BITS Pilani
Pilani Campus

Natural Language Processing DSECL ZG565

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in

innovate achieve lead

BITS Pilani
Pilani Campus

Session 5-Part-of-Speech Tagging
Date – 23rd December 2023

These slides are prepared by the instructor, with grateful acknowledgement of Jurafsky and Martin and many others who made their course materials freely available online.

Session Content

- (Mostly) English Word Classes
- The Penn Treebank Part-of-Speech Tag set
- Part-of-Speech Tagging
- Markov Chains
- Hidden Markov Model
- HMM Part-of-Speech Tagging
- Part-of-Speech Tagging for Morphological Rich Languages

Recap: Ambiguities in language

I. Structural Ambiguities

- Namrata thinks she understands me.
- She thinks Namrata understands me.
- Visiting relatives can be nuisance. (two meanings)

II. Grammatical Ambiguities

- I (feminine or masculine) go.
- Can- Noun = container, Can – Modal(auxiliary verb), Can-verb = to can means to pack etc

III. Lexical Ambiguities:

Polysemy Ex: "understand" (I get it)

- Homonymy Ex: Bank= river, financial bank

Don’t worry! There is no problem with your eyes or computer.

Let’s try

🙜/DT 🙝⑥🙞/NN ⓪⑩/VBZ 🙚🙘🙜⑩⓪⑤🙞/VBG 🙜/DT 🙚🙜⓿/NN 🗐/.

🙜/DT 🙟⑥❹/NN ⓪⑩/VBZ ⑨❶⑤⑤⓪⑤🙞/VBG 🗐/.

🙜/DT 🙛⑥❺/NN ⓪⑩/VBZ ⑩⓪⑤🙞⓪⑤🙞/VBG 🗐/.
🙜/DT 🙘🙜⑦⑦❺/JJ 🙛⓪⑨🙝/NN

What is the POS tag sequence of the following sentence?

🙜 🙘🙜⑦⑦❺ 🙚🙜⓿ ❸🙜⑩ ⑩⓪⑤🙞⓪⑤🙞 🗐

Let’s try

- ♋/DT ♌⑥🙜/NN ⓪⑩/VBZ ♍♑♋⑩⓪⑤🙜/VBG ♋/DT ♍♋⓿/NN 🗐/.
  a/DT dog/NN is/VBZ chasing/VBG a/DT cat/NN ./.
- ♋/DT ♎⑥❹/NN ⓪⑩/VBZ ⑨❶⑤⑤⓪⑤🙜/VBG 🗐/.
  a/DT fox/NN is/VBZ running/VBG ./.
- ♋/DT ♏⑥❺/NN ⓪⑩/VBZ ⑩⓪⑤🙜⓪⑤🙜/VBG 🗐/.
  a/DT boy/NN is/VBZ singing/VBG ./.
- ♋/DT ♑♋⑦⑦❺/JJ ♏⓪⑨♌/NN
  a/DT happy/JJ bird/NN
- ♋ ♑♋⑦⑦❺ ♍♋⓿ ❸♋⑩ ⑩⓪⑤🙜⓪⑤🙜 🗐
  a happy cat was singing .

POS Tagging

- The process of assigning a part-of-speech or lexical class marker to each word in a sentence (and all sentences in a collection).

Input: **`the lead paint is unsafe`**

Output: **`the/Det lead/N paint/N is/V unsafe/Adj`**

Why is POS Tagging Useful?

First step of a vast number of practical tasks

- Helps in stemming/lemmatization
- Parsing
  - Need to know if a word is an N or V before you can parse
  - Parsers can build trees directly on the POS tags instead of maintaining a lexicon
- Named Entity Recognition and Information Extraction
  - Finding names, relations, etc.
- Machine Translation
- Selecting words of specific Parts of Speech (e.g. nouns) in pre-processing documents (for IR etc.)

Parts of Speech

- 8 (ish) 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.

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.

WORD	tag
the	**DET**
koala	**N**
put	**V**
the	**DET**
keys	**N**
on	**P**
the	**DET**
table	**N**

Open and Closed Classes

- Closed class: a small 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 these 4, but not all!

Open Class Words

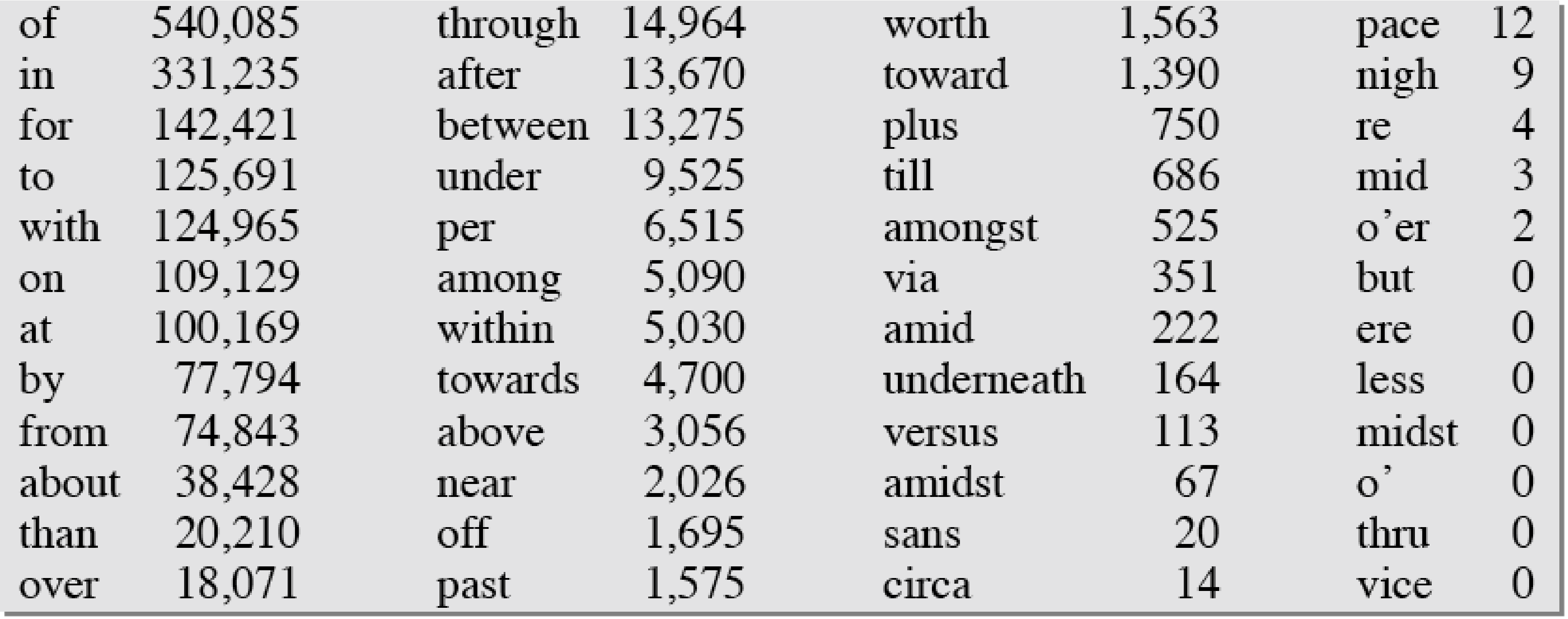
- Nouns
  - Proper nouns (Boulder, Granby, Eli Manning)
    - English capitalizes these.
  - Common nouns (the rest).
  - Count nouns and mass nouns
    - Count: have plurals, get counted: goat/goats, one goat, two goats
    - Mass: don't get counted (snow, salt, communism) (*two snows)
- Adverbs: tend 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:

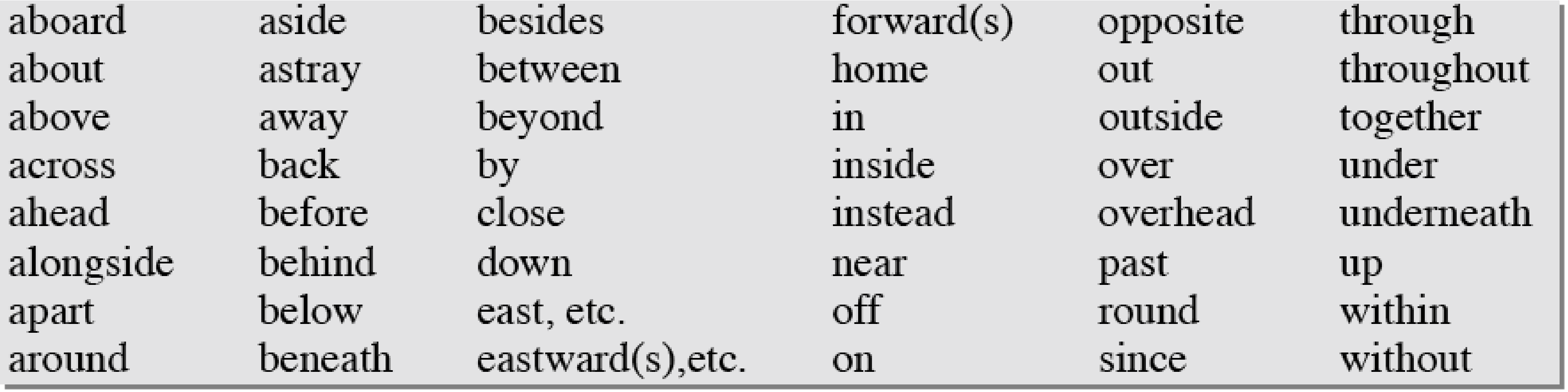
- prepositions: *on, under, over, …*
- particles: *up, down, on, off, …*
- determiners: *a, an, the, …*
- pronouns: *she, who, I, ..*
- conjunctions: *and, but, or, …*
- auxiliary verbs: *can, may should, …*
- numerals: *one, two, three, third, …*

Prepositions from CELEX



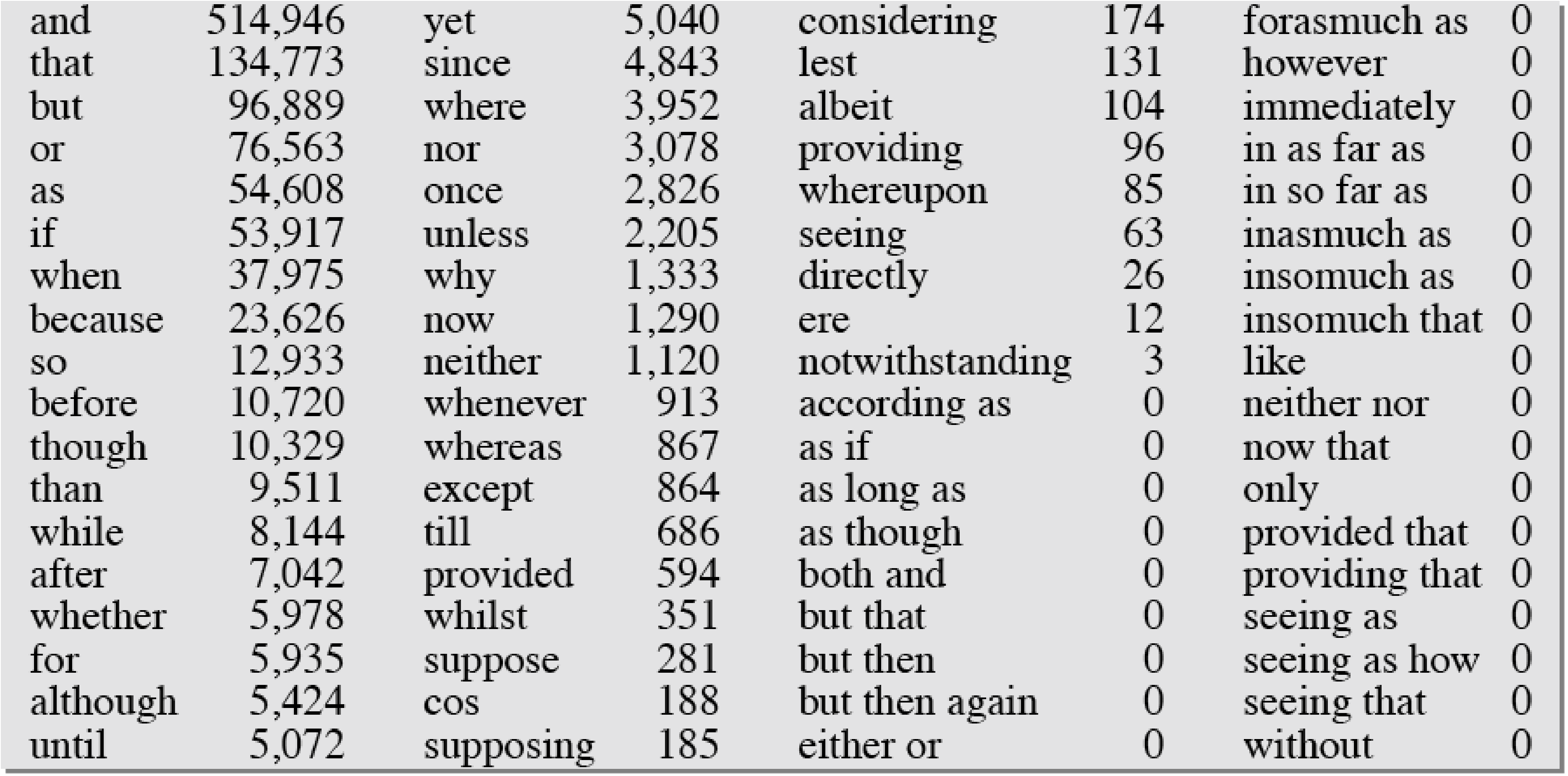
of	540,085	through	14,964	worth	1,563	pace	12
in	331,235	after	13,670	toward	1,390	nigh	9
for	142,421	between	13,275	plus	750	re	4
to	125,691	under	9,525	till	686	mid	3
with	124,965	per	6,515	amongst	525	o’er	2
on	109,129	among	5,090	via	351	but	0
at	100,169	within	5,030	amid	222	ere	0
by	77,794	towards	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 Particles



