Part-of-Speech Tagging and the Viterbi Algorithm

Dr. Demetrios Glinos
University of Central Florida

CAP6640 – Computer Understanding of Natural Language

Today

Parts of speech (POS)

Tagsets

POS Tagging

Rule-based tagging

Hidden Markov Models

The Viterbi algorithm

Parts of Speech

- Parts of speech (POS)
  - the roles that words play in sentence structure
  - also called lexical categories, word classes, morphological classes, lexical tags, etc.
- 8+ traditional syntactic categories
  - noun, verb, adjective, adverb, preposition, article, pronoun, conjunction, interjection
- Not a settled collection
  - much debate among linguists on the number, nature, and universality of the roles
  - we will ignore this debate

POS Examples

Abbreviation	POS	Examples
N	noun	book, bandwidth, score
V	verb	study, debate, eat
ADJ	adjective	red, big, rediculous, complex
ADV	adverb	quickly, unfortunately, well
P	preposition	of, by, to, from, in, among, for
PRO	pronoun	I, me, mine, his, her, its
DET	determiner	a, an, the, that, those

POS Tagging

- The task of assigning a part-of-speech tag to each word in a sentence

WORD	POS tag
the	DET
man	N
put	V
his	PRO
phone	N
on	P
the	DET
table	N

POS Tagging is Useful

- POS is the first step in a many practical NLP tasks
- Speech synthesis
  - How to pronounce "lead" ?
  - OBject obJECT
  - DIScount disCOUNT
  - CONtent conTENT
  - OVERflow overFLOW
- Parsing
  - needs to know POS to extract higher-order structures
- Information extraction
  - POS is needed to extract named entities and relations among them
- Machine translation
  - need to understand what is being stated in order to translate it

Open and Closed Classes

- closed class
  - small, fixed class
  - usually function words (short common words that play a role in grammar)
  - examples
    - prepositions: of, in, by, for, about, ...
    - auxiliaries: may, can, will, had, been, ...
    - pronouns: I, we, you, he, she, it, they, ...

- open class
  - new instances are constantly being created
  - English has 4 open classes: nouns, verbs, adjectives, adverbs
  - Many languages have these 4 class
  - But some do not, e.g., Korean does not have adjectives

Open Class Words (1)

- Nouns
  - persons, places, things
  - proper nouns (proper names)
    - capitalized in English: Orlando, Eli Manning
  - common nouns (the rest)
    - count nouns: have plurals, get counted (bit/bits, one bit, two bits)
    - mass nouns: don't get counted (snow, salt, mercantilism)

- Verbs
  - action verbs (basic forms)
    - have morphological affixes for person, number, and tense (eat, eats, ate)
  - auxiliary verbs: modify a verb, but don't have semantic content
    - e.g., I ***do not like*** oysters; she ***has gone*** to the store
  - copular verbs: equate the terms that they connect
    - e.g., she ***is*** a doctor, Peter ***appears*** content, that ***seems*** reasonable

Open Class Words (2)

- Adverbs
  - verb modifiers
    - ***"Interestingly***, Peter walked home ***extremely slowly yesterday."***
    - directional/locative: here, home, downhill
    - degree: very, highly, extremely, somewhat
    - manner: slowly, delicately, quietly
- Adjectives
  - noun modifiers
    - "Marla drives a ***late-model silver sports*** car."
    - properties or attributes: types, qualities, etc
    - e.g., color, age, value, size

Closed Class Words

- Examples
  - prepositions: in, on, over, under, about, for, ...
  - pronouns: I, she, my, their, who, whose, ...
  - conjunctions: and, but, or, nor, ...
  - determiners: a, an, the, ...
  - particles: up, down, on, off, ...
  - auxiliary verbs: can, may, should, would, ...
  - numerals: one, two, three, first, second, third, ...

English Prepositions from CELEX Corpus

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

source: J&M, Fig. 5.1

English 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

source: J&M, Fig. 5.2

English 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

source: J&M, Fig. 5.3

Today

Parts of speech (POS)

Tagsets

POS Tagging

Rule-based tagging

Hidden Markov Models

The Viterbi algorithm

POS Tagging: Choosing a Tagset

- Many different tagsets
  - focus on different aspects of the language
- Choosing a tagset
  - can we find enough labeled data for training/test?
  - will it support the downstream processing we are interested in?
- Can choose very coarse tagset
  - N, V, Adv, Adj, Prep
- More commonly used
  - Brown corpus (Francis & Kucera, 1982) tagset: 87 tags
  - Penn Treebank tagset: 45 tags
- Even more fine-grained tagsets exist

Penn Treebank 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*			

source: J&M, Fig. 5.6

Using the Penn Treebank Tagset

- Input
  - "The grand jury commented on a number of other topics ."
- Tagged
  - The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.
- prepositions and subordinating conjuctions are tagged IN
  - "although/IN I/PRP ..."
- except the preposition /complementizer "to" is always tagged TO
  - "to/TO study/VB ..."
  - "to/TO the/DT top/NN ..."

Today

Parts of speech (POS)

Tagsets

POS Tagging

Rule-based tagging

Hidden Markov Models

The Viterbi algorithm

The POS Tagging Problem

- Words often have more than one POS
  - as adjective: "the ***back*** door"
  - as noun: "on my ***back***"
  - as adverb: "win the voters ***back***"
  - as verb: "promise to ***back*** the bill"

- The POS tagging problem is to determine the correct POS tag for a particular instance of a word

POS Tagging is Difficult Even for Humans

- We/PRP must/VB walk/VB ***around/IN*** the/DT corner/NN
- Dinner/NN will/MD cost/VBZ ***around/RB*** 40/CD dollars/CD
- Peter/NNP never/RB got/VBD ***around/RP*** to/TO joining/VBG

How Difficult is POS Tagging?

