LING/C SC/PSYC 438/538

Lecture 24

Sandiway Fong

Administrivia

- Homework 4
  - due Wednesday
  - (Text-NSP basics)
- Reading
  - Chapter 5: Part of Speech Tagging

Homework 4

- Part 1 (10 points)
  - **Submit your code** (not the corpus)
  - Use only the text between <text> and </text> markers (*write Perl code*)
    - don't include headers etc.
  - Add words <s> and </s> to mark the start and end of each sentence (*write Perl code*)

Example:

<s> Sun Microsystems Inc. said it will post a larger-than-expected fourth-quarter loss of as much as $26 million and may show a loss in the current first quarter, raising further troubling questions about the once high-flying computer workstation maker. </s>

- Use any n-gram package from CPAN (*or roll your own*) to compute unigram and bigram frequencies for this corpus

e.g. `count.pl` from **Text-NSP**

POS Tagging

- **Task**:
  - assign the right part-of-speech tag, e.g. noun, verb, conjunction, to a word in context
  - **in NLP**: assign one of the 45(48) Penn tags

- **POS taggers**
  - need to be *fast* in order to process large corpora
    - *Linear wrt. size of the corpora*
  - POS taggers assign the correct tag without parsing the sentence
    - the <u>*walk* : **noun**</u> I took …
    - I <u>*walk* : **verb**</u> 2 miles every day

Penn Treebank Part-of-Speech Tags

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

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

- standard tagset (for English)
  - *48-tag subset of the Brown Corpus tagset (87 tags)*
  - http://www.ldc.upenn.edu/Catalog/docs/LDC95T7/cl93.html

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

Table 2:
The Penn Treebank POS tagset

1. CC	Coordinating conjunction	25.TO	to
2. CD	Cardinal number	26.UH	Interjection
3. DT	Determiner	27.VB	Verb, base form
4. EX	Existential there	28.VBD	Verb, past tense
5. FW	Foreign word	29.VBG	Verb, gerund/present participle
6. IN	Preposition/subord.	30.VBN	Verb, past participle
218z	conjunction		
7. JJ	Adjective	31.VBP	Verb, non-3rd ps. sing. present
8. JJR	Adjective, comparative	32.VBZ	Verb, 3rd ps. sing. present
9. JJS	Adjective, superlative	33.WDT	wh-determiner
10.LS	List item marker	34.WP	wh-pronoun
11.MD	Modal	35.WP	Possessive wh-pronoun
12.NN	Noun, singular or mass	36.WRB	wh-adverb
13.NNS	Noun, plural	37. #	Pound sign
14.NNP	Proper noun, singular	38. $	Dollar sign
15.NNPS	Proper noun, plural	39. .	Sentence-final punctuation
16.PDT	Predeterminer	40. ,	Comma
17.POS	Possessive ending	41. :	Colon, semi-colon
18.PRP	Personal pronoun	42. (	Left bracket character
19.PP	Possessive pronoun	43.)	Right bracket character
20.RB	Adverb	44. "	Straight double quote
21.RBR	Adverb, comparative	45. `	Left open single quote
22.RBS	Adverb, superlative	46. "	Left open double quote
23.RP	Particle	47. '	Right close single quote
24.SYM	Symbol (mathematical or scientific)	48. "	Right close double quote

48 tags listed here
36 POS + 12 punctuation

Penn TreeBank Tagset

Part-of-Speech Tagging Guidelines for the Penn Treebank Project

- http://repository.upenn.edu/cgi/viewcontent.cgi?article=1603&context=cis_reports
- Examples:
  - The duchess was entertaining last night.

EXAMPLES: Sampling/NN|VBG data can be fun.
The Duchess was entertaining/JJ|VBG last night.
The Duchess was guarded/JJ|VBN last night.

- From the Penn Treebank itself

```
(VP (TO to)
  (VP (VB put)
    (NP (DT the) (NN genie) )
    (ADVP|PRT (RB back) )
    (PP-PUT (IN in)
      (NP (DT the) (NN bottle) ))))))))))
```

- Treebank (cited by textbook):

(5.4) Mrs./NNP Shaefer/NNP never/RB got/VBD **around/RP** to/TO joining/VBG

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

(5.6) Chateau/NNP Petrus/NNP costs/VBZ **around/RB** 250/CD

Table 2:
The Penn Treebank POS tagset

1.	CC	Coordinating conjunction	25.	TO	to
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential there	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subord. conjunction	30.	VBN	Verb, past participle
218z					
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	wh-determiner
10.	LS	List item marker	34.	WP	wh-pronoun
11.	MD	Modal	35.	WP$	Possessive wh-pronoun
12.	NN	Noun, singular or mass	36.	WRB	wh-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	`	Left open single quote
22.	RBS	Adverb, superlative	46.	"	Left open double quote
23.	RP	Particle	47.	'	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	"	Right close double quote

