Lecture 12b: POS tagging

Motivating Example: Biocuration

Over 50,000 articles published per year relevant to cancer research.
No expert can read or remember that many

DARPA’s goal:
- Create an agent that reads every article
- Create an interface to let cancer boards access this information
- Implement well-informed, individualized cancer treatments

Pipeline of NLP IR Tools
- Scraping (not covered here)
- Sentence splitting
- Tokenization
  (Stemming / Lemmatization)
- Part-of-speech tagging
- Shallow parsing
- Named entity recognition
- Syntactic parsing
  (Semantic Role Labeling)

Part of Speech Tagging
- Parts of speech
  - What’s POS tagging good for anyhow?
- Tag sets
  - Rule-based tagging
  - Statistical tagging
    - Simple most-frequent-tag baseline
- Important ideas
  - Training sets and test sets
  - Unknown words
- HMM tagging

Parts of Speech
- 8 (ish) traditional parts of speech
  - Noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction, etc.
  - Called: parts-of-speech, lexical categories, word classes, morphological classes, lexical tags...
  - Lots of debate within linguistics about the number, nature, and universality of these
  - We’ll completely ignore this debate.

POS examples
- **N** noun
  - chair, bandwidth, pacing
- **V** verb
  - study, debate, munch
- **ADJ** adjective
  - purple, tall, ridiculous
- **ADV** adverb
  - unfortunately, slowly
- **P** preposition
  - of, by, to
- **PRO** pronoun
  - I, me, mine
- **DET** determiner
  - the, a, that, those
POS Tagging Definition

The process of assigning a part-of-speech or lexical class marker to each word in a collection.

Open and Closed Classes

Closed class: a small fixed membership
- Prepositions: of, in, by, ...
- Auxiliaries: may, can, will had, been, ...
- Pronouns: I, you, she, mine, his, them, ...
- Usually function words (short common words which play a role in grammar)

Open class: new ones can be created all the time
- English has 4: Nouns, Verbs, Adjectives, Adverbs
- Many languages have these 4, but not all!

Open Class Words

Nouns
- Proper nouns (Boulder, Eli Manning)
- English capitalizes these.
- Common nouns (the rest).
- Count nouns and mass nouns
  - Count: have plurals, get counted: goat/goats, one goat, two goats
  - Mass: don’t get counted (more: salt, communism) (*two snows)

Adverbs: tend to modify things
- Unfortunately, John walked home extremely slowly yesterday
- Directional/locative adverbs (here, home, downhill)
- Degree adverbs (extremely, very, somewhat)
- Manner adverbs (slowly, slinkily, delicately)

Verbs
- In English, have morphological affixes (eat/eats/eaten)

Closed Class Words

Examples:
- prepositions: on, under, over, ...
- particles: up, down, on, off, ...
- determiners: a, an, the, ...
- pronouns: she, who, I, ...
- conjunctions: and, but, or, ...
- auxiliary verbs: can, may should, ...
- numerals: one, two, three, third, ...

Prepositions from CELEX

| of | 540,085 | through | 14,964 | worth | 1,563 | pace | 12 |
| in | 331,235 | after | 13,670 | toward | 1,390 | high | 9 |
| for | 142,421 | between | 13,275 | plus | 750 | re | 4 |
| to | 125,691 | under | 9,525 | till | 686 | mid | 3 |
| with | 124,965 | per | 6,515 | amongst | 525 | o’er | 2 |
| on | 109,129 | among | 5,090 | via | 351 | but | 0 |
| at | 100,169 | within | 5,030 | amid | 222 | ere | 0 |
| by | 77,794 | towards | 4,700 | underneath | 164 | less | 0 |
| from | 74,843 | above | 3,056 | versus | 113 | might | 0 |
| about | 38,428 | near | 2,026 | amidst | 67 | o’ | 0 |
| than | 20,210 | off | 1,695 | sans | 20 | there | 0 |
| over | 18,071 | past | 1,575 | circa | 14 | vice | 0 |
English Particles

