CS4705

Part of Speech tagging

Some slides adapted from: Dan Jurafsky, Julia Hirschberg, Jim Martin
HW questions?

- Training files, question samples
  - /home/cs4705/corpora/wsj
  - /home/cs4705/corpora/wsj/wsj_2300questions.txt
  - CVN: will post on the CVN web site this afternoon

- Question and answer templates

- Not expected to use tools that we haven’t gone over (e.g., named entity recognition)
  - Must allow paraphrases for indices
  - But company names will be provided in question exactly as they appear in article

- Any other questions?
Smoothing

- Words follow a Zipfian distribution
  - Small number of words occur very frequently
  - A large number are seen only once
  - Zipf’s law: a word’s frequency is approximately inversely proportional to its rank in the word distribution list

- Zero probabilities on one bigram cause a zero probability on the entire sentence
Smoothing is like Robin Hood: Steal from the rich and give to the poor (in probability mass)

- We often want to make predictions from sparse statistics:

  \[ P(w \mid \text{denied the}) \]
  3 allegations
  2 reports
  1 claims
  1 request
  7 total

- Smoothing flattens spiky distributions so they generalize better

  \[ P(w \mid \text{denied the}) \]
  2.5 allegations
  1.5 reports
  0.5 claims
  0.5 request
  2 other
  7 total

- Very important all over NLP, but easy to do badly!
Smoothing Methods

- **Add-one smoothing (easy, but inaccurate)**
  - Add 1 to every word count (Note: this is type)
  - Increment normalization factor by Vocabulary size: N (tokens) + $V\text{(types)}$:
    \[
    p_i^* = \frac{c_i + 1}{N + V}
    \]

- **Backoff models**
  - When a count for an n-gram is 0, back off to the count for the (n-1)-gram
  - These can be weighted

- **Class-based smoothing**
  - For certain types of n-grams, back off to the count of its syntactic class
  - E.g., Count ProperNouns in place of names (e.g., Obama)

- **Good-Turing**
  - Re-estimate amount of probability mass for zero (or low count) n-grams by looking at ngrams with higher counts
  - Estimate
    \[
    c^* = (c + 1)\frac{N_{c+1}}{N_c}
    \]
Garden path sentences

- The old dog the footsteps of the young.
- The cotton clothing is made of grows in Mississippi.
- The horse raced past the barn fell.
Words that somehow ‘behave’ alike:
- Appear in similar contexts
- Perform similar functions in sentences
- Undergo similar transformations

9 (or so) traditional parts of speech
- Noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction,
## POS examples

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>noun</th>
<th>chair, bandwidth, pacing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V</td>
<td>verb</td>
<td>study, debate, munch</td>
</tr>
<tr>
<td></td>
<td>ADJ</td>
<td>adjective</td>
<td>purple, tall, ridiculous</td>
</tr>
<tr>
<td></td>
<td>ADV</td>
<td>adverb</td>
<td>unfortunately, slowly,</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>preposition</td>
<td>of, by, to</td>
</tr>
<tr>
<td></td>
<td>PRO</td>
<td>pronoun</td>
<td>I, me, mine</td>
</tr>
<tr>
<td></td>
<td>DET</td>
<td>determiner</td>
<td>the, a, that, those</td>
</tr>
</tbody>
</table>
The process of assigning a part-of-speech or lexical class marker to each word in a corpus:
Is the first step of a vast number of Comp Ling tasks

Speech synthesis:
- How to pronounce “lead”?
- INsult → inSULT
- OBject → obJECT
- OVERflow → overFLOW
- DIScount → disCOUNT
- CONtent → conTENT

Parsing
- Need to know if a word is an N or V before you can parse

Word prediction in speech recognition
- Possessive pronouns (my, your, her) followed by nouns
- Personal pronouns (I, you, he) likely to be followed by verbs

Machine Translation
Open and closed class words

- **Closed class**: a relatively 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 all 4, but not all!
  - In Lakhota and possibly Chinese, what English treats as adjectives act more like verbs.
Open class words

