Part Of Speech Tagging

Le Anh Cuong

Reading

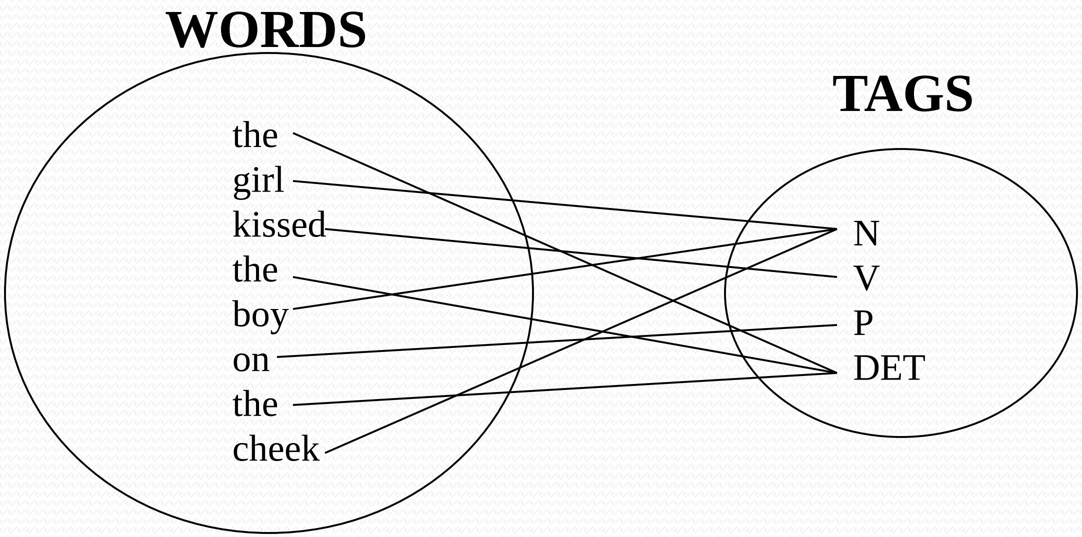
- Chapter 5 [1]
- Chapter 10 [2]

Outline

- Definition and Example
- Motivation
- Word Classes
- Rule-based Tagging
- Stochastic Tagging
- Transformation-Based Tagging
- Tagging Unknown Words

Definition

"The process of assigning a part-of-speech or other lexical class marker to each word in a corpus" (Jurafsky and Martin)



An Example

WORD	LEMMA	TAG
the	the	+DET
girl	girl	+NOUN
kissed	kiss	+VPAST
the	the	+DET
boy	boy	+NOUN
on	on	+PREP
the	the	+DET
cheek	cheek	+NOUN

From: http://www.xrce.xerox.com/competencies/content-analysis/fsnlp/tagger.en.html

Motivation

- Speech synthesis — pronunciation
- Speech recognition — class-based N-grams
- Information retrieval — stemming, selection high-content words
- Word-sense disambiguation
- Corpus analysis of language & lexicography

Word Classes

- Basic word classes: Noun, Verb, Adjective, Adverb, Preposition, …
- POS based on morphology and syntax
- Open vs. Closed classes
  - Open:
    - Nouns, Verbs, Adjectives, Adverbs.
  - Closed:
    - determiners: a, an, the
    - pronouns: she, he, I
    - prepositions: on, under, over, near, by, …

Open Class Words

- Every known human language has nouns and verbs
- Nouns: people, places, things
  - Classes of nouns
    - proper vs. common
    - count vs. mass
- Verbs: actions and processes
- Adjectives: properties, qualities
- Adverbs: hodgepodge!
  - *Unfortunately*, John walked *home extremely slowly yesterday*

Closed Class Words

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

Prepositions from CELEX

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

English Single-Word Particles

aboard	aside	besides	forward(s)	opposite	through
about	astray	between	home	out	throughout
above	away	beyond	in	outside	together
across	back	by	inside	over	under
ahead	before	close	instead	overhead	underneath
alongside	behind	down	near	past	up
apart	below	east, etc.	off	round	within
around	beneath	eastward(s),etc.	on	since	without

