Part-of-Speech Tagging

Chapter 8
(8.1-8.4.6)

Outline

- Parts of speech (POS)
- Tagsets
- POS Tagging
  - Rule-based tagging
  - Probabilistic (HMM) tagging
Garden Path Sentences

- The old dog the footsteps of the young

Parts of Speech

- 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.
Parts of Speech

- Traditional parts of speech
  - ~ 8 of them

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

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

```
the         tag
koala       DET
put
the
keys
on
the
table
```

```
WORD
tag
```

```
the         tag
KOALA       DET
put
the
keys
on
the
table
```

```
WORD
tag
```

```
the         tag
KOALA       DET
put
the
keys
on
the
table
```
POS Tagging

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

<table>
<thead>
<tr>
<th>WORD</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>koala</td>
<td>N</td>
</tr>
<tr>
<td>put</td>
<td>V</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>keys</td>
<td>N</td>
</tr>
<tr>
<td>on</td>
<td>V</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>table</td>
<td>N</td>
</tr>
</tbody>
</table>
**POS Tagging**

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

<table>
<thead>
<tr>
<th>WORD</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>koala</td>
<td>N</td>
</tr>
<tr>
<td>put</td>
<td>V</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>keys</td>
<td>N</td>
</tr>
<tr>
<td>on</td>
<td>V</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>table</td>
<td>N</td>
</tr>
</tbody>
</table>

1/23/2020 Speech and Language Processing - Jurafsky and Martin
POS Tagging

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

<table>
<thead>
<tr>
<th>WORD</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>koala</td>
<td>N</td>
</tr>
<tr>
<td>put</td>
<td>V</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>keys</td>
<td>N</td>
</tr>
<tr>
<td>on</td>
<td>P</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>table</td>
<td></td>
</tr>
</tbody>
</table>
POS Tagging

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

<table>
<thead>
<tr>
<th>WORD</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>koala</td>
<td>N</td>
</tr>
<tr>
<td>put</td>
<td>V</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>keys</td>
<td>N</td>
</tr>
<tr>
<td>on</td>
<td>P</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>table</td>
<td>N</td>
</tr>
</tbody>
</table>

Why is POS Tagging Useful?

- First step of many practical tasks, e.g.
  - Speech synthesis (aka text to speech)
    - How to pronounce "lead"?
    - OBJ ect obJ ect
    - OB ject obJ ect
  - Parsing
    - Need to know if a word is an N or V before you can parse
  - Information extraction
    - Finding names, relations, etc.
  - Language modeling
    - Backoff
Why is POS Tagging Difficult?

- Words often have more than one POS:
  - back
    - The back door = adjective
    - On my back =
    - Win the voters back =
    - Promised to back the bill =
Why is POS Tagging Difficult?

- Words often have more than one POS:
  - *back*
    - The *back* door = adjective
    - On my *back* = noun
    - Win the voters *back* = adverb
    - Promised to *back* the bill =

- 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

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
Open vs. Closed Classes

- Closed class: *why?*
  - Determiners: a, an, the
  - 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: *why?*
  - English has 4: Nouns, Verbs, Adjectives, Adverbs
  - Many languages have these 4, but not all!