Tagging Methods

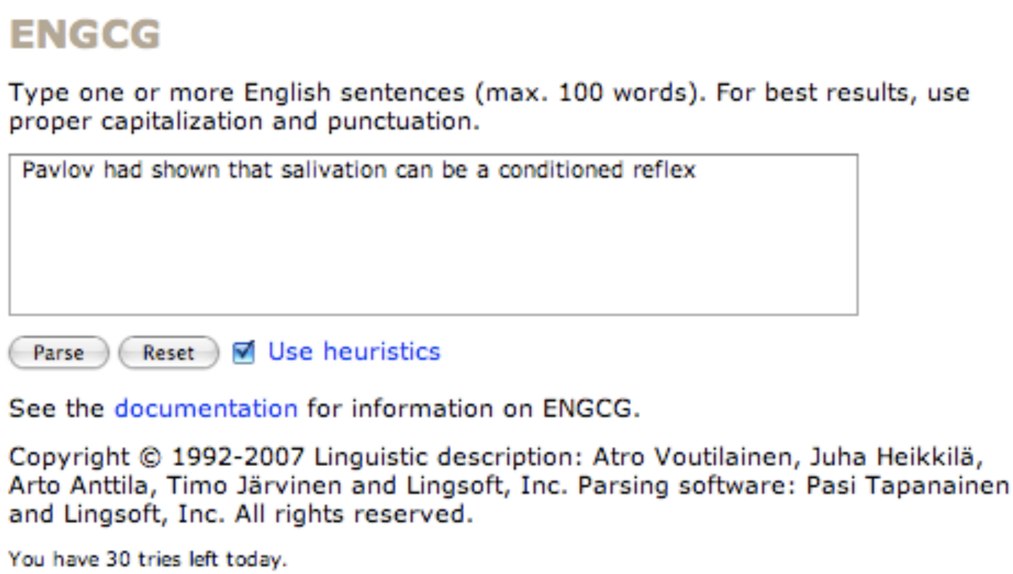
- 3 Basic Approaches
  - Manual rules
  - Machine Learning of rules
  - Statistical models (Hidden Markov Models)

Rule-Based POS Tagging

- **ENGTWOL**
  - English morphological analyzer based on two-level morphology (*recall JM 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 5.4*
- 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

Pavlov had shown that salivation can be a conditioned reflex

(See the description of morphological tags, syntactic tags and other notations.)

```
"<*pavlov>"
        "pavlov" <*> <Proper> N NOM SG  @SUBJ
"<had>"
        "have" <SVO> <SVOC/A> V PAST VFIN  @+FAUXV
"<shown>"
        "show" <Vcog> <SVOO> <SVO> <SV> PCP2  @-FMAINV
"<that>"
        "that" <**CLB> CS @CS
"<salivation>"
        "salivation" N NOM SG  @SUBJ
"<can>"
        "can" V AUXMOD VFIN @+FAUXV
"<be>"
        "be" <SV> <SVC/N> <SVC/A> V INF  @-FMAINV
"<a>"
        "a" <Indef> DET CENTRAL ART SG @DN>
"<conditioned>"
        "condition" <SVO> PCP2  @AN>
"<reflex>"
        "reflex" N NOM SG  @PCOMPL-S
```

```
“<that>""that" <**CLB> CS @CS
```

<CLB>** clause boundary (*who*)

```
"<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

Word	POS	Additional POS features
smaller	ADJ	COMPARATIVE
entire	ADJ	ABSOLUTE ATTRIBUTIVE
fast	ADV	SUPERLATIVE
that	DET	CENTRAL DEMONSTRATIVE SG
all	DET	PREDETERMINER SG/PL QUANTIFIER
dog's	N	GENITIVE SG
furniture	N	NOMINATIVE SG NOINDEFDETERMINER
one-third	NUM	SG
she	PRON	PERSONAL FEMININE NOMINATIVE SG3
show	V	IMPERATIVE VFIN
show	V	PRESENT -SG3 VFIN
show	N	NOMINATIVE SG
shown	PCP2	SVOO SVO SV
occurred	PCP2	SV
occurred	V	PAST VFIN SV

Old textbook figure 8.8

Rule-Based POS Tagging

- **example**
  - it isn't **that**:adv odd
- **rule (from pg. 138)**
  - given input "that"
  - if
    - (+1 A/ADV/QUANT)
    - (+2 SENT-LIM)
    - (NOT -1 SVOC/A)
  - then eliminate non-ADV tags
  - else eliminate ADV tag

that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS

next word (+1)

2nd word (+2)

previous word (-1): verb like *consider*

<SVOC/A> complex transitive with adjective complement (*consider*)

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" <*> <NonMod> PRON NOM SG3 SUBJ @SUBJ
"<is>"
        "be" <SV> <SVC/N> <SVC/A> V PRES SG3 VFIN  @+FMAINV
"<_n't>"
        "not" NEG-PART @NEG
"<that>"
        "that" ADV AD-A> @AD-A>
"<odd>"
        "odd" A ABS  @PCOMPL-S
```

I consider that odd

(See the description of morphological tags, syntactic tags and other notations.)

```
"<*i>"
        "i" <*> <NonMod> PRON PERS NOM SG1 SUBJ @SUBJ
"<consider>"
        "consider" <Vcog> <SVOC/N> <SVOC/A> <as/SVOC/A> <SVO> V PRES -SG3 VFIN @+FMAINV
"<that>"
        "that" PRON DEM SG  @OBJ
"<odd>"
        "odd" A ABS  @PCOMPL-O
```

Rule-Based POS Tagging

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

Technology > Machinese > Demo > Machinese Phrase Tagger - demo

English Machinese Phrase Tagger 4.6 analysis:

Text	Baseform	Phrase syntax and part-of-speech
it	it	nominal head, pro-nominal
is	be	main verb, indicative present
n't	not	adverbial head, adverb
that	that	premodifier, adverb
odd	odd	nominal head, adjective, sentence boundary