<table>
<thead>
<tr>
<th>Preposition</th>
<th>Part of Speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>about</td>
<td>Adverb</td>
</tr>
<tr>
<td>above</td>
<td>Adverb</td>
</tr>
<tr>
<td>across</td>
<td>Adjective</td>
</tr>
<tr>
<td>aboard</td>
<td>Adverb</td>
</tr>
<tr>
<td>about</td>
<td>Adverb</td>
</tr>
<tr>
<td>at</td>
<td>Adjective</td>
</tr>
<tr>
<td>among</td>
<td>Adjective</td>
</tr>
<tr>
<td>around</td>
<td>Adjective</td>
</tr>
<tr>
<td>at</td>
<td>Adjective</td>
</tr>
<tr>
<td>by</td>
<td>Adjective</td>
</tr>
<tr>
<td>behind</td>
<td>Adjective</td>
</tr>
<tr>
<td>beside</td>
<td>Adjective</td>
</tr>
<tr>
<td>between</td>
<td>Adjective</td>
</tr>
<tr>
<td>besides</td>
<td>Adjective</td>
</tr>
<tr>
<td>below</td>
<td>Adjective</td>
</tr>
<tr>
<td>beneath</td>
<td>Adjective</td>
</tr>
<tr>
<td>by</td>
<td>Adjective</td>
</tr>
<tr>
<td>close to</td>
<td>Adjective</td>
</tr>
<tr>
<td>except</td>
<td>Adjective</td>
</tr>
<tr>
<td>except for</td>
<td>Adjective</td>
</tr>
<tr>
<td>for</td>
<td>Adjective</td>
</tr>
</tbody>
</table>

POS Tagging: Choosing a Tagset

There are so many parts of speech, potential distinctions we can draw. To do POS tagging, we need to choose a standard set of tags to work with. Could pick very coarse tagsets:


More commonly used set is finer grained, the “Penn TreeBank tagset”, 45 tags:

- PRP, WRB, WPS, VBG

Even more fine-grained tagsets exist.

Using the Penn Tagset

The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./

Prepositions and subordinating conjunctions marked IN (“although/IN I/PRP..”)
Except the preposition/complementizer “to” is just marked “TO”.

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.
Three Methods for POS Tagging

1. Rule-based tagging
   - (ENGTWOL)

2. Stochastic
   1. Probabilistic sequence models
      - HMM (Hidden Markov Model) tagging
      - MEMMs (Maximum Entropy Markov Models)

3. Transformation Based tagging
   - Brill Tagger

Rule-Based Tagging

Start with a dictionary
Assign all possible tags to words from the dictionary
Write rules by hand to selectively remove tags
Leaving the correct tag for each word.

Start With a Dictionary
• she: PRP
• promised: VBN,VBD
• to: TO
• back: VB, JJ, RB, NN
• the: DT
• bill: NN, VB

Etc... for the ~100,000 words of English with more than 1 tag

Assign Every Possible Tag
NN RB
VBN JJ
VB PRP VBD TO VB DT NN
She promised to back the bill

Write Rules to Eliminate Tags

Eliminate VBN if VBD is an option when VBN|VBD follows "<start> PRP"

She promised to back the bill

Stage 1 of ENGTWOL Tagging

First Stage: Run words through FST morphological analyzer to get all parts of speech.

Example: Pavlov had shown that salivation ...

She promised to back the bill
Stage 2 of ENGTWOL Tagging

Second Stage: Apply NEGATIVE constraints.

Example: Adverbial “that” rule
- Eliminates all readings of “that” except the one in
  - “It isn’t that odd”

**Given input:** “that”
- (+1 A/ADV/QUANT) if next word is adj/adv/quantifier
- (+2 SENT-LIM) following which is E-O-S
- (NOT -1 SVOC/A) and the previous word is not a
  - verb like “consider” which
  - allows adjective complements
  - in “I consider that odd”

**Then** eliminate non-ADV tags
**Else** eliminate ADV

Conditional Probability and Tags

\[
P(\text{Verb}) \text{ is probability of randomly selected word being a verb.}
\]

\[
P(\text{Verb}|\text{race}) \text{ is “what’s the probability of a word being a verb given that it’s the word “race”?}
\]