Open vs. Closed Classes

- Closed class: a small fixed membership
  - Determiners: a, an, the
  - 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 (Pittsburgh, Pat Gallagher)
    - 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 (snow, 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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>of</strong></td>
<td>540,085</td>
</tr>
<tr>
<td><strong>in</strong></td>
<td>331,235</td>
</tr>
<tr>
<td><strong>for</strong></td>
<td>142,421</td>
</tr>
<tr>
<td><strong>to</strong></td>
<td>125,691</td>
</tr>
<tr>
<td><strong>with</strong></td>
<td>124,965</td>
</tr>
<tr>
<td><strong>on</strong></td>
<td>109,129</td>
</tr>
<tr>
<td><strong>at</strong></td>
<td>100,169</td>
</tr>
<tr>
<td><strong>by</strong></td>
<td>77,794</td>
</tr>
<tr>
<td><strong>from</strong></td>
<td>74,843</td>
</tr>
<tr>
<td><strong>about</strong></td>
<td>38,428</td>
</tr>
<tr>
<td><strong>than</strong></td>
<td>20,210</td>
</tr>
<tr>
<td><strong>over</strong></td>
<td>18,071</td>
</tr>
<tr>
<td><strong>through</strong></td>
<td>14,964</td>
</tr>
<tr>
<td><strong>after</strong></td>
<td>13,670</td>
</tr>
<tr>
<td><strong>between</strong></td>
<td>13,275</td>
</tr>
<tr>
<td><strong>under</strong></td>
<td>9,525</td>
</tr>
<tr>
<td><strong>per</strong></td>
<td>6,515</td>
</tr>
<tr>
<td><strong>among</strong></td>
<td>5,090</td>
</tr>
<tr>
<td><strong>within</strong></td>
<td>5,030</td>
</tr>
<tr>
<td><strong>towards</strong></td>
<td>4,700</td>
</tr>
<tr>
<td><strong>above</strong></td>
<td>3,056</td>
</tr>
<tr>
<td><strong>near</strong></td>
<td>2,026</td>
</tr>
<tr>
<td><strong>off</strong></td>
<td>1,695</td>
</tr>
<tr>
<td><strong>past</strong></td>
<td>1,575</td>
</tr>
<tr>
<td><strong>worth</strong></td>
<td>1,563</td>
</tr>
<tr>
<td><strong>toward</strong></td>
<td>1,390</td>
</tr>
<tr>
<td><strong>plus</strong></td>
<td>750</td>
</tr>
<tr>
<td><strong>till</strong></td>
<td>686</td>
</tr>
<tr>
<td><strong>amongst</strong></td>
<td>525</td>
</tr>
<tr>
<td><strong>via</strong></td>
<td>351</td>
</tr>
<tr>
<td><strong>amid</strong></td>
<td>222</td>
</tr>
<tr>
<td><strong>underneath</strong></td>
<td>164</td>
</tr>
<tr>
<td><strong>versus</strong></td>
<td>113</td>
</tr>
<tr>
<td><strong>amidst</strong></td>
<td>67</td>
</tr>
<tr>
<td><strong>sans</strong></td>
<td>20</td>
</tr>
<tr>
<td><strong>circum</strong></td>
<td>14</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
- Even more fine-grained tagsets exist
Penn TreeBank POS Tagset

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Example 1</th>
<th>Tag</th>
<th>Description</th>
<th>Example 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>coordinating conjunction</td>
<td>and, but, or</td>
<td>SYM</td>
<td>symbol</td>
<td>+, %, &amp;</td>
</tr>
<tr>
<td>CD</td>
<td>cardinal number</td>
<td>one, two, three</td>
<td>TO</td>
<td>“to”</td>
<td>to</td>
</tr>
<tr>
<td>DT</td>
<td>determiner</td>
<td>a, the</td>
<td>UH</td>
<td>interjection</td>
<td>ah, oops</td>
</tr>
<tr>
<td>EX</td>
<td>existential ‘there’</td>
<td>there</td>
<td>VB</td>
<td>verb, base form</td>
<td>eat</td>
</tr>
<tr>
<td>FW</td>
<td>foreign word</td>
<td>mea culpa</td>
<td>VBD</td>
<td>verb, past tense</td>
<td>ate</td>
</tr>
<tr>
<td>IN</td>
<td>preposition/sub-conj</td>
<td>of, in, by</td>
<td>VBG</td>
<td>verb, gerund</td>
<td>eating</td>
</tr>
<tr>
<td>JJ</td>
<td>adjective</td>
<td>yellow</td>
<td>VBN</td>
<td>verb, past participle</td>
<td>eaten</td>
</tr>
<tr>
<td>JJR</td>
<td>adj., comparative</td>
<td>bigger</td>
<td>VBP</td>
<td>verb, non-3rd pres</td>
<td>eat</td>
</tr>
<tr>
<td>JJ1</td>
<td>adj., superlative</td>
<td>widest</td>
<td>VBZ</td>
<td>verb, 3rd pres</td>
<td>eats</td>
</tr>
<tr>
<td>LS</td>
<td>list item marker</td>
<td>1., 2. One</td>
<td>WDT</td>
<td>wh determiner</td>
<td>which, that</td>
</tr>
<tr>
<td>MD</td>
<td>modal</td>
<td>can, should</td>
<td>WP</td>
<td>wh-pronoun</td>
<td>what, who</td>
</tr>
<tr>
<td>NN</td>
<td>noun, sing. or mass</td>
<td>llamas</td>
<td>WPS</td>
<td>possessive wh-</td>
<td>whose</td>
</tr>
<tr>
<td>NNS</td>
<td>noun, plural</td>
<td>llamas</td>
<td>WRB</td>
<td>wh-adverb</td>
<td>how, where</td>
</tr>
<tr>
<td>NNP</td>
<td>proper noun, singular</td>
<td>IBM</td>
<td>$</td>
<td>dollar sign</td>
<td>$</td>
</tr>
<tr>
<td>NNP5</td>
<td>proper noun, plural</td>
<td>Carolinas</td>
<td>#</td>
<td>pound sign</td>
<td>#</td>
</tr>
<tr>
<td>PDT</td>
<td>predeterminer</td>
<td>all, both</td>
<td>&quot;</td>
<td>left quote</td>
<td>‘ or ”</td>
</tr>
<tr>
<td>POS</td>
<td>possessive ending</td>
<td>’s</td>
<td>”</td>
<td>right quote</td>
<td>‘ or ”</td>
</tr>
<tr>
<td>PRP</td>
<td>personal pronoun</td>
<td>I, you, he</td>
<td>(</td>
<td>left parenthesis</td>
<td>[, {, &lt;</td>
</tr>
<tr>
<td>PRPS</td>
<td>possessive pronoun</td>
<td>your, one’s</td>
<td>)</td>
<td>right parenthesis</td>
<td>], } , &gt;</td>
</tr>
<tr>
<td>RB</td>
<td>adverb</td>
<td>quickly, never</td>
<td>,</td>
<td>comma</td>
<td></td>
</tr>
<tr>
<td>RBR</td>
<td>adverb, comparative</td>
<td>faster</td>
<td>:</td>
<td>sentence-final punc</td>
<td>: ! ?</td>
</tr>
<tr>
<td>RBIS</td>
<td>adverb, superlative</td>
<td>fastest</td>
<td>;</td>
<td>mid-sentence punc</td>
<td>; . .</td>
</tr>
<tr>
<td>RP</td>
<td>particle</td>
<td>up, off</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Using the Penn Tagset

