Parts of Speech (POSs)

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.:

$$\text{I can}_{\text{Aux}} \text{ kick the can}_{\text{N}}.$$

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

Input:	The	red	ducks	can	run	down	steep	banks
	Det	—	—	—	—	—	—	—
	—	Adj	—	—	—	(Adj)	Adj	—
	—	Noun	Noun	Noun	Noun	Noun	—	Noun
	—	—	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?

Sources of Knowledge about POS

Input:	The	red	ducks	can	run	down	steep	banks
	Det	—	—	—	—	—	—	—
	—	Adj	—	—	—	(Adj)	Adj	—
	—	Noun	Noun	Noun	Noun	Noun	—	Noun
	—	—	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?

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
	—	—	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?

$$
\begin{aligned}
&\operatorname*{argmax}_{t_1 \ldots t_n} P(t_1 \ldots t_n \mid w_1 \ldots w_n) \\
&\doteq \operatorname*{argmax}_{t_1 \ldots t_n} \frac{P(w_1 \ldots w_n | t_1 \ldots t_n) P(t_1 \ldots t_n)}{P(w_1 \ldots w_n)} \\
&= \operatorname*{argmax}_{t_1 \ldots t_n} P(w_1 \ldots w_n \mid t_1 \ldots t_n) P(t_1 \ldots t_n) \\
&\doteq \operatorname*{argmax}_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_1 \ldots t_n) P(t_1 \ldots t_n) \\
&\doteq \operatorname*{argmax}_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_1 \ldots t_n) \\
&\doteq \operatorname*{argmax}_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_n \mid t_{n-1}) \ldots P(t_2 \mid t_1) P(t_1) \\
&= \operatorname*{argmax}_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_i \mid t_{i-1})
\end{aligned}
$$

$$[P(t_1 | t_0) \equiv P(t_1)]$$

With an HMM!

$$
\begin{aligned}
& \operatorname*{argmax}_{t_1 \ldots t_n} P(t_1 \ldots t_n \mid w_1 \ldots w_n) \\
& \doteq \operatorname*{argmax}_{t_1 \ldots t_n} \frac{P(w_1 \ldots w_n | t_1 \ldots t_n) P(t_1 \ldots t_n)}{P(w_1 \ldots w_n)} \\
& = \operatorname*{argmax}_{t_1 \ldots t_n} P(w_1 \ldots w_n \mid t_1 \ldots t_n) P(t_1 \ldots t_n) \\
& \doteq \operatorname*{argmax}_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_1 \ldots t_n) P(t_1 \ldots t_n) \\
& \doteq \operatorname*{argmax}_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_1 \ldots t_n) \\
& \doteq \operatorname*{argmax}_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_n \mid t_{n-1}) \ldots P(t_2 \mid t_1) P(t_1) \\
& = \operatorname*{argmax}_{t_1 \ldots t_n} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_i \mid t_{i-1}) \\
& \qquad\qquad [P(t_1 | t_0) \equiv P(t_1)]
\end{aligned}
$$

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 (π)

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 $\mathbf{X}$
9 kinds of $\mathbf{X}$

Examples:

- NN $\longrightarrow$ VB when t_{i-2} = Det & w_{i+1} = n't (9)
- NN $\longrightarrow$ VB when t_{i-2} = NN or t_{i-1} = 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 := initially tagged corpus (e.g., paradigmatic info only)
2. for $k := 0$ step 1 do
   - $v := \mathrm{argmin}_{\bar{v}}\, E(\bar{v}(C_k))$
   - 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 $\longrightarrow$ B if $t_{i-1} = \mathrm{A}$.
Input: AAAA

Effect/Direction	left-to-right	right-to-left
immediate	ABAB	ABBB
delayed	ABBB	ABBB

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