- **Nouns**
  - Proper nouns (**Columbia University**, **New York City**, **Arthi Ramachandran**, **Metropolitan Transit Center**). English capitalizes these.
  - Common nouns (**the rest**). German capitalizes these.
  - Count nouns and mass nouns
    - Count: have plurals, get counted: **goat/goats**, **one goat**, **two goats**
    - Mass: don’t get counted (**fish**, **salt**, **communism**) (*two fishes*)

- **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**)  
  - Actions (**walk**, **ate**) and states (**be**, **exude**)
Many subclasses, e.g.
- eats/V ⇒ eat/VB, eat/VBP, eats/VBZ, ate/VBD, eaten/VBN, eating/VBG, ...
- Reflect morphological form & syntactic function
How do we decide which words go in which classes?

- **Nouns** denote people, places and things and can be preceded by articles? But...
  
  My typing is very bad.
  *The Mary loves John.

- **Verbs** are used to refer to actions, processes, states
  
  But some are **closed class** and some are **open**

  I will have **emailed** everyone by noon.

- **Adverbs** modify actions

  - Is **Monday** a temporal adverb or a noun? Some others?
Closed Class Words

- Idiosyncratic
- Closed class words (Prep, Det, Pron, Conj, Aux, Part, Num) are easier, since we can enumerate them....but
  - Part vs. Prep
    - George eats up his dinner/George eats his dinner up.
    - George eats up the street/*George eats the street up.
  - Articles come in 2 flavors: **definite (the) and indefinite (a, an)**
To do POS tagging, need to choose a standard set of tags to work with.
Could pick very coarse tagsets:
Brown Corpus (Francis & Kucera ‘82), 1M words, 87 tags
Penn Treebank: hand-annotated corpus of *Wall Street Journal*, 1M words, 45–46 tags
- Commonly used
- Set is finer grained,
Even more fine-grained tagsets exist.
# Penn TreeBank POS Tag set