- The grand jury commented on a number of other topics.
Using the Penn Tagset

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

Recall POS Tagging Difficulty

- 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.

These examples from Dekang Lin
How Hard is POS Tagging?  
Measuring Ambiguity

<table>
<thead>
<tr>
<th></th>
<th>87-tag Original Brown</th>
<th>45-tag Treebank Brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unambiguous (1 tag)</td>
<td>44,019</td>
<td>38,857</td>
</tr>
<tr>
<td>Ambiguous (2–7 tags)</td>
<td>5,490</td>
<td>8844</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,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>open, fit, down)</td>
</tr>
<tr>
<td>8 tags</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 tags</td>
<td></td>
<td>4 (‘s, half, back, a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 (that, more, in)</td>
</tr>
</tbody>
</table>

Tagging Whole Sentences with POS is Hard too

- Ambiguous POS contexts
  - E.g., Time flies like an arrow.

- Possible POS assignments
  - Time/[V,N] flies/[V,N] like/[V,Prep] an/Det arrow/N
  - Time/N flies/V like/Prep an/Det arrow/N
  - Time/V flies/N like/Prep an/Det arrow/N
  - Time/N flies/N like/V an/Det arrow/N
  - .....
How Do We Disambiguate POS?

- Many words have only one POS tag (e.g. is, Mary, smallest)
- Others have a single most likely tag (e.g. Dog is less used as a V)
- Tags also tend to co-occur regularly with other tags (e.g. Det, N)
- In addition to conditional probabilities of words \( P(w_i|w_{n-1}) \), we can look at POS likelihoods \( P(t_i|t_{n-1}) \) to disambiguate sentences and to assess sentence likelihoods

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
Overview: POS Tagging Accuracies

- Rough accuracies:
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%
  - Maxent P(t|w): 93.7% / 82.6%
  - Upper bound: ~98% (human)

Review

- Parts of Speech
  - What?

