LING 438/538
Computational Linguistics
Sandiway Fong
Lecture 21: 11/13
POS Tagging

- **Task:**
  - assign the right part-of-speech tag, e.g. noun, verb, conjunction, to a word in context

- **POS taggers**
  - need to be *fast* in order to process large corpora
    - *time taken should be no more than proportional to the size of the corpora*
  - POS taggers try to assign the correct tag without actually (fully) parsing the sentence
    - the *walk* : **noun** I took …
    - I *walk* : **verb** 2 miles every day
How Hard is Tagging?

• Easy task to do well on:
  – naïve algorithm
    • assign tag by (unigram) frequency
  – 90% accuracy (Charniak et al., 1993)

• Brown Corpus (Francis & Kucera, 1982):
  – 1 million words
  – 39K distinct words
  – 35K words with only 1 tag
  – 4K with multiple tags (DeRose, 1988)

That’s 89.7% from just considering single tag words, even without getting any multiple tag words right
Penn TreeBank Tagset

• A standard tagset (for English)
  – 48-tag subset of the Brown Corpus tagset
    www.ldc.upenn.edu/doc/treebank2/cl93.html

• Simplifications:
  – Tag **TO**:
    • infinitival marker, preposition
    • I want *to* win
    • I went *to* the store

  – Tag **IN**:
    • preposition: *that*, *when*, *although*
    • I know *that* I should have stopped, *although*…
    • I stopped *when* I saw Bill
Penn TreeBank Tagset

• Simplifications:
  – Tag DT:
    • determiner: *any, some, these, those*
    • any man
    • these *man/men

  – Tag VBP:
    • verb, present: *am, are, walk*
    • Am I here?
    • *Walked I here?/Did I walk here?
Tagging Methods

- 3 Basic Approaches
  - Manual rules
  - Statistical models
  - Machine Learning of rules
Rule-Based POS Tagging

- **ENGTWOL**
  - English morphological analyzer based on two-level morphology (Chapter 3)
  - 56K word stems
  - processing
    - apply morphological engine
    - get all possible tags for each word
    - apply rules (1,100) to eliminate candidate tags
Rule-Based POS Tagging

• see section 8.4
• ENGTWOL tagger (now ENGCG)
  – http://www2.lingsoft.fi/cgi-bin/engcg
Rule-Based POS Tagging

- example in the textbook is:
  - *Pavlov had shown that salivation* ...

  - ... elided material is crucial

"<that>" "that" **<CLB>** CS @CS

"<that>" "that" DET CENTRAL DEM SG @DN>
Rule-Based POS Tagging

- Examples of tags:
  - PCP2 past participle
  - **intransitive**: SV subject verb
  - **ditransitive**: SVOO subject verb object object

<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 NOMINAL DETERMINER</td>
</tr>
<tr>
<td>one-half</td>
<td>NUM</td>
<td>SG</td>
</tr>
<tr>
<td>she</td>
<td>PRON</td>
<td>PERSONAL NOMINATIVE SG</td>
</tr>
<tr>
<td>show</td>
<td>V</td>
<td>IMPERATIVE VFIN</td>
</tr>
<tr>
<td>show</td>
<td>V</td>
<td>PRESENT SG VFIN</td>
</tr>
<tr>
<td>show</td>
<td>N</td>
<td>NOMINATIVE SG</td>
</tr>
<tr>
<td>shown</td>
<td>PCP2</td>
<td>SVOO SVOO 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>

figure 8.8
Rule-Based POS Tagging

- **example**
  - it isn’t **that**:adv odd

- **rule (from pg. 302)**
  - given input “that”
  - if
    - (+1 A/ADV/QUANT)
    - (+2 SENT-LIM)
    - (NOT -1 SVOC/A)
    - then eliminate non-ADV tags
  - else eliminate ADV tag