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Coordin. Conjunction</td>
<td><em>and, but, or</em></td>
<td>SYM</td>
<td>Symbol</td>
<td><em>+, %, &amp;</em></td>
</tr>
<tr>
<td>CD</td>
<td>Cardinal number</td>
<td><em>one, two, three</em></td>
<td>TO</td>
<td>“to”</td>
<td><em>to</em></td>
</tr>
<tr>
<td>DT</td>
<td>Determiner</td>
<td><em>a, the</em></td>
<td>UH</td>
<td>Interjection</td>
<td><em>ah, oops</em></td>
</tr>
<tr>
<td>EX</td>
<td>Existential ‘there’</td>
<td><em>there</em></td>
<td>VB</td>
<td>Verb, base form</td>
<td><em>eat</em></td>
</tr>
<tr>
<td>FW</td>
<td>Foreign word</td>
<td><em>mea culpa</em></td>
<td>VBD</td>
<td>Verb, past tense</td>
<td><em>ate</em></td>
</tr>
<tr>
<td>IN</td>
<td>Preposition/sub-conj</td>
<td><em>of, in, by</em></td>
<td>VBG</td>
<td>Verb, gerund</td>
<td><em>eating</em></td>
</tr>
<tr>
<td>JJ</td>
<td>Adjective</td>
<td><em>yellow</em></td>
<td>VBN</td>
<td>Verb, past participle</td>
<td><em>eaten</em></td>
</tr>
<tr>
<td>JJR</td>
<td>Adj., comparative</td>
<td><em>bigger</em></td>
<td>VBP</td>
<td>Verb, non-3sg pres</td>
<td><em>eat</em></td>
</tr>
<tr>
<td>JJS</td>
<td>Adj., superlative</td>
<td><em>wildest</em></td>
<td>VBZ</td>
<td>Verb, 3sg pres</td>
<td><em>eats</em></td>
</tr>
<tr>
<td>LS</td>
<td>List item marker</td>
<td><em>1, 2, One</em></td>
<td>WDT</td>
<td>Wh-determiner</td>
<td><em>which, that</em></td>
</tr>
<tr>
<td>MD</td>
<td>Modal</td>
<td><em>can, should</em></td>
<td>WP</td>
<td>Wh-pronoun</td>
<td><em>what, who</em></td>
</tr>
<tr>
<td>NN</td>
<td>Noun, sing. or mass</td>
<td><em>llama</em></td>
<td>WPS</td>
<td>Possessive wh-</td>
<td><em>whose</em></td>
</tr>
<tr>
<td>NNS</td>
<td>Noun, plural</td>
<td><em>llamas</em></td>
<td>WRB</td>
<td>Wh-adverb</td>
<td><em>how, where</em></td>
</tr>
<tr>
<td>NNP</td>
<td>Proper noun, singular</td>
<td><em>IBM</em></td>
<td>$</td>
<td>Dollar sign</td>
<td>$</td>
</tr>
<tr>
<td>NNPS</td>
<td>Proper noun, plural</td>
<td><em>Carolinanas</em></td>
<td>#</td>
<td>Pound sign</td>
<td>#</td>
</tr>
<tr>
<td>PDT</td>
<td>Predeterminer</td>
<td><em>all, both</em></td>
<td>“</td>
<td>Left quote</td>
<td>(‘ or “)</td>
</tr>
<tr>
<td>POS</td>
<td>Possessive ending</td>
<td><em>’s</em></td>
<td>”</td>
<td>Right quote</td>
<td>(’ or ”)</td>
</tr>
<tr>
<td>PRP</td>
<td>Personal pronoun</td>
<td><em>I, you, he</em></td>
<td>(</td>
<td>Left parenthesis</td>
<td>([, (, {, &lt;)</td>
</tr>
<tr>
<td>PRP$</td>
<td>Possessive pronoun</td>
<td><em>your, one’s</em></td>
<td>)</td>
<td>Right parenthesis</td>
<td>([, ), }, &gt;)</td>
</tr>
<tr>
<td>RB</td>
<td>Adverb</td>
<td><em>quickly, never</em></td>
<td>,</td>
<td>Comma</td>
<td>,</td>
</tr>
<tr>
<td>RBR</td>
<td>Adverb, comparative</td>
<td><em>faster</em></td>
<td>;</td>
<td>Sentence-final punc</td>
<td>(‘ ! ?)</td>
</tr>
<tr>
<td>RBS</td>
<td>Adverb, superlative</td>
<td><em>fastest</em></td>
<td>:</td>
<td>Mid-sentence punc</td>
<td>(: ; ... - -)</td>
</tr>
<tr>
<td>RP</td>
<td>Particle</td>
<td><em>up, off</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Using the UPenn tagset

- The grand jury commented on a number of other topics.

- Prepositions and subordinating conjunctions marked IN ("although I...")

- Except the preposition/complementizer "to" is just marked "to".
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.
How do we assign POS tags to words in a sentence?

- Time flies like an arrow.
- Time/[V,N] flies/[V,N] like/[V,Prep] an/Det arrow/N
- Time/N flies/V like/Prep an/Det arrow/N
- Fruit/N flies/N like/V a/DET banana/N
- Fruit/N flies/V like/Prep a/DET banana/N
- The/Det flies/N like/V a/DET banana/N
How hard is POS tagging?  
Measuring ambiguity

<table>
<thead>
<tr>
<th></th>
<th>Original 87-tag corpus</th>
<th>Treebank 45-tag corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unambiguous (1 tag)</strong></td>
<td><strong>44,019</strong></td>
<td><strong>38,857</strong></td>
</tr>
<tr>
<td><strong>Ambiguous (2–7 tags)</strong></td>
<td><strong>5,490</strong></td>
<td><strong>8844</strong></td>
</tr>
<tr>
<td>Details:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 tags</td>
<td>4,967</td>
<td>6,731</td>
</tr>
<tr>
<td>3 tags</td>
<td>411</td>
<td>1,621</td>
</tr>
<tr>
<td>4 tags</td>
<td>91</td>
<td>357</td>
</tr>
<tr>
<td>5 tags</td>
<td>17</td>
<td>90</td>
</tr>
<tr>
<td>6 tags</td>
<td>2 (well, beat)</td>
<td>32</td>
</tr>
<tr>
<td>7 tags</td>
<td>2 (still, down)</td>
<td>6 (well, set, round, open, fit, down)</td>
</tr>
<tr>
<td>8 tags</td>
<td></td>
<td>4 (’s, half, back, a)</td>
</tr>
<tr>
<td>9 tags</td>
<td></td>
<td>3 (that, more, in)</td>
</tr>
</tbody>
</table>
Many words have only one POS tag (e.g. is, Mary, very, smallest)