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

Conjunctions

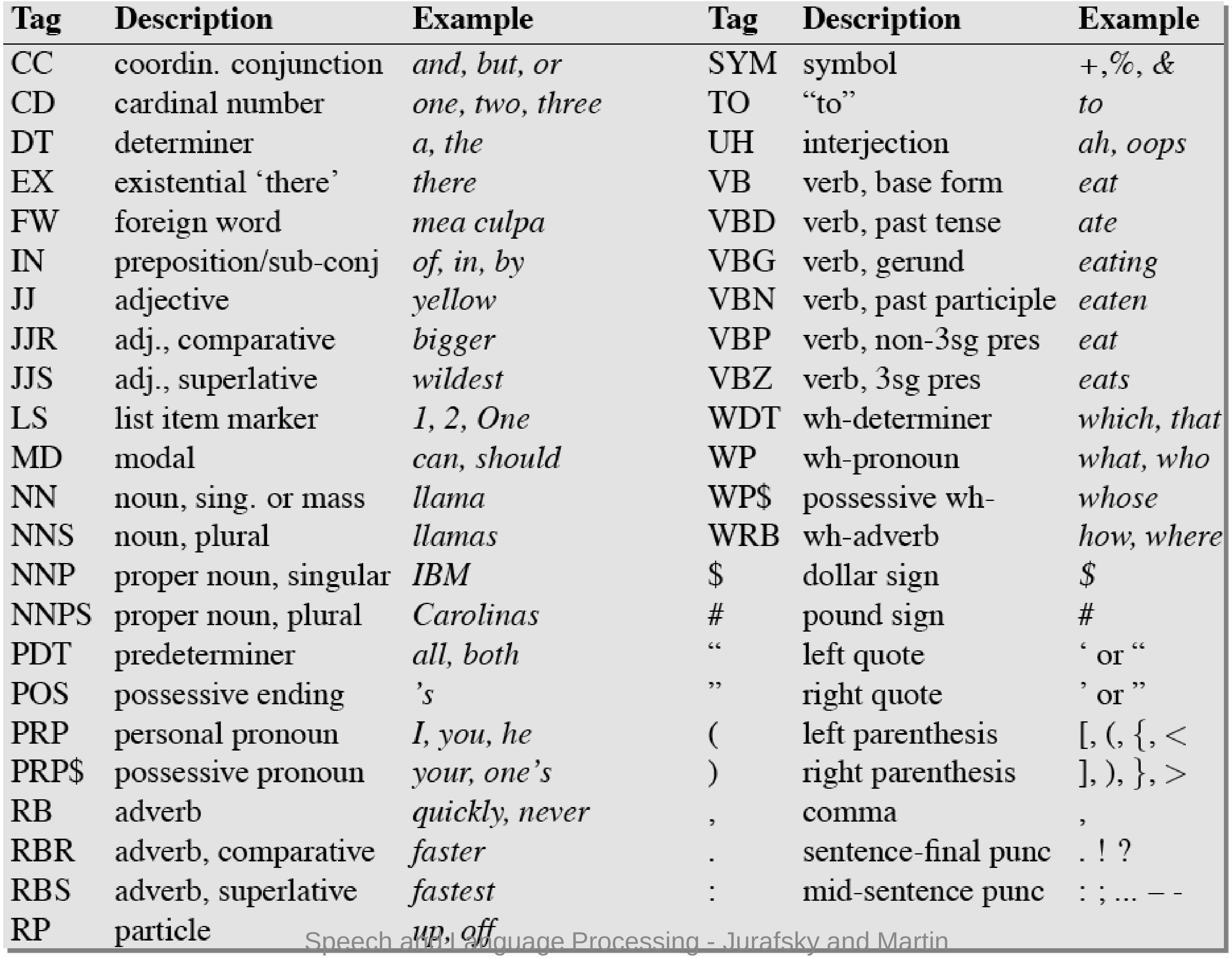


and	514,946	yet	5,040	considering	174	forasmuch as	0
that	134,773	since	4,843	lest	131	however	0
but	96,889	where	3,952	albeit	104	immediately	0
or	76,563	nor	3,078	providing	96	in as far as	0
as	54,608	once	2,826	whereupon	85	in so far as	0
if	53,917	unless	2,205	seeing	63	inasmuch as	0
when	37,975	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

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
  - N, V, Adj, Adv.
- More commonly used set is finer grained, the “Penn TreeBank tagset”, 45 tags
  - PRP$, WRB, WP$, VBG
- Even more fine-grained tagsets exist

Penn TreeBank POS Tagset



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	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	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*			

Using the Penn Tagset

- The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.
- Prepositions and subordinating conjunctions marked IN (“although/IN I/PRP..”)
- Except the preposition “to” is just marked “TO”.

POS Tagging

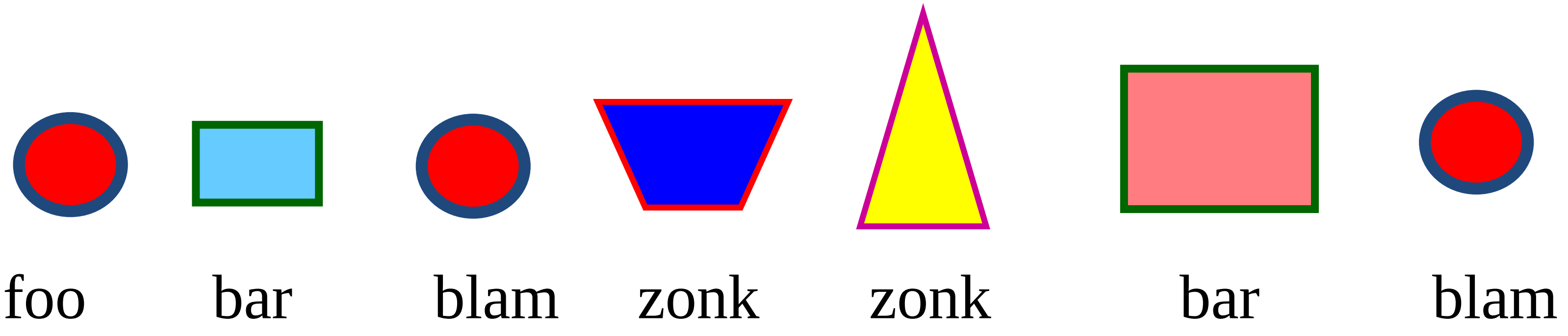
- Words often have more than one POS: *back*
  - – The ***back*** door = JJ
  - – On my ***back*** = NN
  - – Win the voters ***back*** = RB
  - – Promised to ***back*** the bill = VB
- The POS tagging problem is to determine the POS tag for a particular instance of a word.

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 $w_1 \ldots w_n$.

Sequence Labeling Problem

- Many NLP problems can viewed as sequence labeling.
- Each token in a sequence is assigned a label.
- Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors (not i.i.d).



foo bar blam zonk zonk bar blam

Information Extraction

- Identify phrases in language that refer to specific types of entities and relations in text.
- Named entity recognition is task of identifying names of people, places, organizations, etc. in text.

  people organizations places

  – Michael Dell is the CEO of Dell Computer Corporation and lives in Austin Texas.
- Extract pieces of information relevant to a specific application, e.g. used car ads:

  make model year mileage price

  – For sale, 2002 Toyota Prius, 20,000 mi, $15K or best offer. Available starting July 30, 2006.

Semantic Role Labeling

- For each clause, determine the semantic role played by each noun phrase that is an argument to the verb.
  agent patient source destination instrument
  - John drove Mary from Austin to Dallas in his Toyota Prius.
  - The hammer broke the window.
- Also referred to a "case role analysis," "thematic analysis," and "shallow semantic parsing"

Bioinformatics

- Sequence labeling also valuable in labeling genetic sequences in genome analysis.

  extron intron

  – AGCTAACGTTCGATACGGATTACAGCCT

Problems with Sequence Labeling as Classification

- Not easy to integrate information from category of tokens on both sides.
- Difficult to propagate uncertainty between decisions and “collectively” determine the most likely joint assignment of categories to all of the tokens in a sequence.

Probabilistic Sequence Models

- Probabilistic sequence models allow integrating uncertainty over multiple, interdependent classifications and collectively determine the most likely global assignment.
- standard model
  - Hidden Markov Model (HMM)

Hidden Markov Models

- It is a **sequence model**.
- Assigns a label or class to each unit in a sequence, thus mapping a **sequence of observations** to a **sequence of labels**.
- Probabilistic sequence model: given a sequence of units (e.g. words, letters, morphemes, sentences), compute a probability distribution over possible sequences of labels and choose the best label sequence.
- This is a kind of *generative* model.