*cf. I consider **that** odd*
Rule-Based POS Tagging

• examples
  – it isn't **that**: adv odd
  – I consider **that** odd

```
It isn't that odd
(See the description of morphological tags, syntactic tags and other notations.)
"<*it>"
  "it"  느낌 NonMod> PRON NOM SG3 SUBJ @SUBJ
"<is>"
  "be"  느낌 SV <SVC/N> <SVC/A> V PRES SG3 VPIN @PMAINV
"<_n't>"
  "not"  느낌 NEG-PART @NEG
"<that>"
  "that"  느낌 ADV AD-A> @AD-A>
"<odd>"
  "odd"  느낌 A ABS @PCOMPL-S
```

```
I consider that odd
(See the description of morphological tags, syntactic tags and other notations.)
"<*i>"
  "i"  느낌 NonMod> PRON PERS NOM SG1 SUBJ @SUBJ
"<consider>"
  "consider"  느낌 Vcog> <SVOC/N> <SVOC/A> <as/SVOC/A> <SVO> V PRES
"<that>"
  "that"  느낌 PRON DEM SG @OBJ
"<odd>"
  "odd"  느낌 A ABS @PCOMPL-0
```
Rule-Based POS Tagging

- Now ENGCG-2 (4000 rules)
  - don’t see demo online anymore..
Rule-Based POS Tagging

• Now ENGCG-2 (4000 rules)
  – http://www.connexor.eu/technology/machinese/demo/

**English Machinese Phrase Tagger 4.6 analysis:**

<table>
<thead>
<tr>
<th>Text</th>
<th>Baseform</th>
<th>Phrase syntax and part-of-speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pavlov</td>
<td>Pavlov</td>
<td>nominal head, proper noun, single-word noun phrase</td>
</tr>
<tr>
<td>had</td>
<td>have</td>
<td>auxiliary verb, indicative past</td>
</tr>
<tr>
<td>shown</td>
<td>show</td>
<td>main verb, participle perfect</td>
</tr>
<tr>
<td>that</td>
<td>that</td>
<td>prepositional marker, clause marker</td>
</tr>
<tr>
<td>salivation</td>
<td>salivation</td>
<td>nominal head, noun, single-word noun phrase</td>
</tr>
<tr>
<td>can</td>
<td>can</td>
<td>auxiliary verb, indicative present</td>
</tr>
<tr>
<td>be</td>
<td>be</td>
<td>main verb, infinitive</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>premodifier, determiner</td>
</tr>
<tr>
<td>conditioned</td>
<td>conditioned</td>
<td>premodifier, adjective, noun phrase begins</td>
</tr>
<tr>
<td>reflex</td>
<td>reflex</td>
<td>nominal head, noun, noun phrase ends, sentence boundary</td>
</tr>
</tbody>
</table>
Rule-Based POS Tagging

- **best claimed performance of all systems**: 99.7%
  - *no figures are mentioned in textbook*

**Q: What is Connexor technology based on?**

A: There are two basic kinds of language analysis paradigm: the statistical (automatically generated language models from text corpora) and the linguistic (manually coded language models based on intuition and corpora). Connexor technology is based on linguistic methods, and is amply documented and evaluated in international language engineering or computational linguistics conferences and publications (such as ACL and CoLing since early 1990s).

**Q: Why does Connexor use linguistic methods?**

A: For some levels of language analysis, statistical analyzers are relatively quickly implemented and trained, but their quality does not generally suffice for demanding applications where high reliability and informativeness are crucial. Our starting-point was morphologically rich languages where statistical methods have not performed well, so the linguistic option seemed a natural way to go. Numerous more recent evaluations by us and our customers support the view that much higher quality and informativeness can be reached with our methods than what seems possible with mainstream methods.
Rule-Based POS Tagging

- http://www.connexor.com/demo/tagger/
HMM POS Tagging

from section 8.5

• in general, HMM taggers maximize the quantity
  – \( p(\text{word}|\text{tag}) \times p(\text{tag}|\text{previous } n \text{ tags}) \)

• bigram HMM tagger
  – Let \( w_i = ith \) word
  – and \( t_i = \text{tag} \) for the \( ith \) word
  – Then
    • \( t_i = \operatorname{argmax}_j p(t_j|t_{i-1}, w_i) \)
  – Restate as:
    • \( t_i = \operatorname{argmax}_j p(t_j|t_{i-1}) \times p(w_i|t_j) \)
HMM POS Tagging

