as states, words as output symbols
\(P(w_i \mid t_i)\): emission probabilities \((B)\)

\(P(t_i \mid t_{i-1})\): transition probabilities \((A)\)

\(P(t_1)\): initial probabilities \((\pi)\)
Setting parameters of the HMM

\[
P(t^k \mid t^j) = \frac{C(t^j t^k)}{C(t^j)}
\]

\[
P(w^k \mid t^j) = \frac{C(t^j, w^k)}{C(t^j)}
\]

- Counts are generally determined from a manually tagged corpus.
- If training data are sampled from the same domain as the test data, then Baum-Welch is likely to hurt performance.
- If training data are sampled from a different domain, then a few iterations of Baum-Welch might help.

Conditionalizing the probability of a tag on preceding word is much harder to train

Alternative: “transformation-based” tagger - make an imperfect tagging, then correct using (learned) transformational rules.
Dealing with Unknown Words

Three kinds:

1. training word not in lexicon
2. training word in lexicon, but not in corpus
3. test word unknown

Solutions:

• heuristic rules (1,3), e.g., capitalization (noun), morphology (-ing,-ed is probably verb)
• parameter tying using “meta-words” (2): classes with same POS alternations, e.g., \{can, run, ducks, banks\} can all be nouns or verbs.
The Brill Tagger

Transformation-based

Transformation rule: \( t^i \longrightarrow t^j \) when \( X \)

9 kinds of \( X \)

Examples:

- **NN** \( \longrightarrow \) **VB** when \( t_{i-2} = \text{Det} \& w_{i+1} = \text{n’t} \) (9)

- **NN** \( \longrightarrow \) **VB** when \( t_{i-2} = \text{NN} \) or \( t_{i-1} = \text{NN} \) (3)

Unknown words:

1. capitalized \( \Rightarrow \) **NNP** (proper)

2. otherwise **NN** (common)

3. Then apply morphological transformations, e.g.:
   - **NN** \( \longrightarrow \) **NNS** if suffix is -s
Then what do we learn?

The order of the transformations:

1. $C_0 := \text{initially tagged corpus (e.g., paradigmatic info only)}$

2. for $k := 0$ step 1 do

   - $v := \text{argmin}_\tilde{v} E(\tilde{v}(C_k))$
   - if $[E(C'_k) - E(v(C_k)))] < \epsilon$ then break
   - $C'_{k+1} := v(C_k)$
   - $\tau_{k+1} := v$
Why does order matter?

Depends on the style of transformational system:

Example: \( A \rightarrow B \) if \( t_{i-1} = A \).

Input:AAAA

<table>
<thead>
<tr>
<th>Effect/Direction</th>
<th>left-to-right</th>
<th>right-to-left</th>
</tr>
</thead>
<tbody>
<tr>
<td>immediate</td>
<td>ABAB</td>
<td>ABBB</td>
</tr>
<tr>
<td>delayed</td>
<td>ABBB</td>
<td>ABBB</td>
</tr>
</tbody>
</table>

Brill tagger uses a delayed-effect, left-to-right system.