Markov Model / Markov Chain

- A finite state machine with probabilistic state transitions.
- Makes Markov assumption that next state only depends on the current state and independent of previous history.

Hidden Markov Models

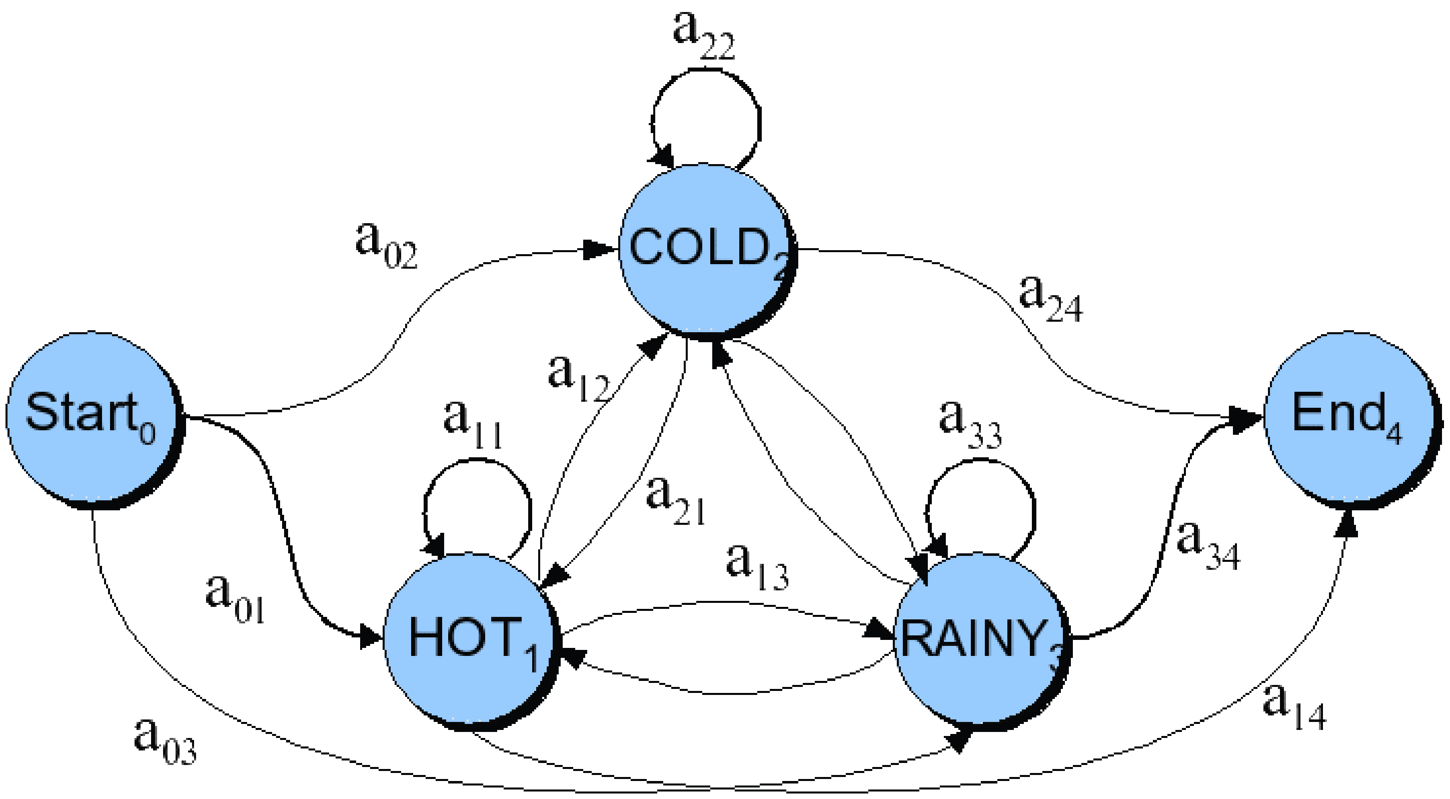
- *Generative* model.
  - There is a hidden underlying generator of observable events
  - The hidden generator can be modeled as a network of states and transitions
  - We want to infer the underlying state sequence given the observed event sequence

Markov Chain: "First-order observable Markov Model"

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

$$P(q_i \mid q_1 \ldots q_{i-1}) = P(q_i \mid q_{i-1})$$

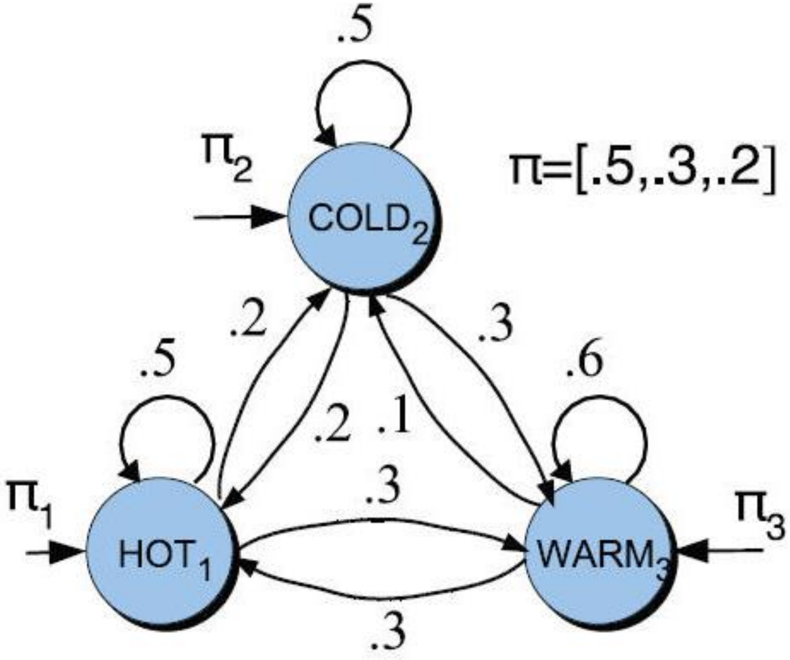
Markov Chain for Weather



Markov Chain for Weather

- What is the probability of 4 consecutive warm days?

- Sequence is warm-warm-warm-warm
- And state sequence is 3-3-3-3
- P(3,3,3,3) =
  - $\pi_3 a_{33} a_{33} a_{33} = 0.2 \text{ x } (0.6)^3 = 0.0432$



HMM to predict the tags

- Two types of information are useful
  - Relations between words and tags
  - Relations between tags and tags
    - DT NN, DT JJ NN…

P(DT JJ NN | a smart dog)
= P(DD JJ NN a smart dog) / P (a smart dog)
= P(DD JJ NN) P(a smart dog | DD JJ NN)

Hidden Markov Models (Formal)

- States $Q = q_1, q_2 \ldots q_{N;}$
- Observations $O = o_1, o_2 \ldots o_{N;}$
  - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \ldots v_V\}$
- Transition probabilities
  - Transition probability matrix $A = \{a_{ij}\}$