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

- **tags (Penn)**
  - NNP proper noun (sg)
  - NN noun (sg or mass)
  - NNS noun (pl)
  - VB verb (base)
  - VBZ verb (3rd pers, present)
  - VBP verb (not 3rd pers, present)
  - VBN verb (past participle)
  - DT determiner, IN preposition, JJ adjective, TO to
HMM POS Tagging

- **1st example**
  - ... to/TO race/??
  - suppose race can have tag VB or NN only
  - formula indicates we should compare
  - \( p(VB|TO) \times p(\text{race}|VB) \)
  - \( \text{with } p(NN|TO) \times p(\text{race}|NN) \)
  - tag sequence probability * probability of word given selected tag

- **tag sequence probability**
  - \( p(NN|TO) = 0.021 \)
  - \( p(VB|TO) = 0.34 \)
  - i.e. a verb is more than ten times as likely to follow TO as a noun

- **lexical likelihood**
  - \( p(\text{race}|NN) = 0.00041 \)
  - \( p(\text{race}|VB) = 0.00003 \)
  - i.e. race as a noun is more than ten times as frequent than as a verb

- **calculation**
  - \( p(VB|TO) \times p(\text{race}|VB) = 0.34 \times 0.00003 = 0.000010 \)
  - \( p(NN|TO) \times p(\text{race}|NN) = 0.021 \times 0.00041 = 0.000009 \)
    - (textbook says: 0.000007)
  - very close: choose to/TO race/VB
HMM POS Tagging

- given
  - word sequence $W = w_1 w_2 \ldots w_n$
  - let $T = t_1 t_2 \ldots t_n$ be a tag sequence

- compute
  - $T^* = \text{argmax}_{T \in \tau} p(T|W)$
  - $\tau$ = set of all possible tag sequences

- using Bayes Law
  - $T^* = \text{argmax}_{T \in \tau} p(T)p(W|T)/p(W)$
  - $T^* = \text{argmax}_{T \in \tau} p(T)p(W|T)$
  - $T^* = \text{argmax}_{T \in \tau} p(t_1 \ldots t_n)p(w_1 \ldots w_n \mid t_1 \ldots t_n)$

- Chain Rule
  - $p(t_1 t_2 t_3 \ldots t_n) = p(t_1) p(t_2|t_1) p(t_3|t_1 t_2) \ldots p(t_n|t_1 \ldots t_{n-1})$
  - $p(t_1 t_2 t_3 \ldots t_n) = p(t_1) p(t_2|w_1 t_1) p(t_3|w_1 t_1 w_2 t_2) \ldots p(t_n|w_1 t_1 \ldots w_{n-2} t_{n-2} w_{n-1} t_{n-1})$
  - $p(w_1 w_2 w_3 \ldots w_n | t_1 t_2 \ldots t_n) = p(w_1 | t_1) p(w_2 | w_1 t_1 t_2) p(w_3 | w_1 t_1 w_2 t_2 t_3) \ldots p(w_n | w_1 t_1 \ldots w_{n-2} t_{n-2} w_{n-1} t_{n-1} t_n)$

- hence
  - $T^* = \text{argmax}_{T \in \tau} p(t_1) p(w_1 | t_1) * p(t_2|w_1 t_1) p(w_2|w_1 t_1 t_2) * \ldots * p(t_n|w_1 t_1 \ldots w_{n-2} t_{n-2} w_{n-1} t_{n-1}) p(w_n|w_1 t_1 \ldots w_{n-2} t_{n-2} w_{n-1} t_{n-1} t_n)$

Math details: see section 8.5 (pgs.305–307)

\[
P(x|y) = P(y|x)P(x)/P(y)
\]
HMM POS Tagging

• simplify
  – \( T^* = \text{argmax}_{T \in \tau} \prod_{i=1}^{n} p(w_i | t_i) p(t_{i+1} | w_{i-1} t_i) \prod_{i=2}^{n} \prod_{j=i-1}^{n} p(w_i | w_{i-1} t_{i-1}) \)

• assume
  – probability of a word is dependent only on its tag
  – i.e. \( p(w_1 | t_1) \cdot p(w_2 | w_1 t_1) \cdot \ldots \cdot p(w_n | w_{n-1} t_{n-1} t_n) \)
  – becomes \( p(w_1 | t_1) \cdot p(w_2 | t_2) \cdot \ldots \cdot p(w_n | t_n) \)