Text	Baseform	Phrase syntax and part-of-speech
I	I	nominal head, pro-nominal
consider	consider	main verb, indicative present
that	that	premodifier, adverb
odd	odd	nominal head, adjective, sentence boundary

Rule-Based POS Tagging

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

Rule-Based POS Tagging

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

Technology > Machinese > Demo > Machinese Phrase Tagger - demo

English Machinese Phrase Tagger 4.6 analysis:

Text	Baseform	Phrase syntax and part-of-speech
Pavlov	Pavlov	nominal head, proper noun, single-word noun phrase
had	have	auxiliary verb, indicative past
shown	show	main verb, participle perfect
that	that	preposed marker, clause marker
salivation	salivation	nominal head, noun, single-word noun phrase
can	can	auxiliary verb, indicative present
be	be	main verb, infinitive
a	a	premodifier, determiner
conditioned	conditioned	premodifier, adjective, noun phrase begins
reflex	reflex	nominal head, noun, noun phrase ends, sentence boundary

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).

back

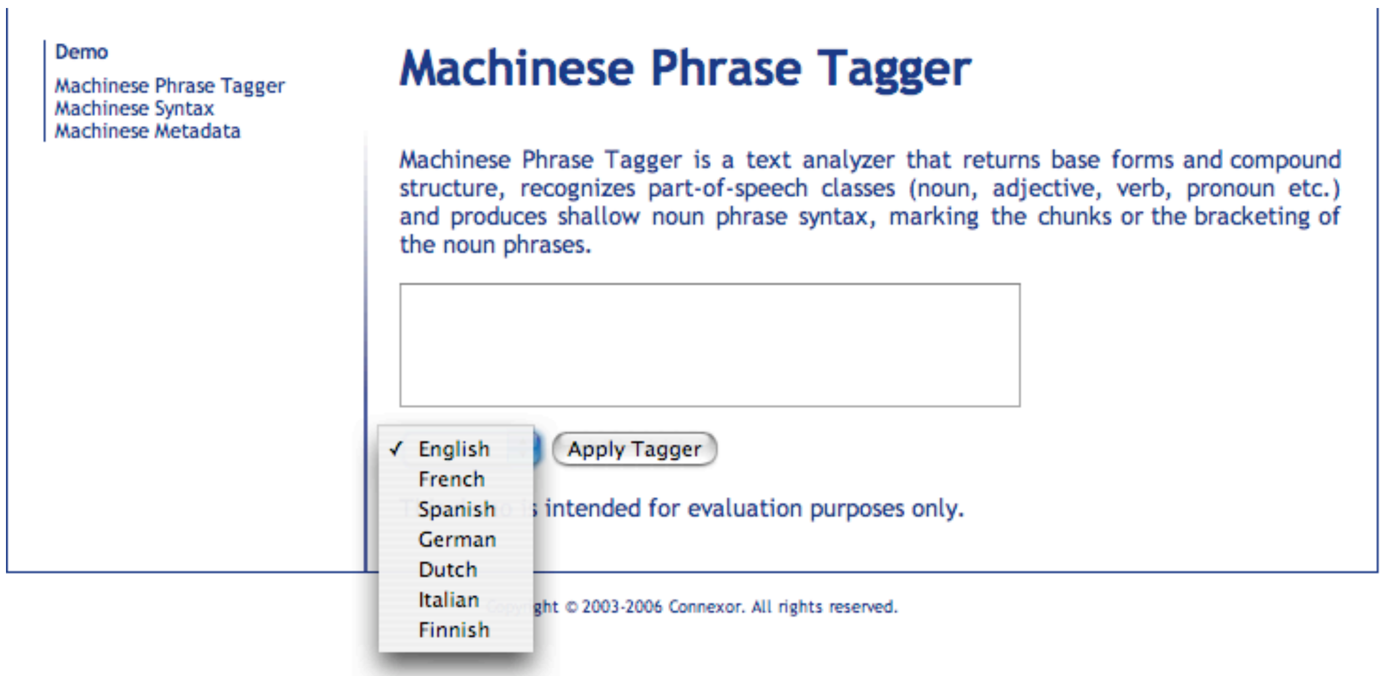
Q: Why does Connexor use linguistic methods?

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.

statistical/
linguistic
divide

Rule-Based POS Tagging

- `http://www.connexor.com/demo/tagger/`



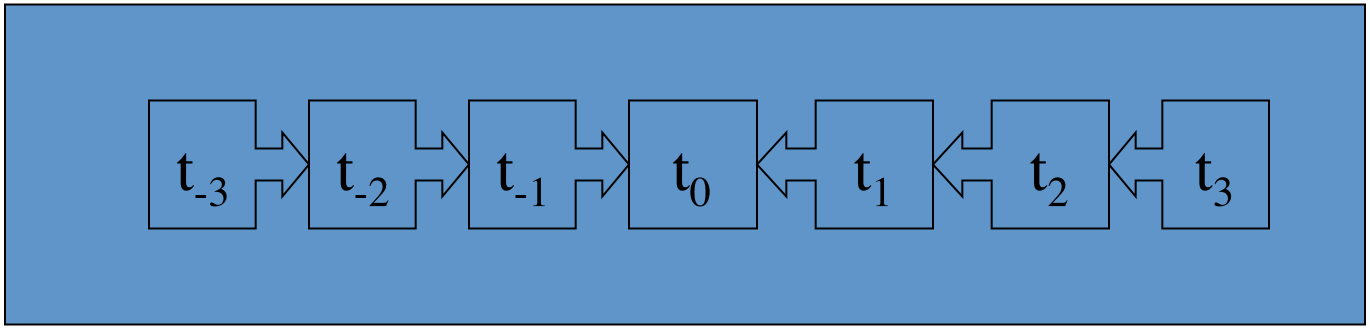
Transformation-Based POS Tagging (TBT)

section 5.6

- **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:



- Prolog-based µ-TBL notation (Lager, 1999):
  - t_0 > t_0’ <- t@[+/-N]
  - *“change current tag* t_0 *to new tag* t_0’ *if word at position +/-N has tag* t”

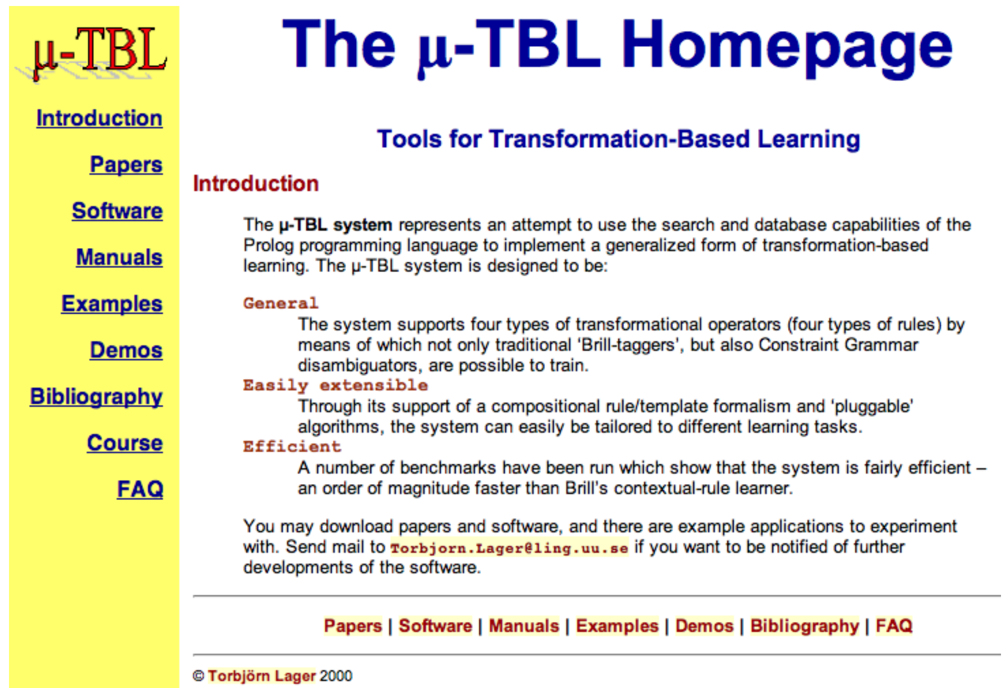
TBT: Rules Learned

- **Examples of rules learned** (Manning & Schütze, 1999) (µ-TBL-style format):
  - NN > VB <- TO@[-1]
    - … to **walk** …
  - VBP > VB <- MD@[-1,-2,-3]
    - … could have **put** …
  - JJR > RBR <- 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
- http://www.ling.gu.se/~lager/mutbl.html
- Full system for Windows (based on Sicstus Prolog)
  - Includes tagged *Wall Street Journal* corpora