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \pounds i, j \pounds N$$

- Observation likelihoods
  - Output probability matrix $B=\{b_i(k)\}$

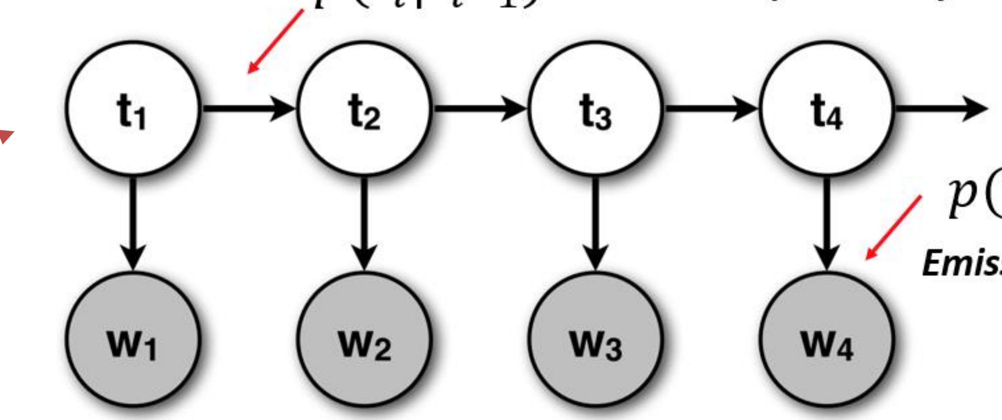
$$b_i(k) = P(X_t = o_k \mid q_t = i)$$

- Special initial probability vector π

$$p_i = P(q_1 = i) \quad 1 \pounds i \pounds N$$

Prediction in generative model

- Inference: What is the most likely sequence of tags for the given sequence of words **w**



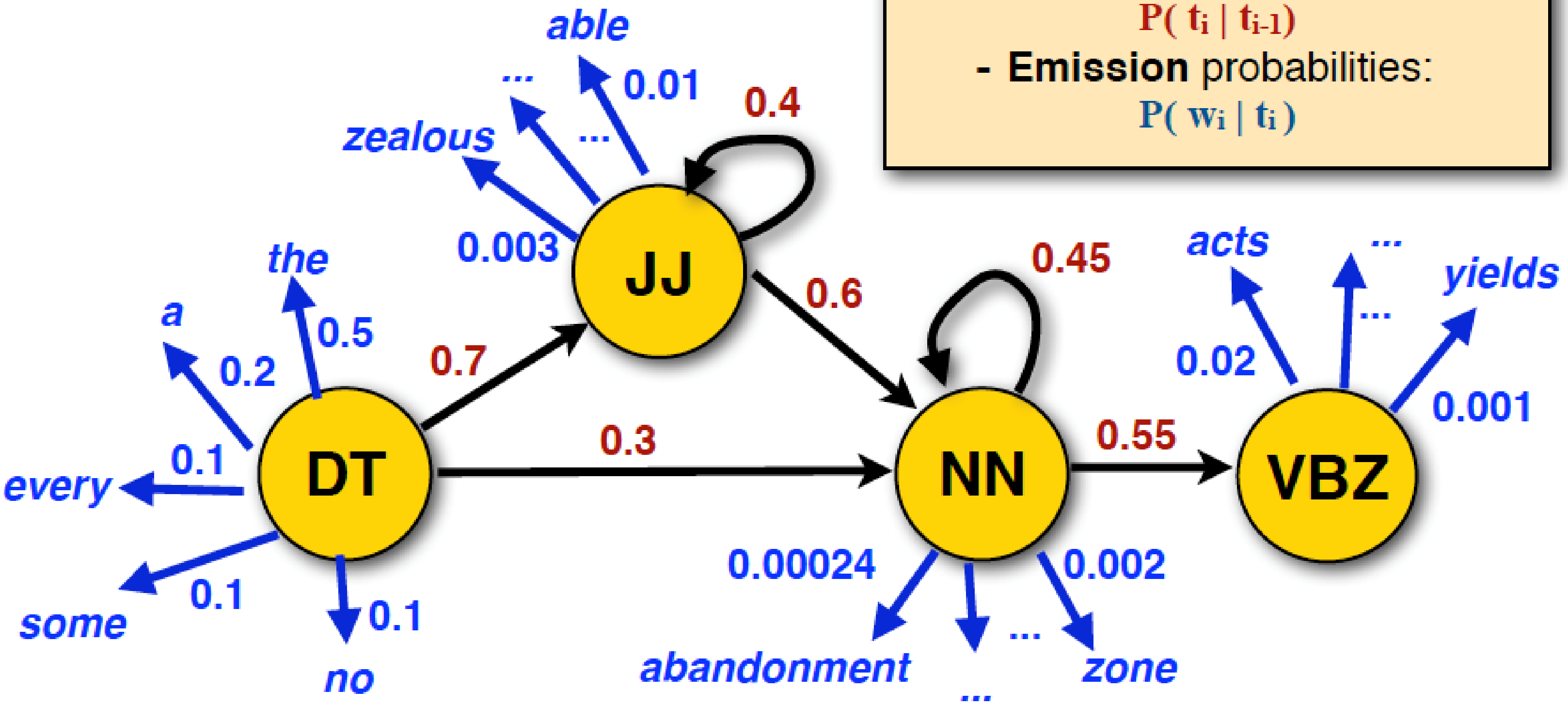
$p(t_i|t_{i-1})$ ***Transition probability***

initial probability
$p(t_1)$

$p(w_i|t_i)$
Emission probability

- What are the latent states that most likely generate the sequence of word **w**

HMMs as probabilistic FSA



An HMM defines

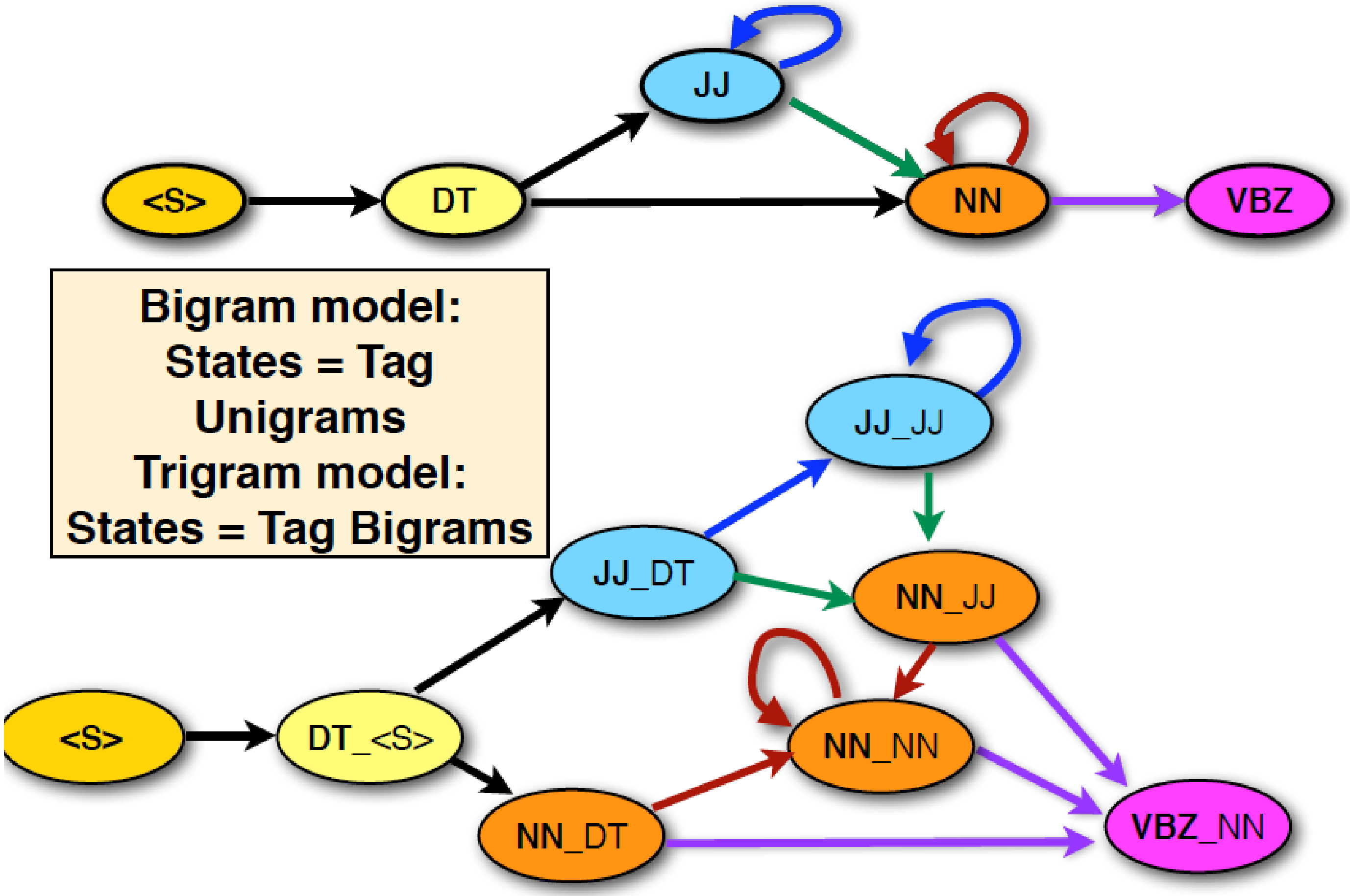
- **Transition** probabilities: $P(t_i \mid t_{i-1})$
- **Emission** probabilities: $P(w_i \mid t_i)$