• assume
  – trigram approximation for tag history
  – i.e. \( p(t_1) \cdot p(t_2 | w_1 t_1) \cdot \ldots \cdot p(t_n | w_{n-1} t_{n-1}) \)
  – becomes \( p(t_1) \cdot p(t_2 | t_1) \cdot \ldots \cdot p(t_n | t_{n-1}) \)

• formula becomes
  – \( T^* = \text{argmax}_{T \in \tau} \prod_{i=1}^{n} p(t_i) \cdot p(t_{i+1} | t_i) \cdot \prod_{i=2}^{n} \prod_{j=i-1}^{n} p(w_i | w_{i-1} t_{i-1}) \cdot p(w_1 | t_1) \cdot p(w_2 | t_2) \cdot \ldots \cdot p(w_n | t_n) \)
HMM POS Tagging

• formula
  – \( T^* = \arg\max_{T \in \tau} p(t_1) p(t_2 | t_1) \ldots p(t_n | t_{n-2}, t_{n-1}) \cdot p(w_1 | t_1) p(w_2 | t_2) \ldots p(w_n | t_n) \)

• corpus frequencies
  – \( p(t_n | t_{n-2}, t_{n-1}) = \frac{f(t_n, t_{n-2}, t_{n-1})}{f(t_{n-2}, t_{n-1})} \)
  – \( p(w_n | t_n) = \frac{f(w_n, t_n)}{f(t_n)} \)

• assume
  – training corpus is tagged (manually)

• we can use
  – Viterbi (see chapter 7) to evaluate the formula for \( T^* \) in a dynamic programming fashion
  – smoothing to deal with zero frequencies in the training corpus

• results
  – > 96%
    • (Weishedel et al., 1993), (DeRose, 1998)
  – baseline: naive unigram frequency algorithm
    • 90% accuracy (Charniak et al., 1993)
  – rule-based tagger: ENGCG-2 (4000 rules)
    • 99.7%
Transformation-Based POS Tagging (TBT)

section 8.6

• **basic idea** (Brill, 1995)
  – **Tag Transformation Rules:**
    • change a tag to another tag by inspection of local context
    • e.g. *the tag before or after*
  – **initially**
    • use the naïve algorithm to assign tags
  – **train** a system to find these rules
    • with a finite search space of possible rules
    • error-driven procedure
      – repeat until errors are eliminated as far as possible
  – **assume**
    • training corpus is already tagged
      – *needed because of error-driven training procedure*
TBT: Space of Possible Rules

- Fixed window around current tag:

  ![Diagram of window around current tag]

- Prolog-based μ-TBL notation (Lager, 1999):
  - current tag > new tag <- `tag@[+/-N]`
  - “change current tag to new tag if tag at position +/-N”
TBT: Rules Learned

- **Examples of rules learned**
  
  (Manning & Schütze, 1999) (μ-TBL-style format):

  - NN > VB <- \texttt{TO@[-1]}
    - … to walk …
  - VBP > VB <- \texttt{MD@[-1,-2,-3]}
    - … could have put …
  - JJR > RBR <- \texttt{JJ@[1]}
    - … more valuable player …
  - VBP > VB <- n’t@[-1,-2]
    - … did n’t cut …
    - (n’t is a separate word in the corpus)

NN = noun, sg. or mass
VB = verb, base form
VBP = verb, pres. (¬3rd person)
JJR = adjective, comparative
RBR = adverb, comparative
The µ-TBL System

- Implements Transformation-Based Learning
  - Can be used for POS tagging as well as other applications
- Implemented in Prolog
  - code and data
- Downloadable from [http://www.ling.gu.se/~lager/mutbl.html](http://www.ling.gu.se/~lager/mutbl.html)
- Full system for Windows (based on Sicstus Prolog)
  - Includes tagged *Wall Street Journal* corpora
The µ-TBL System

- Tagged Corpus (for training and evaluation)
- Format:
  - \( \text{wd}(P,W) \)
    - \( P \) = index of \( W \) in corpus, \( W \) = word
  - \( \text{tag}(P,T) \)
    - \( T \) = tag of word at index \( P \)
  - \( \text{tag}(T_1,T_2,P) \)
    - \( T_1 \) = tag of word at index \( P \), \( T_2 \) = correct tag
- (For efficient access: Prolog first argument indexing)
The µ-TBL System

