# POS Tagging

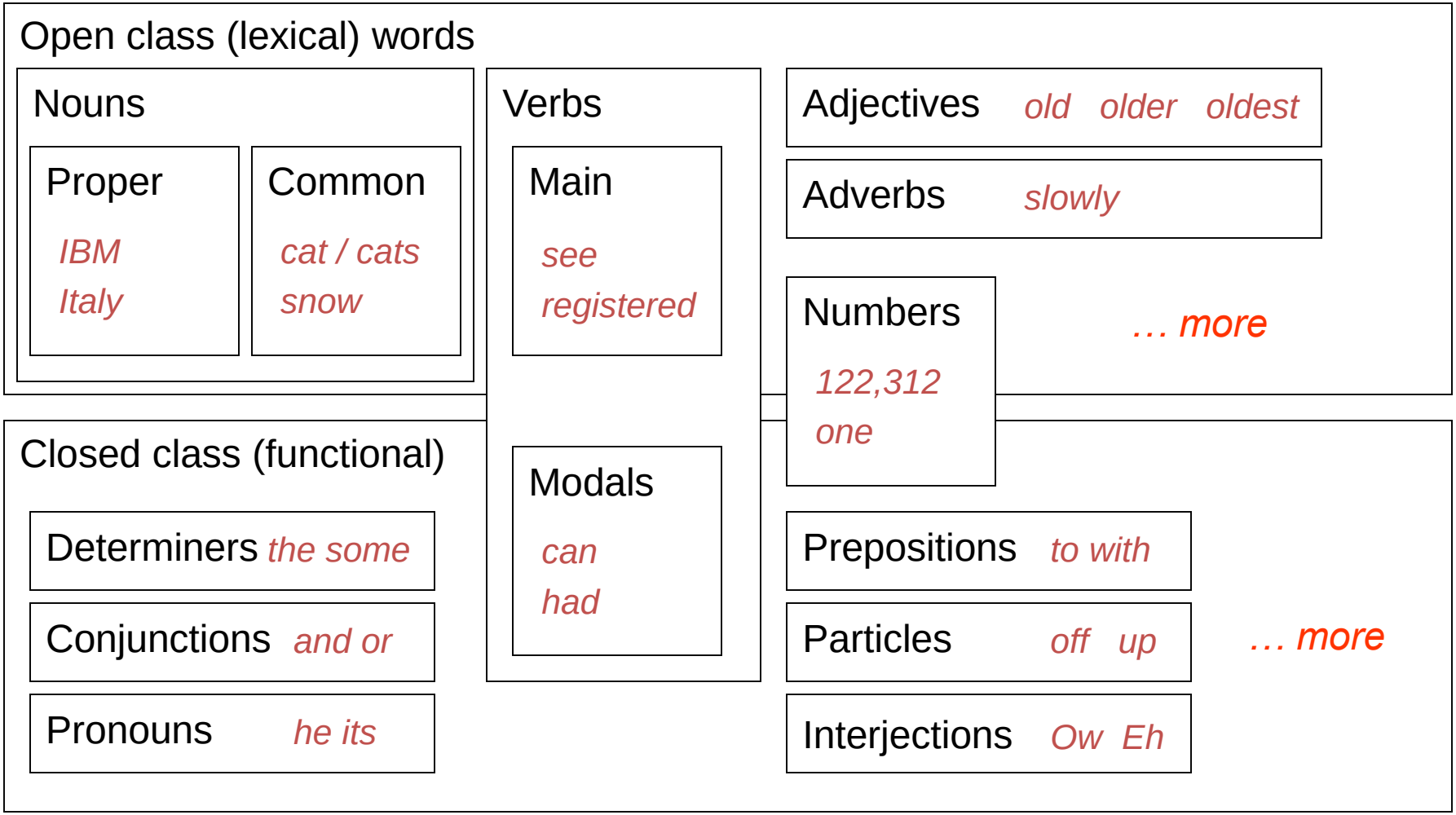
(materials mostly from Jurafsky & Martin(2009))

# Parts of Speech

- Perhaps starting with Aristotle in the West (384–322 BCE), there was the idea of having parts of speech
  - a.k.a lexical categories, word classes, "tags", POS
- It comes from Dionysius Thrax of Alexandria (c. 100 BCE) the idea that is still with us that there are 8 parts of speech