Pronouns in CELEX

it	199,920	how	13,137	yourself	2,437	no one	106
I	198,139	another	12,551	why	2,220	wherein	58
he	158,366	where	11,857	little	2,089	double	39
you	128,688	same	11,841	none	1,992	thine	30
his	99,820	something	11,754	nobody	1,684	summat	22
they	88,416	each	11,320	further	1,666	suchlike	18
this	84,927	both	10,930	everybody	1,474	fewest	15
that	82,603	last	10,816	ourselves	1,428	thyself	14
she	73,966	every	9,788	mine	1,426	whomever	11
her	69,004	himself	9,113	somebody	1,322	whosoever	10
we	64,846	nothing	9,026	former	1,177	whomsoever	8
all	61,767	when	8,336	past	984	wherefore	6
which	61,399	one	7,423	plenty	940	whereat	5
their	51,922	much	7,237	either	848	whatsoever	4
what	50,116	anything	6,937	yours	826	whereon	2
my	46,791	next	6,047	neither	618	whoso	2
him	45,024	themselves	5,990	fewer	536	aught	1
me	43,071	most	5,115	hers	482	howsoever	1
who	42,881	itself	5,032	ours	458	thrice	1
them	42,099	myself	4,819	whoever	391	wheresoever	1
no	33,458	everything	4,662	least	386	you-all	1
some	32,863	several	4,306	twice	382	additional	0
other	29,391	less	4,278	theirs	303	anybody	0
your	28,923	herself	4,016	wherever	289	each other	0
its	27,783	whose	4,005	oneself	239	once	0
our	23,029	someone	3,755	thou	229	one another	0
these	22,697	certain	3,345	'un	227	overmuch	0
any	22,666	anyone	3,318	ye	192	such and such	0
more	21,873	whom	3,229	thy	191	whate'er	0
many	17,343	enough	3,197	whereby	176	whenever	0
such	16,880	half	3,065	thee	166	whereof	0
those	15,819	few	2,933	yourselves	148	whereto	0
own	15,741	everyone	2,812	latter	142	whereunto	0
us	15,724	whatever	2,571	whichever	121	whichsoever	0

Conjunctions

and	514,946	yet	5,040	considering	174	forasmuch as	0
that	134,773	since	4,843	lest	131	however	0
but	96,889	where	3,952	albeit	104	immediately	0
or	76,563	nor	3,078	providing	96	in as far as	0
as	54,608	once	2,826	whereupon	85	in so far as	0
if	53,917	unless	2,205	seeing	63	inasmuch as	0
when	37,975	why	1,333	directly	26	insomuch as	0
because	23,626	now	1,290	ere	12	insomuch that	0
so	12,933	neither	1,120	notwithstanding	3	like	0
before	10,720	whenever	913	according as	0	neither nor	0
though	10,329	whereas	867	as if	0	now that	0
than	9,511	except	864	as long as	0	only	0
while	8,144	till	686	as though	0	provided that	0
after	7,042	provided	594	both and	0	providing that	0
whether	5,978	whilst	351	but that	0	seeing as	0
for	5,935	suppose	281	but then	0	seeing as how	0
although	5,424	cos	188	but then again	0	seeing that	0
until	5,072	supposing	185	either or	0	without	0

Auxiliaries

can	70,930	might	5,580	shouldn't	858
will	69,206	couldn't	4,265	mustn't	332
may	25,802	shall	4,118	'll	175
would	18,448	wouldn't	3,548	needn't	148
should	17,760	won't	3,100	mightn't	68
must	16,520	'd	2,299	oughtn't	44
need	9,955	ought	1,845	mayn't	3
can't	6,375	will	862	dare	??
have	???				

Word Classes: Tag Sets

- Vary in number of tags: a dozen to over 200
- Size of tag sets depends on language, objectives and purpose
  - Some tagging approaches (e.g., constraint grammar based) make fewer distinctions e.g., conflating prepositions, conjunctions, particles
  - Simple morphology = more ambiguity = fewer tags

Word Classes: Tag set example

	Tag	Description	Example	Tag	Description	Example
	CC	Coordin. 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, past tense	*ate*
	IN	Preposition/sub-conj	*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*	WP$	Possessive wh-	*whose*
	NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
	NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
	NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
	PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
	POS	Possessive ending	*’s*	”	Right quote	(’ or ”)
PRP	PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PRP$	PP$	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*			

Example of Penn Treebank Tagging of Brown Corpus Sentence

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

VB DT NN .
Book that flight .

VBZ DT NN VB NN ?
Does that flight serve dinner ?