• Example of tagged WSJ corpus:
  - wd(63, 'Longer'). tag(63, 'JJR'). tag('JJR', 'JJR', 63).
  - wd(64, 'maturities'). tag(64, 'NNS'). tag('NNS', 'NNS', 64).
  - wd(65, 'are'). tag(65, 'VBP'). tag('VBP', 'VBP', 65).
  - wd(66, 'thought'). tag(66, 'VBN'). tag('VBN', 'VBN', 66).
  - wd(67, 'to'). tag(67, 'TO'). tag('TO', 'TO', 67).
  - wd(68, 'indicate'). tag(68, 'VBP'). tag('VBP', 'VB', 68).
  - wd(69, 'declining'). tag(69, 'VBG'). tag('VBG', 'VBG', 69).
  - wd(70, 'interest'). tag(70, 'NN'). tag('NN', 'NN', 70).
  - wd(71, 'rates'). tag(71, 'NNS'). tag('NNS', 'NNS', 71).
  - wd(72, 'because'). tag(72, 'IN'). tag('IN', 'IN', 72).
  - wd(73, 'they'). tag(73, 'PP'). tag('PP', 'PP', 73).
  - wd(74, 'permit'). tag(74, 'VB'). tag('VB', 'VB', 74).
  - wd(75, 'portfolio'). tag(75, 'NN'). tag('NN', 'NN', 75).
  - wd(76, 'managers'). tag(76, 'NNS'). tag('NNS', 'NNS', 76).
  - wd(77, 'to'). tag(77, 'TO'). tag('TO', 'TO', 77).
  - wd(78, 'retain'). tag(78, 'VB'). tag('VB', 'VB', 78).
  - wd(79, 'relatively'). tag(79, 'RB'). tag('RB', 'RB', 79).
  - wd(80, 'higher'). tag(80, 'JJR'). tag('JJR', 'JJR', 80).
  - wd(81, 'rates'). tag(81, 'NNS'). tag('NNS', 'NNS', 81).
  - wd(82, 'for'). tag(82, 'IN'). tag('IN', 'IN', 82).
  - wd(83, 'a'). tag(83, 'DT'). tag('DT', 'DT', 83).
  - wd(84, 'longer'). tag(84, 'RB'). tag('RB', 'JJR', 84).
The µ-TBL System
The µ-TBL System

DATA STATISTICS:

Corpus Size: 9625
Number of Tags: 9625
Number of Correct Tags: 9228
Number of Errors: 397
Recall: 95.9%
Precision: 95.9%
F-Score: 95.9%
Number of Tags per Word: 1.000

Applied 8 rule(s) for feature(s) [tag] in 0.120 seconds

DATA STATISTICS:

Corpus Size: 9625
Number of Tags: 9625
Number of Correct Tags: 9245
Number of Errors: 380
Recall: 96.1%
Precision: 96.1%
F-Score: 96.1%
Number of Tags per Word: 1.000

Saving the rule sequence(s) in file 'rules/test.pl'.
Generating data for the Error Browser...
Load (or reload) the file "error_data.html" into a HTML browser to view error data.
Finished!
The µ-TBL System

• **Recall**
  – percentage of words that are tagged correctly with respect to the reference (gold-standard)

• **Precision**
  – percentage of words that are tagged correctly with respect to what the tagger emits

• **F-score**
  – combined weighted average of precision and recall
  – Equally weighted:
    • \( \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \)
The µ-TBL System

Concordance

31 occurrences tagged as VBN that should be VBD:

31: VBN>VBD
31: RB>JJ
28: VBD>VBN
22: IN>RB
20: IN>DT
19: NN>VB
15: NN>NN
15: IN>TO
14: VBP>VB
12: IN>DT
11: JJ>NN
10: JJR>RRB
9: RB>IN
9: NN>VBC
8: VB>VB
8: NP>NN
7: NP>NN

Index
The µ-TBL System

• see demo …
  – Off the webpage

• tag transformation rules are
  – human readable
  – more powerful than simple bigrams
  – take less “effort” to train