# Parts of Speech

- But actually his 8 aren't exactly the ones we are taught today
  - Thrax: noun, verb, article, adverb, preposition, conjunction, participle, pronoun
  - School grammar: noun, verb, adjective, adverb, preposition, conjunction, pronoun, interjection


Open class (lexical) words
Nouns
Proper
IBM
Italy
Common
cat / cats
snow
Verbs
Main
see
registered
Modals
can
had
Adjectives old older oldest
Adverbs slowly
Numbers
122,312
one
… more
Closed class (functional)
Determiners the some
Conjunctions and or
Pronouns he its
Prepositions to with
Particles off up
Interjections Ow Eh
… more

# Open vs. Closed classes

- Open vs. Closed classes
  - Closed:
    - determiners: ***a, an, the***
    - pronouns: ***she, he, I***
    - prepositions: ***on, under, over, near, by, …***
    - Why “closed”?
  - Open:
    - Nouns, Verbs, Adjectives, Adverbs.

# Penn Tagsets

| TAG | DESCRIPTION | EXAMPLE |
|---|---|---|
| CC | conjunction, coordinating | *and, or, but* |
| CD | cardinal number | *five, three, 13%* |
| DT | determiner | *the, a, these* |
| EX | existential there | *there were six boys* |
| FW | foreign word | *mais* |
| IN | conjunction, subordinating or preposition | *of, on, before, unless* |
| JJ | adjective | *nice, easy* |
| JJR | adjective, comparative | *nicer, easier* |
| JJS | adjective, superlative | *nicest, easiest* |

| | | |
|---|---|---|
| LS | list item marker | |
| MD | verb, modal auxillary | *may, should* |
| NN | noun, singular or mass | *tiger, chair, laughter* |
| NNS | noun, plural | *tigers, chairs, insects* |
| NNP | noun, proper singular | *Germany, God, Alice* |
| NNPS | noun, proper plural | *we met two Christmases ago* |
| PDT | predeterminer | *both his children* |
| POS | possessive ending | *'s* |
| PRP | pronoun, personal | *me, you, it* |
| PRP$ | pronoun, possessive | *my, your, our* |

| | | |
|---|---|---|
| RB | adverb | *extremely, loudly, hard* |
| RBR | adverb, comparative | *better* |
| RBS | adverb, superlative | *best* |
| RP | adverb, particle | *about, off, up* |
| SYM | symbol | *%* |
| TO | infinitival to | *what to do?* |
| UH | interjection | *oh, oops, gosh* |

| | | |
|---|---|---|
| VB | verb, base form | *think* |
| VBZ | verb, 3rd person singular present | *she thinks* |
| VBP | verb, non-3rd person singular present | *I think* |
| VBD | verb, past tense | *they thought* |
| VBN | verb, past participle | *a sunken ship* |
| VBG | verb, gerund or present participle | *thinking is fun* |
| WDT | *wh*-determiner | *which, whatever, whichever* |
| WP | *wh*-pronoun, personal | *what, who, whom* |
| WP$ | *wh*-pronoun, possessive | *whose, whosever* |
| WRB | *wh*-adverb | *where, when* |

| | | |
|---|---|---|
| . | punctuation mark, sentence closer | .;?* |
| , | punctuation mark, com ma | , |
| : | punctuation mark, colon | : |
| ( | contextual separator, left paren | ( |
| ) | contextual separator, rig ht paren | ) |

# POS Tagging

- Words often have more than one POS: *back*
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB
- The POS tagging problem is to determine the POS tag for a particular instance of a word.

# POS Tagging

- Input: Plays well with others
- Ambiguity: NNS/VBZ UH/JJ/NN/RB IN NNS
- Output: Plays/VBZ well/RB with/IN others/NNS
- Uses:
  - Text-to-speech (how do we pronounce “lead”?)
  - Can write regexps like (Det) Adj* N+ over the output for phrases, etc.
  - As input to or to speed up a full parser
  - If you know the tag, you can back off to it in other tasks

# POS tagging performance

- How many tags are correct? (Tag accuracy)
  - About 97% currently
  - But baseline is already 90%
    - Baseline is performance of stupidest possible method
      - Tag every word with its most frequent tag
      - Tag unknown words as nouns
  - Partly easy because
    - Many words are unambiguous
    - You get points for them (*the*, *a*, etc.) and for punctuation marks!