The Problem

- Words often have more than one word class: *this*
  - *This* is a nice day = PRP
  - *This* day is nice = DT
  - You can go *this* far = RB

Word Class Ambiguity (in the Brown Corpus)

Unambiguous (1 tag): 35,340

Ambiguous (2-7 tags): 4,100

2 tags	3,760
3 tags	264
4 tags	61
5 tags	12
6 tags	2
7 tags	1

(Derose, 1988)

Part-of-Speech Tagging

- Rule-Based Tagger: ENGTWOL
- Stochastic Tagger: HMM-based
- Transformation-Based Tagger: Brill

Rule-Based Tagging

- Basic Idea:
  - Assign all possible tags to words
  - Remove tags according to set of rules of type: *if word+1 is an adj, adv, or quantifier and the following is a sentence boundary and word-1 is not a verb like "consider" then eliminate non-adv else eliminate adv.*
  - Typically more than 1000 hand-written rules, but may be machine-learned.

Sample ENGTWOL Lexicon

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

Stage 1 of ENGTWOL Tagging

- First Stage: Run words through Kimmo-style morphological analyzer to get all parts of speech.
- Example: *Pavlov had shown that salivation …*

Pavlov	**PAVLOV N NOM SG PROPER**
had	**HAVE V PAST VFIN SVO**
	HAVE PCP2 SVO
shown	**SHOW PCP2 SVOO SVO SV**
that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	**N NOM SG**

Stage 2 of ENGTWOL Tagging

- Second Stage: Apply constraints.
- Constraints used in negative way.
- Example: Adverbial "that" rule
  Given input: "that"
  If
  (+1 A/ADV/QUANT)
  (+2 SENT-LIM)
  (NOT -1 SVOC/A)
  Then eliminate non-ADV tags
  Else eliminate ADV

Stochastic Tagging

- Based on probability of certain tag occurring given various possibilities
- Necessitates a training corpus
- No probabilities for words not in corpus.
- Training corpus may be too different from test corpus.

Stochastic Tagging (cont.)

Simple Method: Choose most frequent tag in training text for each word!

- Result: 90% accuracy
- Why?
- Baseline: Others will do better
- HMM is an example

HMM Tagger

- Intuition: Pick the most likely tag for this word.
- HMM Taggers choose tag sequence that maximizes this formula:
  - P(word|tag) × P(tag|previous n tags)
- Let $T = t_1, t_2, \ldots, t_n$
  Let $W = w_1, w_2, \ldots, w_n$
  Find POS tags that generate a sequence of words, i.e., look for most probable sequence of tags T underlying the observed words W.

Start with Bigram-HMM Tagger

- $\text{argmax}_T\ P(T|W)$
- $\text{argmax}_T P(T)P(W|T)$
- $\text{argmax}_t P(t_1 \ldots t_n)P(w_1 \ldots w_n|t_1 \ldots t_n)$
- $\text{argmax}_t [P(t_1)P(t_2|t_1) \ldots P(t_n|t_{n-1})][P(w_1|t_1)P(w_2|t_2) \ldots P(w_n|t_n)]$
- To tag a single word: $t_i = \text{argmax}_j\ P(t_j|t_{i-1})P(w_i|t_j)$
- How do we compute $P(t_i|t_{i-1})$?
  - $c(t_{i-1}t_i)/c(t_{i-1})$
- How do we compute $P(w_i|t_i)$?
  - $c(w_i,t_i)/c(t_i)$
- How do we compute the most probable tag sequence?
  - Viterbi

An 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
- to/TO race/???
  the/DT race/???
- $t_i = \text{argmax}_j \; P(t_j|t_{i-1})P(w_i|t_j)$
  - max[P(VB|TO)P(race|VB) , P(NN|TO)P(race|NN)]
- Brown:
  - P(NN|TO) = .021 × P(race|NN) = .00041 = .000007
  - P(VB|TO) = .34 × P(race|VB) = .00003 = .00001

An Early Approach to Statistical POS Tagging

- PARTS tagger (Church, 1988): Stores probability of tag given word instead of word given tag.
- $P(\text{tag}|\text{word}) \times P(\text{tag}|\text{previous n tags})$
- Compare to:
  $P(\text{word}|\text{tag}) \times P(\text{tag}|\text{previous n tags})$
- Consider this alternative (on your own).

http://www.comp.lancs.ac.uk/ucrel/claws/trial.html

Transformation-Based Tagging (Brill Tagging)