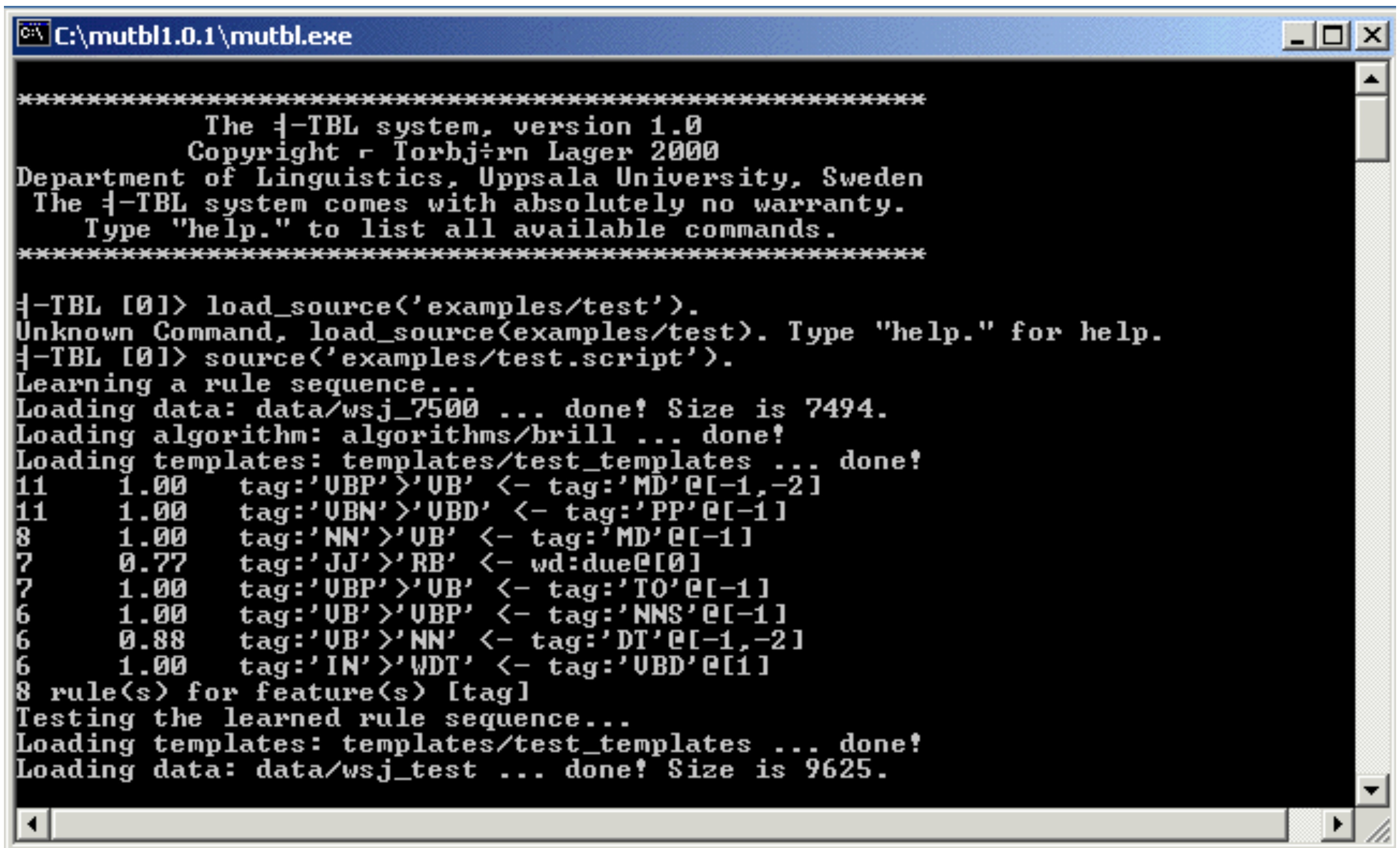
The μ-TBL System

- Tagged Corpus (for training and evaluation)
- Format:
  - wd(P,W)
    - P = index of W in corpus, W = word
  - tag(P,T)
    - T = tag of word at index P
  - 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

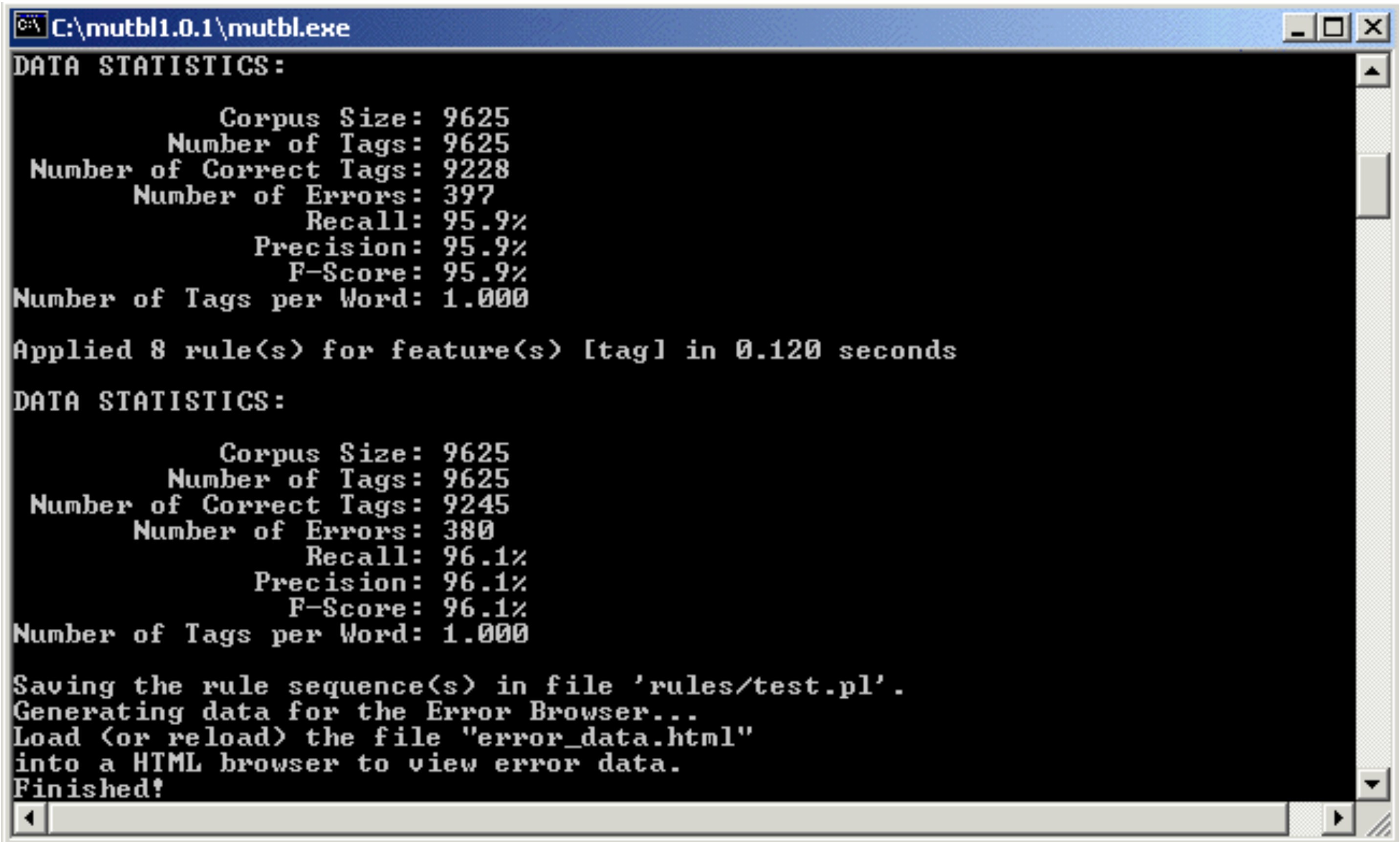
- Example of tagged WSJ corpus:
  - `wd(63,'Longer').   tag(63,'JJR'). tag('JJR','JJR',63).`
  - `wd(64,maturities). tag(64,'NNS'). tag('NNS','NNS',64).`
  - `wd(65,are).        tag(65,'VBP'). tag('VBP','VBP',65).`
  - `wd(66,thought).    tag(66,'VBN'). tag('VBN','VBN',66).`
  - `wd(67,to).         tag(67,'TO').  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','VBP',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



```
C:\mutbl1.0.1\mutbl.exe
*****************************************************
          The ╡-TBL system, version 1.0
         Copyright ⌐ Torbj÷rn Lager 2000
Department of Linguistics, Uppsala University, Sweden
 The ╡-TBL system comes with absolutely no warranty.
    Type "help." to list all available commands.
*****************************************************

╡-TBL [0]> load_source('examples/test').
Unknown Command, load_source(examples/test). Type "help." for help.
╡-TBL [0]> source('examples/test.script').
Learning a rule sequence...
Loading data: data/wsj_7500 ... done! Size is 7494.
Loading algorithm: algorithms/brill ... done!
Loading templates: templates/test_templates ... done!
11    1.00   tag:'VBP'>'VB' <- tag:'MD'@[-1,-2]
11    1.00   tag:'VBN'>'VBD' <- tag:'PP'@[-1]
8     1.00   tag:'NN'>'VB' <- tag:'MD'@[-1]
7     0.77   tag:'JJ'>'RB' <- wd:due@[0]
7     1.00   tag:'VBP'>'VB' <- tag:'TO'@[-1]
6     1.00   tag:'VB'>'VBP' <- tag:'NNS'@[-1]
6     0.88   tag:'VB'>'NN' <- tag:'DT'@[-1,-2]
6     1.00   tag:'IN'>'WDT' <- tag:'VBD'@[1]
8 rule(s) for feature(s) [tag]
Testing the learned rule sequence...
Loading templates: templates/test_templates ... done!
Loading data: data/wsj_test ... done! Size is 9625.
```