- Part of Speech Tagging
  - What?
  - Why?
  - Easy or hard?
  - Evaluation
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:
- promised:
- to
- back:
- the:
- bill:
Start With a Dictionary

- she: PRP
- promised: VBN,VBD
- to: TO
- back: VB, JJ, RB, NN
- the: DT
- bill: NN, VB

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

POS tag sequences

- Some tag sequences are more likely to occur than others
- POS Ngram view
  https://books.google.com/ngrams/graph?content=_ADJ_+_NOUN_%2C_ADJ_+_NOUN_%28ADV_+_VERB_+

Existing methods often model POS tagging as a sequence tagging problem
**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 \( w_1 \ldots w_n \).

**How do you predict the tags?**

- Two types of information are useful
  - Relations between *words* and *tags*
  - Relations between *tags* and *tags*
    - DT NN, DT JJ NN…
Getting to HMMs  
(Hidden Markov Models)

- We want, out of all sequences of \( n \) tags \( t_1 \ldots t_n \) the single tag sequence such that \( P(t_1 \ldots t_n | w_1 \ldots w_n) \) is highest.

\[
\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)
\]

- Hat \(^\wedge\) means “our estimate of the best one”
- Argmax\(_x\) \( f(x) \) means “the \( x \) such that \( f(x) \) is maximized”

This equation is guaranteed to give us the best tag sequence

\[
\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)
\]

- But how to make it operational? How to compute this value?
- Intuition of Bayesian classification:
  - Use Bayes rule to transform this equation into a set of other probabilities that are easier to compute
Using Bayes Rule

\[ P(x|y) = \frac{P(y|x)P(x)}{P(y)} \]

\[ \hat{t}^n_1 = \underset{t^n_1}{\text{argmax}} \frac{P(w^n_1|t^n_1)P(t^n_1)}{P(w^n_1)} \]

\[ \hat{t}^n_1 = \underset{t^n_1}{\text{argmax}} P(w^n_1|t^n_1)P(t^n_1) \]

Likelihood and Prior

\[ \hat{t}^n_1 = \underset{t^n_1}{\text{argmax}} \frac{\text{likelihood}}{\text{prior}} \]

\[ P(w^n_1|t^n_1) \approx \prod_{i=1}^{n} P(w_i|t_i) \]

\[ P(t^n_1) \approx \prod_{i=1}^{n} P(t_i|t_{i-1}) \]

\[ \hat{t}^n_1 = \underset{t^n_1}{\text{argmax}} P(t^n_1|w^n_1) \approx \underset{t^n_1}{\text{argmax}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}) \]
Two Kinds of Probabilities

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

Two Kinds of Probabilities

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

- Two independent assumptions
  - Approximate $P(t)$ by a bi(or N)-gram model
  - Assume each word depends only on its POS tag

Table representation

<table>
<thead>
<tr>
<th>Transition Matrix $A$</th>
<th>Emission Matrix $B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>$N$</td>
</tr>
<tr>
<td>$D$</td>
<td>0.8</td>
</tr>
<tr>
<td>$N$</td>
<td>0.7</td>
</tr>
<tr>
<td>$V$</td>
<td>0.6</td>
</tr>
<tr>
<td>$A$</td>
<td>0.8</td>
</tr>
<tr>
<td>$.$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$D$</th>
<th>$N$</th>
<th>$V$</th>
<th>$A$</th>
<th>$.$</th>
<th>$\mathbf{th}$</th>
<th>$\mathbf{man}$</th>
<th>$\mathbf{ball}$</th>
<th>$\mathbf{throws}$</th>
<th>$\mathbf{sees}$</th>
<th>$\mathbf{red}$</th>
<th>$\mathbf{blue}$</th>
<th>$.$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N$</td>
<td></td>
<td>0.7</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V$</td>
<td></td>
<td></td>
<td>0.6</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td></td>
<td></td>
<td></td>
<td>0.8</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$.$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Initial state vector $\pi$

<table>
<thead>
<tr>
<th>$D$</th>
<th>$N$</th>
<th>$V$</th>
<th>$A$</th>
<th>$.$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi$</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Let $\lambda = \{A, B, \pi\}$ represents all parameters
**Prediction in generative model**

- **Inference:** What is the most likely sequence of tags for the given sequence of words \( \mathbf{w} \)

- What are the latent states that most likely generate the sequence of word \( \mathbf{w} \)

**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?
Example

- $P(\text{NN}|\text{TO}) = .00047$
- $P(\text{VB}|\text{TO}) = .83$
- $P(\text{race}|\text{NN}) = .00057$
- $P(\text{race}|\text{VB}) = .00012$
- $P(\text{NR}|\text{VB}) = .0027$
- $P(\text{NR}|\text{NN}) = .0012$
- $P(\text{VB}|\text{TO})P(\text{NR}|\text{VB})P(\text{race}|\text{VB}) = .00000027$
- $P(\text{NN}|\text{TO})P(\text{NR}|\text{NN})P(\text{race}|\text{NN}) = .00000000032$

- So we (correctly) choose the verb reading
Hidden Markov Models

- What we’ve described with these two kinds of probabilities is a Hidden Markov Model (HMM).