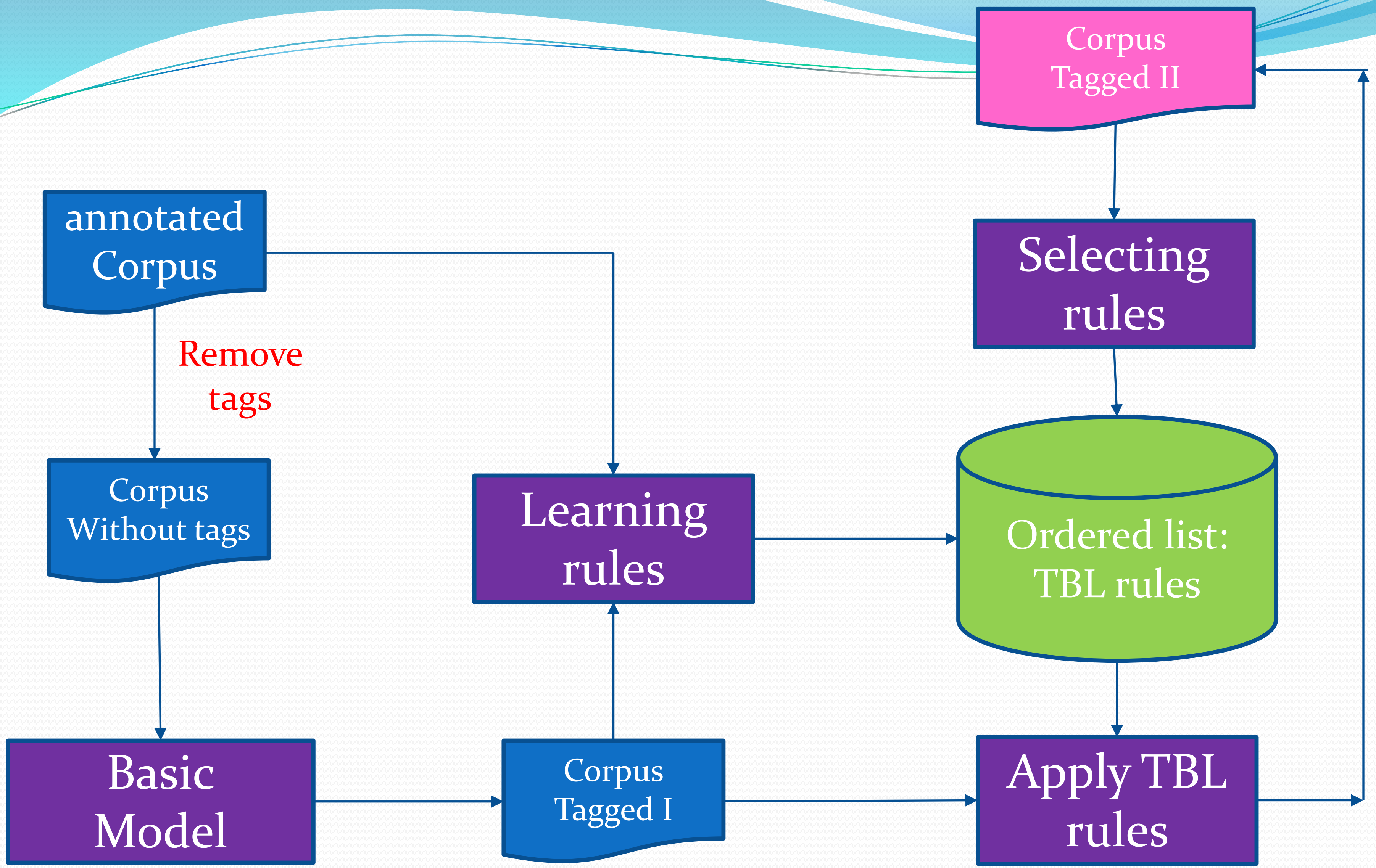
- Combination of Rule-based and stochastic tagging methodologies
  - Like rule-based because rules are used to specify tags in a certain environment
  - Like stochastic approach because machine learning is used—with tagged corpus as input
- Input:
  - tagged corpus
  - dictionary (with most frequent tags)

Transformation-Based Tagging (cont.)