Julia Hockenmaier: Intro to NLP

How to build a second-order HMM?

- Second-order HMM
  - Current state only depends on previous 2 states
- Example
  - Trigram model over POS tags
  - $P(\boldsymbol{t}) = \Pi_{i=1}^{n} P(t_i \mid t_{i-1}, t_{i-2})$
  - $P(\boldsymbol{w}, \boldsymbol{t}) = \Pi_{i=1}^{n} P(t_i \mid t_{i-1}, t_{i-2}) P(w_i \mid t_i)$

Probabilistic FSA for second-order HMM



Bigram model:
States = Tag Unigrams
Trigram model:
States = Tag Bigrams

Julia Hockenmaier: Intro to NLP

Hidden Markov Models (POS Tagging)

- States $T = t_1, t_2 \ldots t_{N;}$
- Observations $W = w_1, w_2 \ldots w_{N;}$
  - Each observation is a symbol from a vocabulary V = $\{v_1, v_2, \ldots v_V\}$
- Transition probabilities
  - Transition probability matrix $A = \{a_{ij}\}$

$$a_{ij} = P(\, t_i = j \mid t_{i-1} = i \,)\;\; 1 \le i, j \le N$$

- Observation likelihoods
  - Output probability matrix $B = \{b_i(k)\}$

$$b_i(k) = P(\, w_i = v_k \mid t_i = i \,)$$

- Special initial probability vector π

$$\pi_i = P(t_1 = i)\;\; 1 \le i \le N$$

Statistical POS Tagging

- 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.

$$\hat{t}_1^n = \operatorname*{argmax}_{t_1^n} P(t_1^n | w_1^n)$$

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

Statistical POS Tagging

- This equation should give us the best tag sequence

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}}\, P(t_1^n | w_1^n)$$

- But how to make it operational? How to compute this value?
- Intuition of Bayesian inference:
  - – Use Bayes rule to transform this equation into a set of probabilities that are easier to compute (and give the right answer)

Using Bayes Rule

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}}\, P(t_1^n|w_1^n)$$

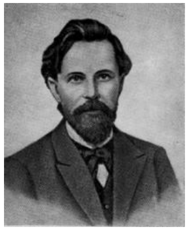
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}}\, \frac{P(w_1^n|t_1^n)P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}}\, P(w_1^n|t_1^n)P(t_1^n)$$

Likelihood and Prior

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \overbrace{P(w_1^n|t_1^n)}^{\text{likelihood}} \; \overbrace{P(t_1^n)}^{\text{prior}}$$

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$



$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

$$\boxed{\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-1})}$$

Statistical POS tagging

- What is the most likely sequence of tags for the given sequence of words w

$$
\begin{aligned}
\operatorname*{argmax}_{\mathbf{t}} P(\mathbf{t}|\mathbf{w}) &= \operatorname*{argmax}_{\mathbf{t}} \frac{P(\mathbf{t},\mathbf{w})}{P(\mathbf{w})} \\
&= \operatorname*{argmax}_{\mathbf{t}} P(\mathbf{t},\mathbf{w}) \\
&= \operatorname*{argmax}_{\mathbf{t}} P(\mathbf{t})P(\mathbf{w}|\mathbf{t})
\end{aligned}
$$

P(DT JJ NN | a smart dog)
= P(DD JJ NN a smart dog) / P (a smart dog)
= P(DD JJ NN) P(a smart dog | DD JJ NN)

Transition Probability

- Joint probability $P(\boldsymbol{t}, \boldsymbol{w}) = P(\boldsymbol{t})P(\boldsymbol{w}|\boldsymbol{t})$
- $P(\boldsymbol{t}) = P(t_1, t_2, \ldots t_n)$

$$= P(t_1)P(t_2 \mid t_1)P(t_3 \mid t_2, t_1) \ldots P(t_n \mid t_1 \ldots t_{n-1})$$

$$\sim \mathrm{P}(\mathrm{t}_1)\mathrm{P}(\mathrm{t}_2 \mid t_1)P(t_3 \mid t_2) \ldots P(t_n \mid t_{n-1})$$

$$= \Pi_{i=1}^{n} P(t_i \mid t_{i-1})$$

Markov assumption

- Bigram model over POS tags!
  (similarly, we can define a n-gram model over POS tags, usually we called high-order HMM)

Emission Probability

- Joint probability $P(\boldsymbol{t}, \boldsymbol{w}) = P(\boldsymbol{t})P(\boldsymbol{w}|\boldsymbol{t})$
- Assume words only depend on their POS-tag
- $P(\boldsymbol{w}|\boldsymbol{t}) \sim P(w_1 \mid t_1)P(w_2 \mid t_2) \ldots P(w_n \mid t_n)$

$$= \Pi_{i=1}^{n} P(w_i|t_i)$$

Independent assumption

i.e., P(a smart dog | DD JJ NN)

= P(a | DD) P(smart | JJ) P(dog | NN)

Put them together

- Joint probability $P(\boldsymbol{t}, \boldsymbol{w}) = P(\boldsymbol{t})P(\boldsymbol{w}|\boldsymbol{t})$
- $P(\boldsymbol{t}, \boldsymbol{w})$

$$= \mathrm{P}(\mathrm{t}_1)\mathrm{P}(\,\mathrm{t}_2 \mid t_1\,)P(\,t_3 \mid t_2\,) \ldots P(\,t_n \mid t_{n-1}\,)$$

$$P(\,w_1 \mid t_1\,)P(\,w_2 \mid t_2\,) \ldots P(w_n \mid t_n)$$

$$= \Pi_{i=1}^{n} P(w_i|t_i)P(t_i \mid t_{i-1})$$

e.g., P(a smart dog , DD JJ NN)

= P(a | DD) P(smart | JJ) P(dog | NN)

P(DD | start) P(JJ | DD) P(NN | JJ)

Two Kinds of Probabilities

1. **State transition probabilities** -- $p(t_i|t_{i-1})$
   – State-to-state transition probabilities

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

2. **Observation/Emission probabilities** -- $p(w_i|t_i)$
   – Probabilities of observing various values at a given state

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

Two Kinds of Probabilities

1. **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(NN|DT) and P(JJ|DT) to be high
   - Compute P(NN|DT) by counting in a labeled corpus:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

$$P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = .49$$

Two Kinds of Probabilities

2. **Word likelihood/emission** probabilities $p(w_i|t_i)$
   - VBZ (3sg Pres Verb) likely to be "is"
   - Compute P(is|VBZ) by counting in a labeled corpus:

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$

Sample Transition Probabilities

Bigram Probabilities

Bigram(Ti, Tj)	Count(i, i + 1)	Prob(Tj|Ti)
φ,ART	213	.71 (213/300)
φ,N	87	.29 (87/300)
φ,V	10	.03 (10/300)
ART,N	633	1
N,V	358	.32
N,N	108	.10
N,P	366	.33
V,N	134	.37
V,P	150	.42
V,ART	194	.54
P,ART	226	.62
P,N	140	.38
V,V	30	.08

Tag Frequencies

Φ	ART	N	V	P
310	633	1102	358	366

Φ – Start Symbol

Sample Lexical Generation Emission Probabilities

• P(an | ART)	.36
• P(an | N)	.001
• P(flies | N)	.025
• P(flies | V)	.076
• P(time | N)	.063
• P(time | V)	.012
• P(arrow | N)	.076
• P(like | N)	.012
• P(like | V)	.10

Table representation

Transition Matrix A

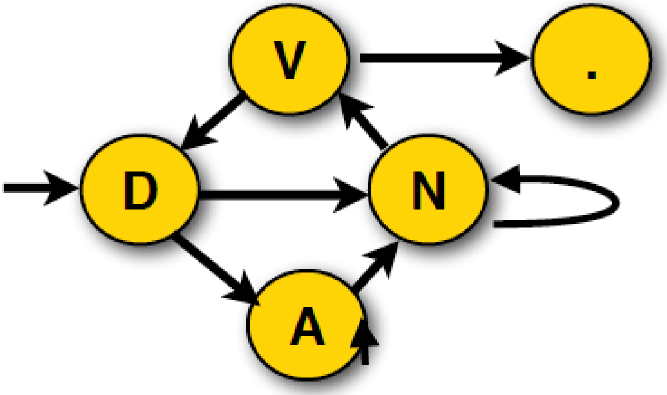
	D	N	V	A	.
D		0.8		0.2	
N		0.7	0.3		
V	0.6				0.4
A		0.8		0.2	
.					

Emission Matrix B

	the	*man*	*ball*	*throws*	*sees*	*red*	*blue*	.
D	1.0							
N		0.7	0.3					
V				0.6	0.4			
A						0.8	0.2	
.								1