Others have a single most likely tag (e.g. a, dog)

But tags also tend to co-occur regularly with other tags (e.g. Det, N)

In addition to conditional probabilities of words $P(w_1|w_{n-1})$, we can look at POS likelihoods $P(t_1|t_{n-1})$ to disambiguate sentences and to assess sentence likelihoods.
3 methods for POS tagging

1. Rule-based tagging
   ◦ (ENGTWOL)

2. Transformation-based tagging
   1. Learned rules (statistic and linguistic)
      ◦ Brill tagger

3. Stochastic (=Probabilistic) tagging
   ◦ HMM (Hidden Markov Model) tagging
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
Use the dictionary to assign every possible tag

She promised to back the bill
Write rules to eliminate tags

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

```
NN
RB
VBN
JJ
VB
PRP VBD TO VB DT NN
She promised to back the bill
```
Sample ENGTWOL Lexicon

<table>
<thead>
<tr>
<th>Word</th>
<th>POS</th>
<th>Additional POS features</th>
</tr>
</thead>
<tbody>
<tr>
<td>smaller</td>
<td>ADJ</td>
<td>COMPARATIVE</td>
</tr>
<tr>
<td>entire</td>
<td>ADJ</td>
<td>ABSOLUTE ATTRIBUTIVE</td>
</tr>
<tr>
<td>fast</td>
<td>ADV</td>
<td>SUPERLATIVE</td>
</tr>
<tr>
<td>that</td>
<td>DET</td>
<td>CENTRAL DEMONSTRATIVE SG</td>
</tr>
<tr>
<td>all</td>
<td>DET</td>
<td>PREDETERMINER SG/PL QUANTIFIER</td>
</tr>
<tr>
<td>dog’s</td>
<td>N</td>
<td>GENITIVE SG</td>
</tr>
<tr>
<td>furniture</td>
<td>N</td>
<td>NOMINATIVE SG NOINDEFDETERMINER</td>
</tr>
<tr>
<td>one-third</td>
<td>NUM</td>
<td>SG</td>
</tr>
<tr>
<td>she</td>
<td>PRON</td>
<td>PERSONAL FEMININE NOMINATIVE SG3</td>
</tr>
<tr>
<td>show</td>
<td>V</td>
<td>IMPERATIVE VFIN</td>
</tr>
<tr>
<td>show</td>
<td>V</td>
<td>PRESENT -SG3 VFIN</td>
</tr>
<tr>
<td>show</td>
<td>N</td>
<td>NOMINATIVE SG</td>
</tr>
<tr>
<td>shown</td>
<td>PCP2</td>
<td>SVOO SVO SV</td>
</tr>
<tr>
<td>occurred</td>
<td>PCP2</td>
<td>SV</td>
</tr>
<tr>
<td>occurred</td>
<td>V</td>
<td>PAST VFIN SV</td>
</tr>
</tbody>
</table>
### 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 ...*

<table>
<thead>
<tr>
<th>Word</th>
<th>Tagging</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pavlov</td>
<td>PAVLOV N NOM SG PROPER</td>
</tr>
<tr>
<td>had</td>
<td>HAVE V PAST VFIN SVO</td>
</tr>
<tr>
<td>shown</td>
<td>SHOW PCP2 SVOO SVO SV</td>
</tr>
<tr>
<td>that</td>
<td>ADV PRON DEM SG DET CENTRAL DEM SG</td>
</tr>
<tr>
<td>salivation</td>
<td>N NOM SG</td>
</tr>
</tbody>
</table>
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”