- 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:
  - Write a set of rule templates
  - Among the set of rules, find one with highest score
  - Continue from 2 until lowest score threshold is passed
  - Keep the ordered set of rules
- Rules make errors that are corrected by later rules

TBL Rule Application

- Tagger labels every word with its most-likely tag
  - For example: *race* has the following probabilities in the Brown corpus:
    - *P(NN|race) = .98*
    - *P(VB|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*

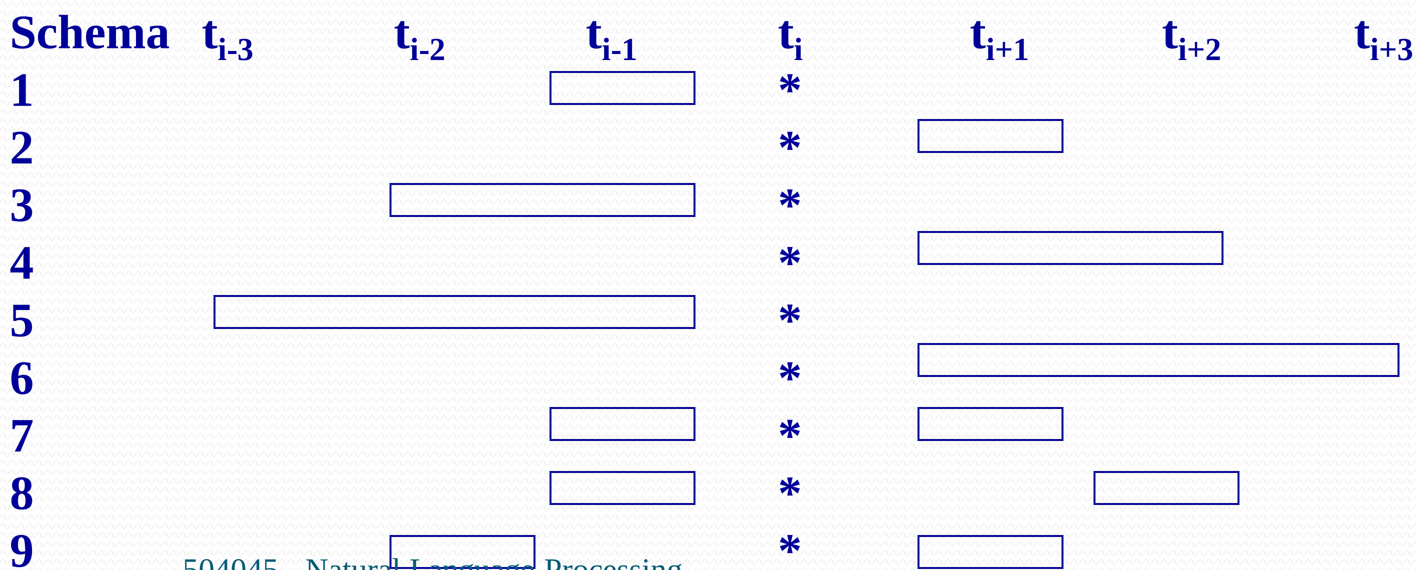

Corpus
Tagged II
annotated
Corpus
Selecting
rules
Remove
tags
Corpus
Without tags
Learning
rules
Ordered list:
TBL rules
Basic
Model
Corpus
Tagged I
Apply TBL
rules

TBL: The 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

TBL: Rule Learning

- 2 parts to a rule
  - Triggering environment
  - Rewrite rule
- The range of triggering environments of templates *(from manning & Schutze 1999:363)*



TBL: Rule Learning (cont.)

- 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**.

#	Change tags From	To	Condition	Example
1	NN	VB	Previous tag is TO	to/TO race/NN → VB
2	VBP	VB	One of the previous 3 tags is MD	might/MD vanish/VBP → VB
3	NN	VB	One of the previous 2 tags is MD	might/MD not reply/NN → VB
4	VB	NN	One of the previous 2 tags is DT	
5	VBD	VBN	One of the previous 3 tags is VBZ	

TBL: Problems

- First 100 rules achieve 96.8% accuracy
  First 200 rules achieve 97.0% accuracy
- Execution Speed: TBL tagger is slower than HMM approach
- Learning Speed: Brill's implementation can take over a day (600k tokens)

BUT …

(1) Learns small number of simple, non-stochastic rules

(2) Can be made to work faster with FST

(3) Best performing algorithm on unknown words

Tagging Unknown Words

- New words added to (newspaper) language 20+ per month
- Plus many proper names …
- Increases error rates by 1-2%
- Method 1: assume they are nouns
- Method 2: assume the unknown words have a probability distribution similar to words only occurring once in the training set.
- Method 3: Use morphological information, e.g., words ending with –ed tend to be tagged VBN.

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.