- Race can be a noun or a verb.
- It’s more likely to be a noun.

\[
P(\text{Verb}|\text{race}) \text{ can be estimated by looking at some corpus and}
\]

\[
\text{saying “out of all the times we saw ‘race’, how many were verbs?}
\]

\[
P(\text{Verb}|\text{race}) = \frac{\text{Count(race is verb)}}{\text{Total Count(race)}}
\]

Most frequent tag

Some ambiguous words have a more frequent tag and a less frequent tag:
Consider the word “a” in these 2 sentences:

- would/MD prohibit/VB a/DT suit/NN for/IN refund/NN
- of/IN section/NN 381/CD ([/a/NN]/.)

Which do you think is more frequent?

Counting in a corpus

We could count in a corpus

A corpus: an on-line collection of text, often linguistically annotated
- The Brown Corpus: 1 million words from 1961 Part of speech tagged at U Penn
- I counted in this corpus
- The results for “a”:

<table>
<thead>
<tr>
<th>Word</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>DT</td>
</tr>
<tr>
<td>b</td>
<td>NN</td>
</tr>
<tr>
<td>2</td>
<td>FW</td>
</tr>
</tbody>
</table>

The Most Frequent Tag algorithm

For each word, we said:
- Create a dictionary with each possible tag for a word...

Where does the dictionary come from?
- One option is to use the same corpus that we use for computing the tags

The/Dt City/NNP Purchasing/NNP
Department/NNP ./, the/Dt jury/NN said/VBD ./
is/VBZ lacking/VBG in/IN experienced/VBN clerical/JJ personnel/NNS ...
Evaluating performance

How do we know how well a tagger does?
Say we had a test sentence, or a set of test sentences, that were already tagged by a human
- a "Gold Standard"

We could run a tagger on this set of test sentences
And see how many of the tags we got right.
- This is called "Tag accuracy" or "Tag percent correct"

Test set

We take a set of test sentences
- Hand-label them for part of speech
- The result is a "Gold Standard" test set

Who does this?
- Brown corpus: done by U Penn
- Grad students in linguistics

Don’t they disagree?
- Yes! But on about 97% of tags no disagreements
- And if you let the taggers discuss the remaining 3%, they often reach agreement

NOTE: we can’t train our frequencies on the test set sentences.

Computing % correct

Computing % correct
- Of all the words in the test set
- For what percent of them did the tag chosen by the tagger equal the human-selected tag.

Human tag set: ("Gold Standard" set)

\[ \text{\% correct} = \frac{\text{# of words tagged correctly in test set}}{\text{total \# of words in test set}} \]

Unknown Words

Most-frequent-tag approach has a problem!!
What about words that don’t appear in the training set?
For example, here are some words that occur in a small Brown Corpus test set but not the training set:
- Abernathy azalea alligator
- absolution baby-sitter asparagus
- Adrien bantered boxcar
- ajar bare-armed boxcars
- Alicia big-boned bumped
- all-american boy boathouses

Unknown words

New words added to (newspaper) language 20+ per month
Plus many proper names ...

Increases error rates by 1-2%
- Method 1: assume they are nouns
- Method 2: assume the unknown words have a probability distribution similar to words only occurring once in the training set.
- Method 3: Use morphological information, e.g., words ending with -ed tend to be tagged VBN.

Rule-Based Tagger

The Linguistic Complaint
- Where is the linguistic knowledge of a tagger?
- Just a massive table of numbers
- Aren't there any linguistic insights that could emerge from the data?
- Could thus use handcrafted sets of rules to tag input sentences, for example, if input follows a determiner tag it as a noun.
The Brill tagger

An example of TRANSFORMATION-BASED LEARNING

Very popular (freely available, works fairly well)

A SUPERVISED method: requires a tagged corpus

Basic idea: do a quick job first (using frequency), then revise it using contextual rules

An example

Examples:

- They are expected to race tomorrow.
- The race for outer space.