Initial state vector π

	D	N	V	A	.
π	1.0				

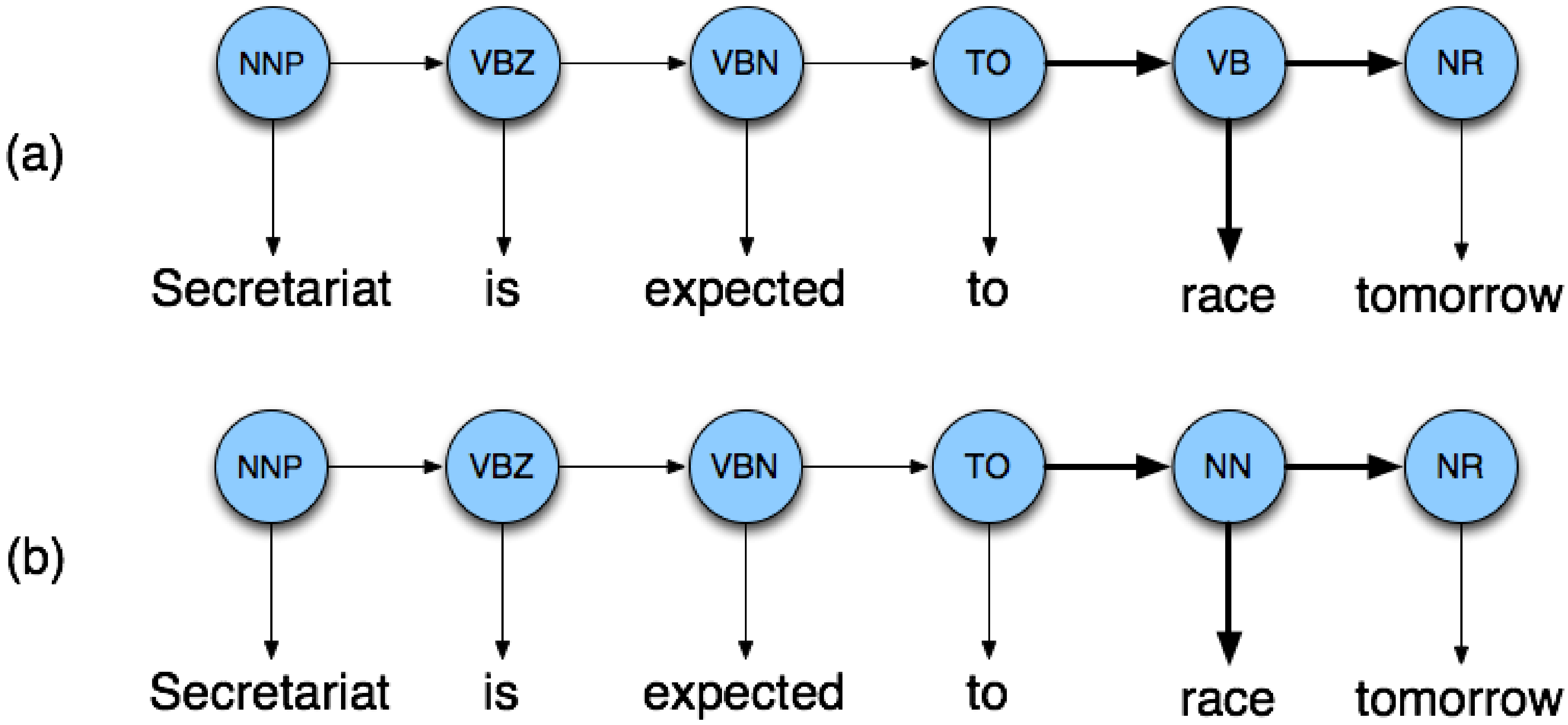


Let $\lambda = \{A, B, \pi\}$ represents all parameters

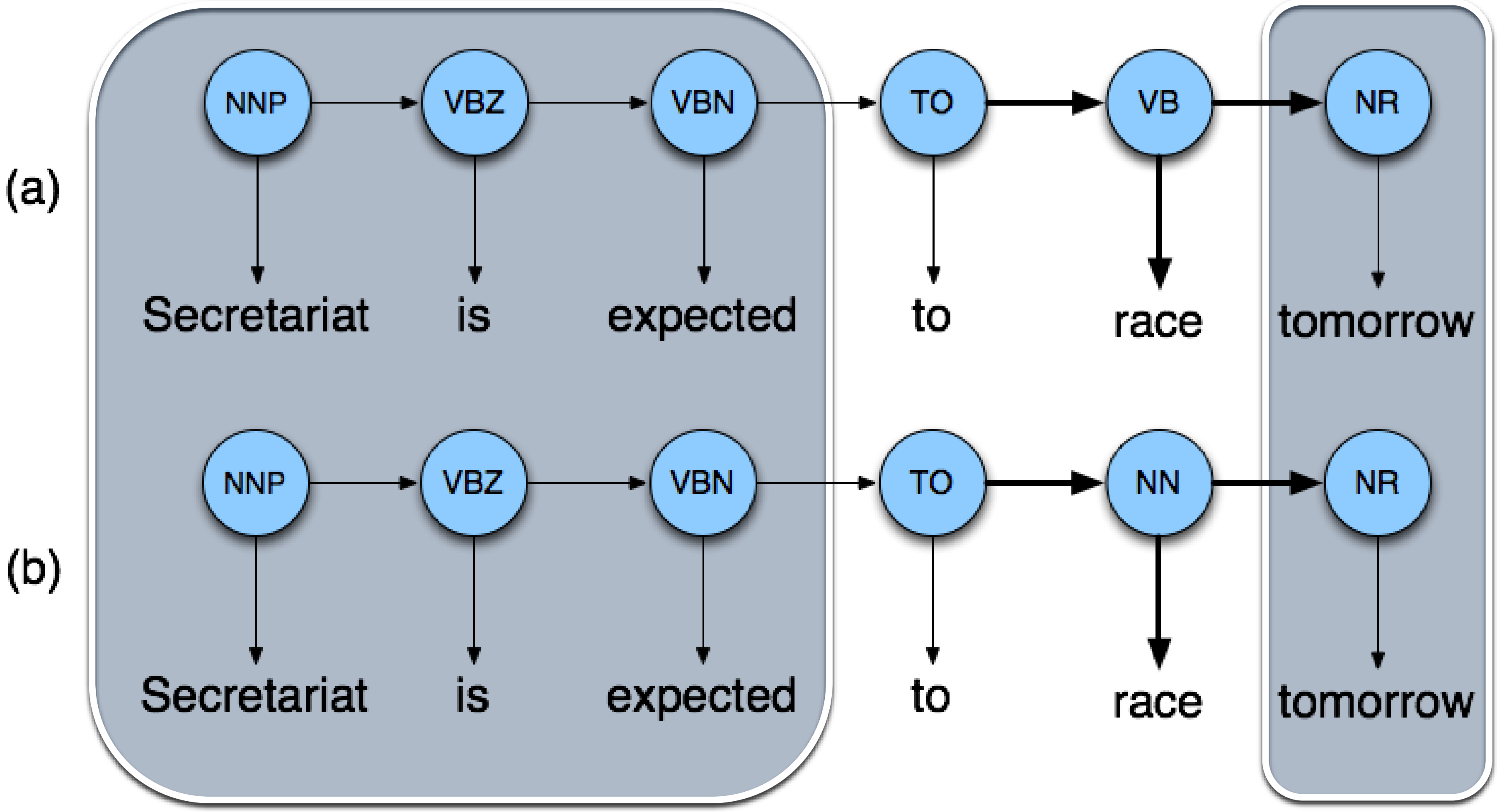
Example: The Verb "race"

- 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?

Disambiguating “race”

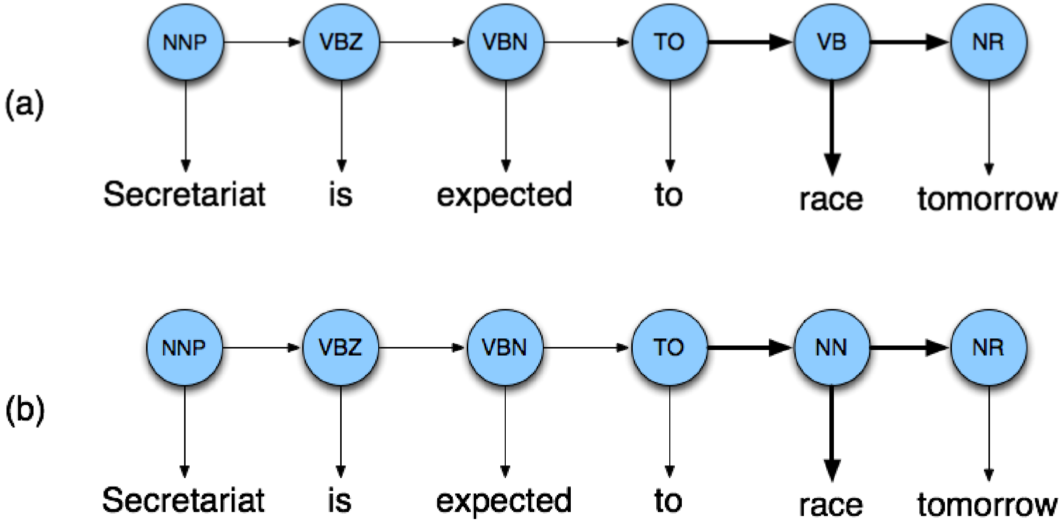


Disambiguating “race”



Example : NLTK POS tags

- P(NN|TO) = .00047
- P(VB|TO) = .83
- P(race|NN) = .00057
- P(race|VB) = .00012
- P(NR|VB) = .0027
- P(NR|NN) = .0012



NR-Adverbial Noun (tomorrow), TO-proposition

- P(VB|TO)P(NR|VB)P(race|VB) = .00000027
- P(NN|TO)P(NR|NN)P(race|NN)=.00000000032
- So we (correctly) choose the verb tag for “race”

https://www.guru99.com/pos-tagging-chunking-nltk.html

POS Tagging Demo

There are different open source POS taggers available

- Stanford PoS Tagger python bind- Java based, but can be used in python but difficult to install
- Flair - POS tagger available for python.
- NLTK implementation to be very precise, around 97% and its quite fast.
- SPACY

Part-of-Speech Tagging for Morphological Rich Languages

- Augmentations to tagging algorithms become necessary when dealing with languages with rich morphology like Czech, Hungarian and Indian languages
- Highly inflectional languages also have much more information than English coded in word morphology, like case (nominative, accusative etc) or gender (masculine, feminine).Ex: I am speaking in English doesn’t tell about gender but in hindi/Marathi you can come to know gender depending on Bolti or Bolta(inflections)
- This information is important for tasks like parsing and coreference resolution, part-of-speech taggers for morphologically rich languages need to label words with case and gender information.

Part-of-Speech Tagging for Morphological Rich Languages

- Using a morphological parse sequence like Noun+A3sg+Pnon+Gen as the part-of-speech tag greatly increases the number of parts of speech, and so tagsets can be 4 to 10 times larger than the 50–100 tags we have seen for English.
- With such large tagsets, each word needs to be morphologically analyzed to generate the list of possible morphological tag sequences (part-of-speech tags) for the word.
- The role of the tagger is then to disambiguate among these tags.
- This method also helps with unknown words since morphological parsers can accept unknown stems and still segment the affixes properly.

Part-of-Speech Tagging for Morphological Rich Languages

- For non-word-space languages like Chinese, word segmentation is either applied before tagging or done jointly.
- Although Chinese words are on average very short (around 2.4 characters per unknown word compared with 7.7 for English) the problem of unknown words is still large.
- While English unknown words tend to be proper nouns in Chinese the majority of unknown words are common nouns and verbs because of extensive compounding.
- Tagging models for Chinese use similar unknown word features to English, including character prefix and suffix features

References

NLTK :: nltk.tag package

NLTK POS Tagging – Python Examples - Python Examples

https://www.coursera.org/lecture/probabilistic-models-in-nlp/part-of-speech-tagging-VbnrA

https://www.coursera.org/lecture/text-mining-analytics/3-5-how-to-do-ner-and-pos-tagging-yQEYw

https://github.com/rajesh-iiith/POS-Tagging-and-CYK-Parsing-for-Indian-Languages/tree/master/data

https://marcossilva.github.io/en/2020/09/07/coursera-nlp-module-2-week-2.html

https://www.intechopen.com/online-first/88505

Dr. Chetana Gavankar has over 25 years of Teaching, Research and Industry experience. She has published papers in peer reviewed international conferences and journals. She is also reviewer for multiple conferences and journals. She has worked on different projects with multiple industries and received awards for her research work . Her areas of research interests include Natural Language Processing, Information Retrieval, Web Mining and Semantic Web, Ontology, Big Data Analytics, Machine learning, Deep learning and Artificial Intelligence.

Thank you!!