**If**
- (+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
Transformation–Based Tagging (Brill Tagging)

- Combination of Rule–based and stochastic tagging methodologies
  - Like rule–based because rules are used to specify tags in a certain environment
  - Like stochastic approach because machine learning is used—with tagged corpus as input
    - Rules are learned

- Input:
  - tagged corpus
  - dictionary (with most frequent tags)
Transformation-Based Tagging

- **Basic Idea:**
  - Set the most probable tag for each word as a start value
  - Change tags according to rules of type “if word-1 is a determiner and word is a verb then change the tag to noun” in a specific order

- **Training is done on tagged corpus:**
  - Use a set of rule templates
  - Among the set of rules, find one with highest score
  - Continue finding rules until lowest score threshold is passed
  - Keep the ordered set of rules

- **Rules make errors that are corrected by later rules**
Tagger labels every word with its most-likely tag
  ◦ For example: race has the following probabilities in the Brown corpus:
    • \( P(\text{NN}/\text{race}) = .98 \)
    • \( P(\text{VB}/\text{race}) = .02 \)

Transformation rules make changes to tags
  ◦ “Change NN to VB when previous tag is TO”
    ... is/VBZ expected/VBN to/TO race/NN tomorrow/NN becomes
    ... is/VBZ expected/VBN to/TO race/VB tomorrow/NN
### TBL: Rule Learning

- **2 parts to a rule**
  - Triggering environment
  - Rewrite rule

- The range of triggering environments of templates *(from Manning & Schutze 1999:363)*

<table>
<thead>
<tr>
<th>Schema</th>
<th>$t_i-3$</th>
<th>$t_i-2$</th>
<th>$t_i-1$</th>
<th>$t_i$</th>
<th>$t_{i+1}$</th>
<th>$t_{i+2}$</th>
<th>$t_{i+3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TBL: The Tagging Algorithm

- Step 1: Label every word with most likely tag (from dictionary)
- Step 2: Check every possible transformation & select one which most improves tagging
- Step 3: Re-tag corpus applying the rules
- Repeat 2–3 until some criterion is reached, e.g., X% correct with respect to training corpus
- RESULT: Sequence of transformation rules
Problem: Could apply transformations ad infinitum!

Constrain the set of transformations with “templates”:
- Replace tag X with tag Y, provided tag Z or word Z’ appears in some position

Rules are learned in ordered sequence

Rules may interact.

Rules are compact and can be inspected by humans
Templates for TBL

The preceding (following) word is tagged z.
The word two before (after) is tagged z.
One of the two preceding (following) words is tagged z.
One of the three preceding (following) words is tagged z.
The preceding word is tagged z and the following word is tagged w.
The preceding (following) word is tagged z and the word
two before (after) is tagged w.

<table>
<thead>
<tr>
<th>#</th>
<th>From</th>
<th>To</th>
<th>Condition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NN</td>
<td>VB</td>
<td>Previous tag is TO</td>
<td>to/TO race/NN → VB</td>
</tr>
<tr>
<td>2</td>
<td>VBP</td>
<td>VB</td>
<td>One of the previous 3 tags is MD</td>
<td>might/MD vanish/VBP → VB</td>
</tr>
<tr>
<td>3</td>
<td>NN</td>
<td>VB</td>
<td>One of the previous 2 tags is MD</td>
<td>might/MD not reply/NN → VB</td>
</tr>
<tr>
<td>4</td>
<td>VB</td>
<td>NN</td>
<td>One of the previous 2 tags is DT</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>VBD</td>
<td>VBN</td>
<td>One of the previous 3 tags is VBZ</td>
<td></td>
</tr>
</tbody>
</table>
Comparison of two approaches

- Accuracy
- Coverage
- Ease of building such a system
  - What is needed?
- Ease in porting to a new genre/new domain
- Baseline?
Summary

Parts of speech
- What’s POS tagging good for anyhow?
- Tag sets
- Rule–based tagging
- Learning rules: statistical and linguistic
- Next time:
  - HMM Tagging