# Deciding on the correct part of speech can be difficult even for people

Mrs/NNP Shaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG

All/DT we/PRP gotta/VBN do/VB is/VBZ go/VB around/IN the/DT corner/NN

Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

# How difficult is POS tagging?

- About 11% of the word types in the Brown corpus are ambiguous with regard to part of speech
- But they tend to be very common words. E.g., *that*
  - I know *that* he is honest = IN
  - Yes, *that* play was nice = DT
  - You can't go *that* far = RB
- 40% of the word tokens are ambiguous

# Sources of information

- What are the main sources of information for POS tagging?
  - Knowledge of neighboring words
    - Bill saw that man yesterday
    - NNP NN DT NN NN
    - VB VB(D) IN VB NN
  - Knowledge of word probabilities
    - *man* is rarely used as a verb….
- The latter proves the most useful, but the former also helps

# More and Better Features ➔ Feature-based tagger

- Can do surprisingly well just looking at a word by itself:
  - Word: the: the → DT
  - Lowercased word: Importantly: importantly → RB
  - Prefixes: unfathomable: un- → JJ
  - Suffixes: Importantly: -ly → RB
  - Capitalization: Meridian: CAP → NNP
  - Word shapes: 35-year: d-x → JJ
- Then build a maxent (or whatever) model to predict tag
  - Maxent P(t|w): 93.7% overall / 82.6% unknown

# Overview: POS Tagging Accuracies

- Rough accuracies:
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%
  - Maxent P(t|w): 93.7% / 82.6%
  - TnT (HMM++): 96.2% / 86.0%
  - MEMM tagger: 96.9% / 86.9%
  - Bidirectional dependencies: 97.2% / 90.0%
  - Upper bound: ~98% (human agreement)

# How to improve supervised results?

- Build better features!

RB

PRP VBD IN RB IN PRP VBD .
They left as soon as he arrived .

- We could fix this with a feature that looked at the next word

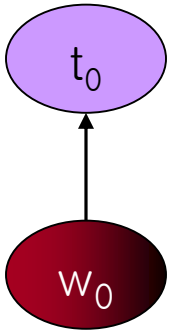
JJ

NNP NNS VBD VBN .
Intrinsic flaws remained undetected .

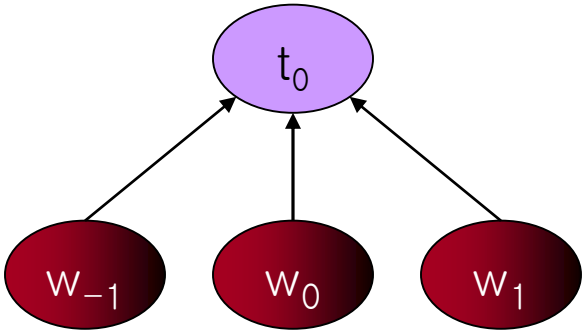
- We could fix this by linking capitalized words to their lowercase versions

# Tagging Without Sequence Information

Baseline




Three Words




| Model | Features | Token | Unknown | Sentence |
|---|---|---|---|---|
| Baseline | 56,805 | **93.69%** | 82.61% | 26.74% |
| 3Words | 239,767 | **96.57%** | 86.78% | 48.27% |

Using words only in a straight classifier works as well as a basic (HMM or discriminative) sequence model!!

# POS Tagging with NLTK

1. Tokenization
   - From nltk.tokenize import word_tokenize
   - text="Intelligence obtained in recent weeks found that an al Qaeda affiliate was perfecting techniques for hiding explosives in batteries and battery compartments of electronic devices, according to a US official"
   - tokens=nltk.word_tokenize(text)
2. POS Tagging
   - tagged=nltk.pos_tag(tokens)

# Exercise

- From the example sentence ‘text’, find all the nouns.
  - Use for-loop
  - Use POS_tagger