Definitions

- A weighted finite-state automaton adds probabilities to the arcs
  - The sum of the probabilities leaving any arc must sum to one
- A Markov chain is a special case of a WFSA in which the input sequence uniquely determines which states the automaton will go through
- Markov chains can’t represent inherently ambiguous problems
  - Useful for assigning probabilities to unambiguous sequences
Markov Chain for Weather

Weather continued
Markov Chain for Words

Markov Chain: “First-order observable Markov Model”

- A set of states
  - $Q = q_1, q_2, ..., q_N$; the state at time $t$ is $q_t$
- Transition probabilities:
  - A set of probabilities $A = a_{01}a_{02}...a_{n1}...a_{nn}$.
  - Each $a_{ij}$ represents the probability of transitioning from state $i$ to state $j$.
  - The set of these is the transition probability matrix $A$.
- Current state only depends on previous state

$$P(q_i \mid q_1...q_{i-1}) = P(q_i \mid q_{i-1})$$
Markov Chain for Weather

- What is the probability of 4 consecutive rainy days?
- Sequence is rainy-rainy-rainy-rainy
- I.e., state sequence is 3-3-3-3
- \( P(3,3,3,3) = \)

\[ \pi_3 a_{33} a_{33} a_{33} = 0.2 \times (0.6)^3 = 0.0432 \]
**HMM for Ice Cream**

- You are a climatologist in the year 2799
- Studying climate change
- You can’t find any records of the weather in Pittsburgh for summer of 2019
- But you find a diary
- Which lists how many ice-creams someone ate every date that summer
- Our job: figure out how hot it was

**Hidden Markov Model**

- For Markov chains, the output symbols are the same as the states.
  - See **hot** weather: we’re in state **hot**
- But in part-of-speech tagging (and other things)
  - The output symbols are **words**
  - But the hidden states are **part-of-speech tags**
- So we need an extension!
- A Hidden Markov Model is an extension of a Markov chain in which the input symbols are not the same as the states.
- This means we don’t know which state we are in.
**Hidden Markov Models**

- **States** \( Q = q_1, q_2...q_N; \)
- **Observations** \( O = o_1, o_2...o_N; \)
  - Each observation is a symbol from a vocabulary \( V = \{v_1,v_2...v_V\} \)
- **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 \( B = \{b_i(k)\} \)
    \[ b_i(k) = P(X_t = o_k \mid q_t = i) \]
    \[ \pi_i = P(q_1 = i) \quad 1 \leq i \leq N \]
- **Special initial probability vector** \( \pi \)

**Task**

- **Given**
  - Ice Cream Observation Sequence: 1,2,3,2,2,2,3...
- **Produce:**
Weather/Ice Cream HMM
- Hidden States:
- Transition probabilities:
- Observations:

Weather/Ice Cream HMM
- Hidden States: \{Hot,Cold\}
- Transition probabilities (A Matrix) between H and C
- Observations: \{1,2,3\} # of ice creams eaten per day
HMM for Ice Cream

\[ \pi = [0.8, 0.2] \]

\[ \begin{align*}
\pi_1 & = 0.7 \\
\pi_2 & = 0.6
\end{align*} \]

\[ B_1 = \begin{bmatrix} P(1 \mid \text{HOT}) \\ P(2 \mid \text{HOT}) \\ P(3 \mid \text{HOT}) \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.4 \\ 0.4 \end{bmatrix} \]

\[ B_2 = \begin{bmatrix} P(1 \mid \text{COLD}) \\ P(2 \mid \text{COLD}) \\ P(3 \mid \text{COLD}) \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.4 \\ 0.1 \end{bmatrix} \]

Back to POS Tagging: Transition Probabilities

\[ \begin{align*}
to_{02} & \quad to_{11} \\
to_{12} & \quad to_{21} \\
to_{22} & \quad to_{31} \\
to_{32} & \quad to_{41}
\end{align*} \]

\[ \begin{align*}
to_{01} & \quad to_{13} \\
to_{23} & \quad to_{33} \\
to_{34} & \quad to_{43}
\end{align*} \]

\[ \begin{align*}
to_{03} & \quad to_{14} \\
to_{24} & \quad to_{34} \\
to_{34} & \quad to_{44}
\end{align*} \]
Observation Likelihoods

What can HMMs Do?

