

---

## Statistical NLP: Lecture 12

### Part-of-Speech Tagging

(Ch 10)

---

## Part-of-speech tagging

- Part-of-speech tagging, PoS tagging: Assigning a part-of-speech category to each word-token in a text.

**The red ducks can run down steep banks.**

Det	-	-	-	-	-	-	-	-
-	noun	noun	noun	noun	noun	-	-	noun
-	-	verb	verb	verb	verb	verb	verb	verb
-	adj	-	-	-	adj	adj	-	-
-	-	-	-	-	prep	-	-	-

---

## Tagsets

- Need to agree on exactly what the possible PoS tags are: the tagset.
- Simple part-of-speech categories are not sufficient to describe language.
- Make as many helpful distinctions as possible.
- Major English tagsets: Penn (45 tags); Brown (87 tags); Lancaster: CLAWS series of tagsets, C5, and C7 (for BNC, 146 tags).

---

## The Penn tagset (for words)

CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	<i>there</i> is
FW	foreign word	d'hoevre
IN	preposition/subordinating conjunction	in, of, like
JJ	adjective	green
JJR	adjective, comparative	greener
JJS	adjective, superlative	greenest
LS	list marker	1)
MD	modal	could, will

---

NN	noun, singular or mass	table
NNS	noun plural	tables
NNP	proper noun, singular	John
NNPS	proper noun, plural	Vikings
PDT	predeterminer	<i>both</i> the boys
POS	possessive ending	friend's
PRP	personal pronoun	I, he, it
PRP\$	possessive pronoun	my, his
RB	adverb	however, usually, naturally, here, good
RBR	adverb, comparative	better
RBS	adverb, superlative	best

---

RP	particle	<i>give up</i>
TO	to	<i>to go, to him</i>
UH	interjection	uhhuhhuhh
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, gerund/present participle	taking
VBN	verb, past participle	taken
VBP	verb, sing. present, non-3d	take
VBZ	verb, 3rd person sing. present	takes
WDT	wh-determiner	which
WP	wh-pronoun	who, what
WP\$	possessive wh-pronoun	whose
WRB	wh-abverb	where, when

## Penn tagset (for punctuation)

---

- # Pound sign
- \$ Dollar sign
- . Sentence-final punctuation
- , Comma
- : Colon, semi-colon, ellipsis
- ( Left bracket character
- ) Right bracket character
- ” Straight double quote
- ‘ Left open single quote
- “ Left open double quote
- ’ Right close single quote
- ” Right close double quote

## Lexical resources for tagging

---

- Possible starting points:
  - Lexicon giving possible PoS’s for each word.
  - Hand-tagged corpus.
- Lexicon could include a priori frequency information.

## What counts as good?

---

- Dumb and easy tagging method: Give each word its most frequent tag.
- Accuracy of this method: about 90%
- But we want 99% or better.

## Rule-based tagging

---

- Dependencies between tags: i.e., DT + VB not allowed; agreement subject verb.
- Rule-based tagging: Use set of rules for what tags can and can’t follow or precede other tags.
  - “If this word’s possible tags include both verb tags and non-verb tags, and the previous word is a preposition, then eliminate the verb tags from consideration.”
  - ... *pieces of clean rag* ...
- Very specific rules; problems with long-distance dependencies.

## Tag bigrams

---

- Choose the most likely tag for the current word given the previous word and tag

$$t_i = \operatorname{argmax}_j P(\bar{t}^j | t_{i-1}, w_i) = \operatorname{argmax}_j P(\bar{t}^j | t_{i-1})P(w_i | \bar{t}^j)$$

- Get the probabilities from a tagged corpus.
- For all words  $w_k$  and tags  $t^k$ ,  
 $P(t^k | \bar{t}) = C(\bar{t}, t^k) / C(\bar{t})$   
 $P(w_k | \bar{t}) = C(w_k, \bar{t}) / C(\bar{t})$

## Markov models for tagging

---

- Markov model: Tags are states; words are output.
  - Find most probable sequence of states (tags) for observed output (sentence to be tagged).
- Parameters of the model determined from pre-tagged corpus—same as for bigram model.
  - Train as visible model.
- Use Viterbi algorithm to find most likely tag sequence.
  - Use as hidden model.

## Unknown words

---

- In training: word is in lexicon but not in training data, or vice versa.
- In use: word is completely unknown to tagger.
- Heuristics: Unknown capitalized word likely to be NNP; *-ing* likely to indicate VBG; etc.
- Smoothing for possible events not in training data.

## Completely hidden models

---

- Training without a tagged corpus.
  - For other genres or languages.
- Use Baum-Welch algorithm to estimate parameters.
- Various possible starting guesses for tag probabilities.

## Transformation-based-learning taggers I

---

- Brill tagger: Starts off with dumb method; then applies ordered sequence of patches to correct the errors.
- Patches are transformations learned from correctly-tagged corpus.
- Pattern-action rules: match present tag and context, change present tag.
  - Change NN to VB if previous tag is TO.
  - Change JJ to NNP if next tag is NNP.

## Transformation-based-learning taggers II

---

- Training method:
  - Get count of each error type for present corpus by comparing with training corpus.
  - Try all applicable transformations; for each, count number of errors removed and errors added.
  - Add the transformation that made the most improvement to the sequence of patches.
- Repeat until no more improvement.

## Transformation-based-learning taggers III - Improvements

---

- Allow lexicalized transformations.
  - Change IN to RB if the second word to the right is *as*.
  - Change VBP to VB if one of the previous two words is *n't*.
- Learn best treatment of unknown words.
  - Start as NNP or NN. Change NN to NNS if last character is *s*.
  - Change anything to JJ if adding *ly* would make a known word.
  - Change NN to VB if the word *would* ever appears to the left.
- Results: Over 97% accuracy; lexicalized transformations don't help much; resistant to overfitting.

## Applications of tagging

---

- Identifying features in text for classification.
- First step in finding the meaning of a word.
- First step in partial or complete parsing.
- Noun-phrase and named-entity identification.

## Noun phrases and named entities

---

- Non-recursive noun phrases, including named entities. Mark the NPs:
  - *The dogs on the other ends of these locking bars are thus forced into notches in other tappets.*
  - *General Electric Co. agreed to a minority role in a mobile communications joint venture with Telefon AB L.M. Ericsson of Sweden.*

## Noun-phrase detection

---

- Detect NPs by looking at sequences of tags.
- Rule for simple NPs: [DT] + JJ\* + NN | NNS
- But in practice, it's more complex: proper names, sequences of nouns:  
*General/NNP Electric/NNP Co./NNP agreed/VBD to/TO a/DT minority/NN role/NN in/IN a/DT mobile/JJ communications/NNS joint/JJ venture/NN with/IN Telefon/NNP AB/NNP L.M./NNP Ericsson/NNP of/IN Sweden/NNP.*
- So use HMM, learn from tagged corpus with NPs marked.

## HMMs for noun-phrase detection

---

- Noun-phrase detection as scanning text, inserting markers <NP> and </NP>
- HMM for this:
  - Five states: <NP>, </NP>, </NP> <NP>, in-NP, not-in-NP
  - Output is pairs (bigrams) of tags: e.g., DT-NNS, NNS-IN, IN-DT, DT-JJ,
- To find noun phrases in a sentence, represent it as sequence of tag bigrams, and use Viterbi algorithm to find sequence of states that HMM went through to generate it.
- Train the HMM by Baum–Welch algorithm.