Tagging algorithm:

- Tag all uses of "race" as NN (most likely tag in the Brown corpus)
  - They are expected to race/NN tomorrow
  - the race/NN for outer space
- Use a transformation rule to replace the tag NN with VB for all uses of "race" preceded by the tag TO:
  - They are expected to race/VB tomorrow
  - the race/NN for outer space

Brill Tagging: In more detail

Start with simple (less accurate) rules...learn better ones from tagged corpus

- Tag each word initially with most likely POS
- Examine set of transformations to see which improves tagging decisions compared to tagged corpus
- Re-tag corpus using best transformation
- Repeat until, e.g., performance doesn't improve
- Result: tagging procedure (ordered list of transformations) which can be applied to new, untagged text

Transformation Rules for Tagging Unknown Words

Hidden Markov Model Tagging

Using an HMM to do POS tagging is a special case of Bayesian inference

- Foundational work in computational linguistics
- Bledsoe 1959: OCR
- Mosteller and Wallace 1964: authorship identification

It is also related to the "noisy channel" model that's the basis for ASR, OCR and MT

First 20 Transformation Rules

<table>
<thead>
<tr>
<th>Transformation Rule</th>
<th>Condition</th>
<th>New Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[4] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[5] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[6] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[7] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[8] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[9] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[10] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[11] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[12] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[13] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[14] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[15] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[16] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[17] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[18] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[19] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
<tr>
<td>[20] VB to VB</td>
<td>NY to VB</td>
<td>VB</td>
</tr>
</tbody>
</table>
**POS Tagging as Sequence Classification**

We are given a sentence (an "observation" or "sequence of observations")

- Secretariat is expected to race tomorrow

What is the best sequence of tags that corresponds to this sequence of observations?

Probabilistic view:
- Consider all possible sequences of tags
- Out of this universe of sequences, choose the tag sequence which is most probable given the observation sequence of n words $w_1 \ldots w_n$.

**Sequence Labeling as Classification**

Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

Classifier

- NNP
- VBD
- DT
- NN
- CC
Sequence Labeling as Classification
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.
Sequence Labeling as Classification
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

Using Outputs as Inputs
Better input features are usually the categories of the surrounding tokens, but these are not available yet.
Can use category of either the preceding or succeeding tokens by going forward or back and using previous output.

Forward Classification

John saw the saw and decided to take it to the table.

John saw the saw and decided to take it to the table.

John saw the saw and decided to take it to the table.
John saw the saw and decided to take it to the table.
John saw the saw and decided to take it to the table.

Disambiguating “to” in this case would be even easier backward.
Backward Classification
Disambiguating “to” in this case would be even easier backward.

John saw the saw and decided to take it to the table.

Classifier
VB

Classifier
TO
Backward Classification

Disambiguating "to" in this case would be even easier backward.

John saw the saw and decided to take it to the table.

**HMMs: A Probabilistic Approach**

What you want to do is find the "best sequence" of POS tags T=T1...Tn for a sentence W=W1...Wn.
- (Here T is pos_tag(Wi)).
- Find a sequence of POS tags T that maximizes P(T|W).

Using Bayes' Rule, we can say

\[ P(T|W) = \frac{P(W|T) * P(T)}{P(W)} \]

We want to find the value of T which maximizes the RHS. The denominator can be discarded (same for every T).

Find T which maximizes \( P(W|T) * P(T) \)

Example: He will race

Possible sequences:
- He/PRP will/MOD race/NN
- He/PRP will/NN race/NN
- He/PRP will/MOD race/VB
- He/PRP will/NN race/VB

W = W1 W2 W3
- He will race

\( T = T1 T2 T3 \)
- Choices:
  - T1 = PRP MOD NN
  - T1 = PRP NN NN
  - T1 = PRP MOD VB
  - T1 = PRP NN VB

**Independence Assumptions**

Assume that current event is based only on previous n-1 events for a bigram model, it's based only on previous 1 event)

\[ P(T1...Tn) \geq \Pi_{i=1}^{n} P(Ti|Ti-1) \]

- Assumes that the event of a POS tag occurring is independent of the event of any other POS tag occurring, except for the immediately previous POS tag.
- From a linguistic standpoint, this seems an unreasonable assumption, due to long-distance dependencies.

\[ P(W1...Wn | T1...Tn) \geq \Pi_{i=1}^{n} P(Wi | Ti) \]

- Assumes that the event of a word appearing in a category is independent of the event of any surrounding word or tag, except for the tag at this position.