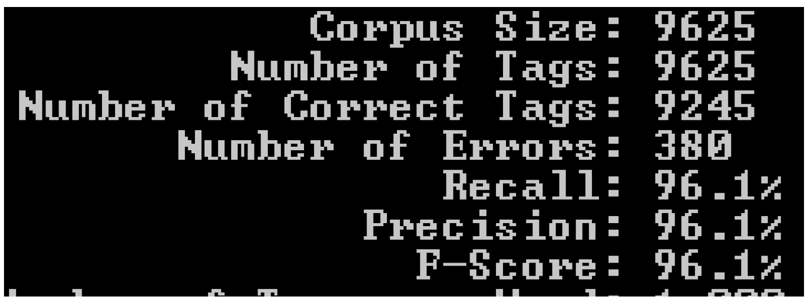
The µ-TBL System



C:\mutbl1.0.1\mutbl.exe

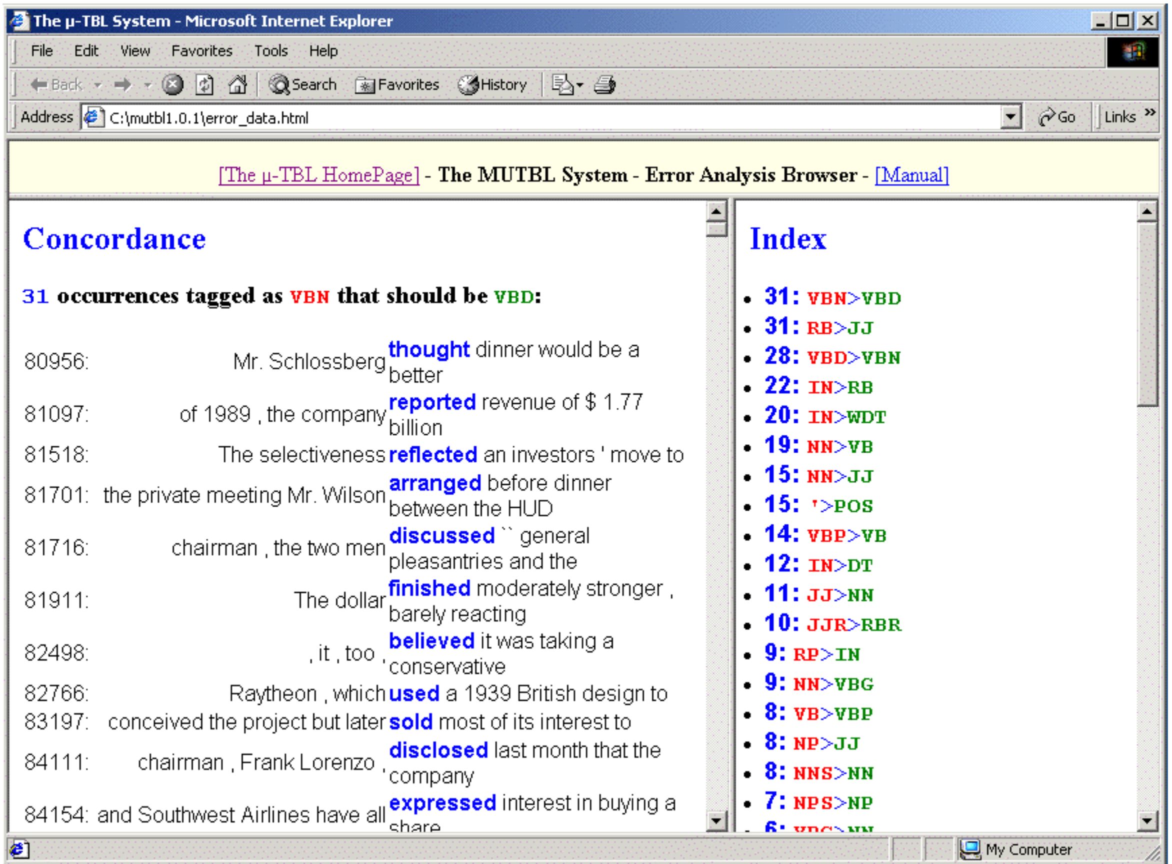
```
DATA STATISTICS:

            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:
    - 2*Precision*Recall/(Precison +Recall)

The μ-TBL System



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

Statistical POS Tagging

- Section 5.5
  - describes HMM POS Tagging
- Personally, I've used the MXPOST tagger in my work
  - Java code (portable) and freely available
  - Maximum entropy tagging
  - Reference:
    - Adwait Ratnaparkhi. A Maximum Entropy Part-Of-Speech Tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, May 17-18, 1996.
    - http://www.inf.ed.ac.uk/resources/nlp/local_doc/mxpost_doc.pdf

HMM POS Tagging

- **Recall, given a word sequence**
  - $w_1\ w_2\ w_3 \ldots w_n$
- **chain rule**
  - *how to compute the probability of a sequence of words*
  - $p(w_1\ w_2\ w_3 \ldots w_n) = p(w_1)\ p(w_2|w_1)\ p(w_3|w_1 w_2) \ldots p(w_n|w_1 \ldots w_{n-2}\ w_{n-1})$
- **Bigram approximation**
  - *just look at the previous word only* (*not all the proceedings words*)
  - **Markov Assumption**: **finite length history (**1st order Markov Model)
  - $\mathbf{p(w_1\ w_2\ w_3 \ldots w_n) \approx p(w_1)\ p(w_2|w_1)\ p(w_3|w_2) \ldots p(w_n|w_{n-1})}$

We can apply the chain rule and bigram approximation to sequences of tags if corpus contains POS tagged words

Compute the best $t_1\ t_2\ t_3 \ldots t_n$ given $w_1\ w_2\ w_3 \ldots w_n$
i.e. *find best tag sequence for sentence*

Maximize P(*tag sequence*|observed word sequence)

HMM POS Tagging

- in general, HMM taggers maximize the quantity
  - p(word|tag) * p(tag|previous *n* tags)

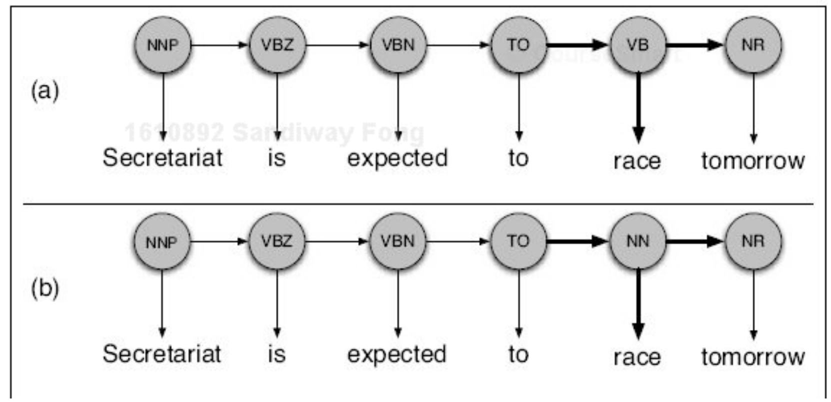
- **bigram HMM tagger**
  - Let w_i = *ith* word
  - and t_i = tag for the *ith* word
  - Then
    - $t_i = \text{argmax}_j \; p(t_j | t_{i-1}, w_i)$
  - Restate as:
    - $t_i = \text{argmax}_j \; p(t_j | t_{i-1}) * p(w_i | t_j)$

bigram formula:
$t_i = \text{argmax}_j\ p(t_j|t_{i-1}) * p(w_i|t_j)$

HMM POS Tagging

- **1st edition**
  - ... to/TO **race/??**
  - suppose *race* can have tag VB or NN only
  - formula indicates we should compare
  - p(VB|TO) * p(race|VB)
  - with p(NN|TO) * p(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(race|NN) = 0.00041
  - p(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) * p(race|VB) = 0.34 * 0.00003 = 0.000010
  - p(NN|TO) * p(race|NN) = 0.021 * 0.00041 = 0.000009
    - (textbook says: *0.000007*)
  - *very close: choose* to/TO **race/VB**

- **2nd edition**



$$P(\text{race}|\text{NN}) = .00057$$
$$P(\text{race}|\text{VB}) = .00012$$

Finally, we need to represent the tag sequence probability for the following tag (in this case the tag NR for *tomorrow*):

$$P(\text{NR}|\text{VB}) = .0027$$
$$P(\text{NR}|\text{NN}) = .0012$$