- **Likelihood**: Given an HMM $\lambda$ and an observation sequence $O$, determine the likelihood $P(O, \lambda)$: *language modeling*

- **Decoding**: Given an observation sequence $O$ and an HMM $\lambda$, discover the *best* hidden state sequence $Q$: Given seq of ice creams, what was the most likely weather on those days? (*tagging*)

- **Learning**: Given an observation sequence $O$ and the set of states in the HMM, learn the HMM *parameters*
Decoding

- Ok, now we have a complete model that can give us what we need. Recall that we need to get

\[ \hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n) \]

- We could just enumerate all paths given the input and use the model to assign probabilities to each.
  - Not a good idea.
  - In practice: Viterbi Algorithm (dynamic programming)

Viterbi Algorithm

- Intuition: since state transition out of a state only depend on the current state (and not previous states), we can record for each state the optimal path

- We record
  - Cheapest cost to state at step
  - Backtrace for that state to best predecessor
Viterbi Summary

- Create an array
  - With columns corresponding to inputs
  - Rows corresponding to possible states
- Sweep through the array in one pass filling the columns left to right using our transition probs and observations probs
- Dynamic programming key is that we need only store the MAX prob path to each cell (not all paths).

Viterbi Example
Another Viterbi Example

- Analyzing “Fish sleep”
  - Done in class

Evaluation

- So once you have your POS tagger running how do you evaluate it?
  - Overall error rate with respect to a gold-standard test set.
  - Error rates on particular tags
  - Error rates on particular words
  - Tag confusions...
- Need a baseline – just the most frequent tag is 90% accurate!
Error Analysis

- Look at a confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>IN</th>
<th>JJ</th>
<th>NN</th>
<th>NNP</th>
<th>RB</th>
<th>VBD</th>
<th>VBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>.2</td>
<td>.7</td>
<td>.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JJ</td>
<td>.2</td>
<td>3.3</td>
<td>.2</td>
<td>.7</td>
<td></td>
<td>.2</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>8.7</td>
<td></td>
<td></td>
<td>.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNP</td>
<td>.2</td>
<td>3.3</td>
<td>4.1</td>
<td></td>
<td>.2</td>
<td>.2</td>
<td>.2</td>
</tr>
<tr>
<td>RB</td>
<td>2.2</td>
<td>.5</td>
<td></td>
<td>.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VBD</td>
<td>.3</td>
<td>.5</td>
<td></td>
<td></td>
<td></td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td>VBN</td>
<td>2.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.6</td>
</tr>
</tbody>
</table>

- See what errors are causing problems
  - Noun (NN) vs ProperNoun (NNP) vs Adj (JJ)
  - Preterite (VBD) vs Participle (VBN) vs Adjective (JJ)

Evaluation

- The result is compared with a manually coded “Gold Standard”
  - Typically accuracy reaches 96-97%
  - This may be compared with result for a baseline tagger (one that uses no context).
  - Important: 100% is impossible even for human annotators.
More Complex Issues

- Tag indeterminacy: when ‘truth’ isn’t clear
  Caribbean cooking, child seat
- Tagging multipart words
  wouldn’t --> would/MD n’t/RB
- How to handle unknown words
  - Assume all tags equally likely
  - Assume same tag distribution as all other singletons in corpus
  - Use morphology, word length,....

Other Tagging Tasks

- Noun Phrase (NP) Chunking
  [the student] said [the exam] is hard
- Three tabs
  - B = beginning of NP
  - I = continuing in NP
  - O = other word
- Tagging result
  - The/B student/I said/O the/B exam/I is/0 hard/0
Summary

- Parts of speech
- Tagsets
- Part of speech tagging
- Rule-Based, HMM Tagging
  - Other methods later in course