**Ngram Models**

POS problem formulation
- Given a sequence of words, find a sequence of categories that maximizes \( P(T1...Tn | W1...Wn) \)
- i.e., that maximizes \( P(W1...Wn | T1...Tn) * P(T1...Tn) \) (by Bayes’ Rule)

Chain Rule of probability:

\[ P(W|T) = \Pi_{i=1}^{n} P(Wi | Ti) \]

- prob. of this word based on previous words & tags

\[ P(T) = \Pi_{i=1}^{n} P(Ti | W1...Wn..Ti-1) \]

- prob. of this tag based on previous words & tags

But we don’t have sufficient data for this, and we would likely overfit the data, so we would make some assumptions to simplify the problem.

**Hidden Markov Models**

Linguists know both these assumptions are incorrect!
- But, nevertheless, statistical approaches based on these assumptions work pretty well for part-of-speech tagging

In particular, with Hidden Markov Models (HMMs)
- Very widely used in both POS-tagging and speech recognition, among other problems
- A Markov model, or Markov chain, is just a weighted Finite State Automation
POS Tagging Based on Bigrams

Problem: Find T which maximizes P(W | T) * P(T)
  - Here W=W1...Wn and T=T1...Tn

Using the bigram model, we get:
  - Transition probabilities (prob. of transitioning from one state/tag to another):
    - P(T1...Tn) = Π P(Ti | Ti-1)
  - Emission probabilities (prob. of emitting a word at a given state):
    - P(W1...Wn | T1...Tn) = Π P(Wi | Ti)

So, we want to find the value of T1...Tn which maximizes:
  Π P(Wi | Ti) * P(Ti | Ti-1)

Two Kinds of Probabilities

Tag transition probabilities p(ti | ti-1)
  - Determiners likely to precede adjs and nouns
  - That/DT flight/NN
  - The/DT yellow/JJ hat/NN
  - So we expect P(NN|DT) and P(JJ|DT) to be high
  - But P(DT|JJ) to be:
    - Compute P(NN|DT) by counting in a labeled corpus:
      \[ P(t_i | t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \]
      \[ P(\text{NN}|\text{DT}) = \frac{56,509}{116,454} = .49 \]

Two Kinds of Probabilities

Word likelihood probabilities p(wi | ti)
  - VBZ (3sg Pres verb) likely to be “is”
  - Compute P(is|VBZ) by counting in a labeled corpus:
    \[ P(w_i | t_i) = \frac{C(t_i, w_i)}{C(t_i)} \]
    \[ P(\text{is}|\text{VBZ}) = \frac{10,073}{21,627} = .47 \]

Example: The Verb “race”

Secretariat/NNP is/VBZ expected/VBN to/TO race/VB tomorrow/NR
People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN

How do we pick the right tag?

Disambiguating “race”

Example

P(NN|TO) = .00047
P(VB | TO) = .83
P(race | NN) = .00057
P(race | VB) = .00012
P(NR | VB) = .0027
P(NR | NN) = .0012
P(VB | TO)P(NR | VB)P(race | VB) = .00000027
P(NN | TO)P(NR | NN)P(race | NN) = .0000000032

So we (correctly) choose the verb reading,
Hidden Markov Model for POS

States \( Q = q_1, q_2, \ldots, q_N \) are POS tags

Observations \( O = o_1, o_2, \ldots, o_N \):
- Each observation is a symbol (usually word) from the vocabulary \( V = \{v_1, v_2, \ldots, v_V\} \)

Transition probabilities
- Transition probability matrix \( A = \{a_{ij}\} \)
  \[ a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \leq i, j \leq N \]

Observation likelihoods
- Output probability matrix \( B = \{b_i(k)\} \)
  \[ b_i(k) = P(X_t = o_k \mid q_t = i) \]

Special initial probability vector \( \pi \)
- \[ \pi_i = P(q_1 = i) \quad 1 \leq i \leq N \]