If we multiply the lexical likelihoods with the tag sequence probabilities, we see that the probability of the sequence with the VB tag is higher and the HMM tagger correctly tags *race* as a VB in Fig. 5.12 despite the fact that it is the less likely sense of *race*:

$$P(\text{VB}|\text{TO})P(\text{NR}|\text{VB})P(\text{race}|\text{VB}) = .00000027$$
$$P(\text{NN}|\text{TO})P(\text{NR}|\text{NN})P(\text{race}|\text{NN}) = .00000000032$$

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**
  - $T^* = \text{argmax}_{T\in\tau}\ p(T|W)$
  - τ = set of all possible tag sequences
- **using Bayes Law**
  - $T^* = \text{argmax}_{T\in\tau}\ p(T)p(W|T)/p(W)$
  - $T^* = \text{argmax}_{T\in\tau}\ p(T)p(W|T)$ (*p(W) a constant here*) $P(x|y) = P(y|x)P(x)/P(y)$
  - $T^* = \text{argmax}_{T\in\tau}\ p(t_1 \ldots t_n)p(w_1 \ldots w_n \mid t_1 \ldots t_n)$
- **Chain Rule**
  - $p(t_1\ t_2\ t_3 \ldots t_n) = p(t_1)\ p(t_2|t_1)\ p(t_3|t_1t_2) \ldots\ p(t_n|t_1 \ldots t_{n-2}\ t_{n-1})$
  - $p(t_1\ t_2\ t_3 \ldots t_n) = p(t_1)\ p(t_2|\boldsymbol{w_1}t_1)\ p(t_3|\boldsymbol{w_1}t_1\boldsymbol{w_2}t_2) \ldots\ p(t_n|\boldsymbol{w_1}t_1 \ldots \boldsymbol{w_{n-2}}t_{n-2}\ \boldsymbol{w_{n-1}}t_{n-1})$
  - $p(w_1\ w_2\ w_3 \ldots w_n\ |t_1\ t_2 \ldots t_n) = p(w_1\ |t_1)\ p(w_2|w_1t_1t_2)\ p(w_3|w_1t_1w_2t_2t_3) \ldots\ p(w_n|w_1\ t_1 \ldots w_{n-2}\ t_{n-2}\ w_{n-1}\ t_{n-1}\ t_n)$
- **hence**
  - $T^* = \text{argmax}_{T\in\tau}\ p(t_1)\ p(w_1\ |t_1) * p(t_2|w_1t_1)\ p(w_2|w_1t_1t_2) * \ldots * p(t_n|w_1t_1 \ldots w_{n-2}t_{n-2}\ w_{n-1}t_{n-1})\ p(w_n|w_1\ t_1 \ldots w_{n-2}\ t_{n-2}\ w_{n-1}\ t_{n-1}\ t_n)$

HMM POS Tagging

- **simplify**
  - $T^* = \text{argmax}_{T\in\tau}\, p(t_1)\, p(w_1 | t_1) * p(t_2|w_1t_1)\, p(w_2|w_1t_1t_2) * \ldots * p(t_n|w_1t_1\ldots w_{n-2}t_{n-2}\, w_{n-1}t_{n-1})\, p(w_n|w_1 t_1\ldots w_{n-2}\, t_{n-2}\, w_{n-1}\, t_{n-1}\, t_n)$
- **assume**
  - probability of a word is dependent only on its tag
  - i.e. $p(w_1 | t_1)\, p(w_2|w_1t_1t_2) \ldots p(w_n|w_1\, t_1\ldots w_{n-2}\, t_{n-2}\, w_{n-1}\, t_{n-1}\, t_n)$
  - becomes $p(w_1 | t_1)\, p(w_2|t_2) \ldots p(w_n|t_n)$
- **assume**
  - *trigram approximation for tag history*
  - i.e. $p(t_1)\, p(t_2|w_1t_1) \ldots p(t_n|w_1t_1\ldots w_{n-2}t_{n-2}\, w_{n-1}t_{n-1})$
  - becomes $p(t_1)\, p(t_2|t_1) \ldots p(t_n|\mathrm{t}_{n-2}\, t_{n-1})$
- **formula becomes**
  - **$T^* = \text{argmax}_{T\in\tau}\, p(t_1)\, p(t_2|t_1) \ldots p(t_n|\mathrm{t}_{n-2}\, t_{n-1}) * p(w_1 | t_1)\, p(w_2|t_2) \ldots p(w_n|t_n)$**

HMM POS Tagging

- **formula**
  - $T^* = \text{argmax}_{T \in \tau}\ p(t_1)\ p(t_2|t_1) \ldots p(t_n|t_{n-2}\, t_{n-1}) * p(w_1\,|t_1)\ p(w_2|t_2) \ldots p(w_n|t_n)$
- **corpus frequencies**
  - $p(t_n|t_{n-2}\, t_{n-1}) = f(t_{n-2}\, t_{n-1} t_n) / f(t_{n-2}\, t_{n-1})$
  - $p(w_n|t_n) = f(w_n, t_n) / f(t_n)$
- **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%