- The Brown corpus (1961)
  - the first electronic corpus
  - 1 million words
  - 500 documents from 15 genres
- 11% of word types are ambiguous
  - tends to be common words like "that"
- 40% of tokens are ambiguous

Measuring Ambiguity

		87-tag Original Brown		45-tag Treebank Brown	
Unambiguous (1 tag)		**44,019**		**38,857**	
Ambiguous (2–7 tags)		**5,490**		**8844**	
Details:	2 tags	4,967		6,731	
	3 tags	411		1621	
	4 tags	91		357	
	5 tags	17		90	
	6 tags	2	(*well*, *beat*)	32	
	7 tags	2	(*still*, *down*)	6	(*well*, *set*, *round*, *open*, *fit*, *down*)
	8 tags			4	(*'s*, *half*, *back*, *a*)
	9 tags			3	(*that*, *more*, *in*)

source: J&M, Fig. 5.10

Today

Parts of speech (POS)

Tagsets

POS Tagging

Rule-based tagging

Hidden Markov Models

The Viterbi algorithm

POS Tagging Approaches

- Rule-based
  - generally use a large DB of hand-written rules to resolve tagging ambiguities
  - we will illustrate using EngCG ENGTWOL (English Two Level tagger) of (Voutilainen, 1995, 1999)
    - based on constraint grammer concepts
- Stochastic
  - use a training corpus to compute conditional probabilities of a word in its lexical context
  - we will examine
    - HMM (Hidden Markov Model) tagging using the Viterbi algorithm (this lecture)
    - MEMM (Maximum Entropy Markov Model) tagging (next lecture)

ENTWOL Rule-Based Tagging

- Start with a dictionary
- Stage 1: Assign all possible tags to each word of the test sentence
- Stage 2: Write rules by hand to selectively remove possible tags
- With enough rules, this should leave the correct tag for each word

Start With a Dictionary

- Example words in dictionary

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

- precompute for all words in dictionary
  - including ~100K words of English that have > 1 tag

Stage 1: Assign Every Possible Tag

Given the test sentence

			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

Stage 2: Apply Rules to Eliminate Tags

Rule: 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**

Today

Parts of speech (POS)

Tagsets

POS Tagging

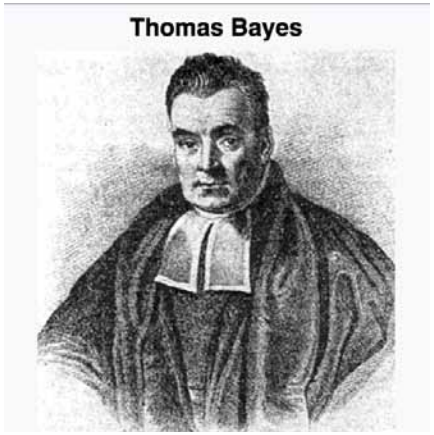
Rule-based tagging

Hidden Markov Models

The Viterbi algorithm

Hidden Markov Model Tagging

- Using HMM for tagging is an application of Bayesian inferencing
  - paradigm known since the time of Bayes (1763)
  - in NLP
    - OCR: Bledsoe (1959)
    - authoriship: Mosteller and Wallace (1964)



- also related to the "noisy channel" model that is the basis for ASR, OCR, and MT

Probabilistic View of POS Tagging

- Consider the sentence: "Secretariat is expected to race tomorrow"
- Probabilistic view:
  - The words of the sentence constitute a sequence of observations
  - The tags for those words represent hidden states
  - The probabilistic question:
    - What is the most probable sequence of tags for this sequence of observations, considering all possible sequences?

Mathematical Formulation

- Given
  - a sequence of words $\mathbf{w_1, \ldots, w_n}$
- Want
  - the tag sequence $\mathbf{t_1, \ldots, t_n}$ such that $\mathbf{P(t_1, \ldots, t_n \mid w_1, \ldots, w_n)}$ is maximal

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

  - where
    - Hat ^ means "our estimate of the best one"
    - $\operatorname{argmax}_x f(x)$ meams "the x such that f(x) is maximized"

Using Bayes Rule

Bayes Rule: $$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

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

Dropping the denominator:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \underbrace{P(\, w_1^n \mid t_1^n \,)}_{\text{likelihood}} \underbrace{P(t_1^n)}_{\text{prior}}$$

Simplifying Assumptions

Applying the assumption that the probability of word depends only on its tag, we have

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

Applying the Markov assumption

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

Therefore,

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

Computing Tag Transition Probabilities

- Tag transition probabilites are of the form $P(t_i|t_{i-1})$
  - Example: Determiners are likely to precede adjectives 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, but P(DT|JJ) to be low

- We compute P(NN|DT) by counting in a labeled corpus: $P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$

- In the 45-tag Treebank Brown corpus: $P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56{,}509}{116{,}454} = .49$

Computing Word Likelihood Probabilities

- Word likelihoos probabilites are of the form $P(w_i|t_i)$
  - Example: The likelihood that a VBZ (3sg Pres verb) is the verb "is"
- We compute by counting in a labeled corpus: $P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$
- In the 45-tag Treebank Brown corpus: $P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10{,}073}{21{,}627} = .47$

Computing the Most Likely Sequence

- Running example: "Secretariat is expected to race tomorrow"
- How do we properly tag this sentence?
- In particular, how do we distinguish the proper tag for "race"
  - Goal (using the Brown 87-tag tagset):

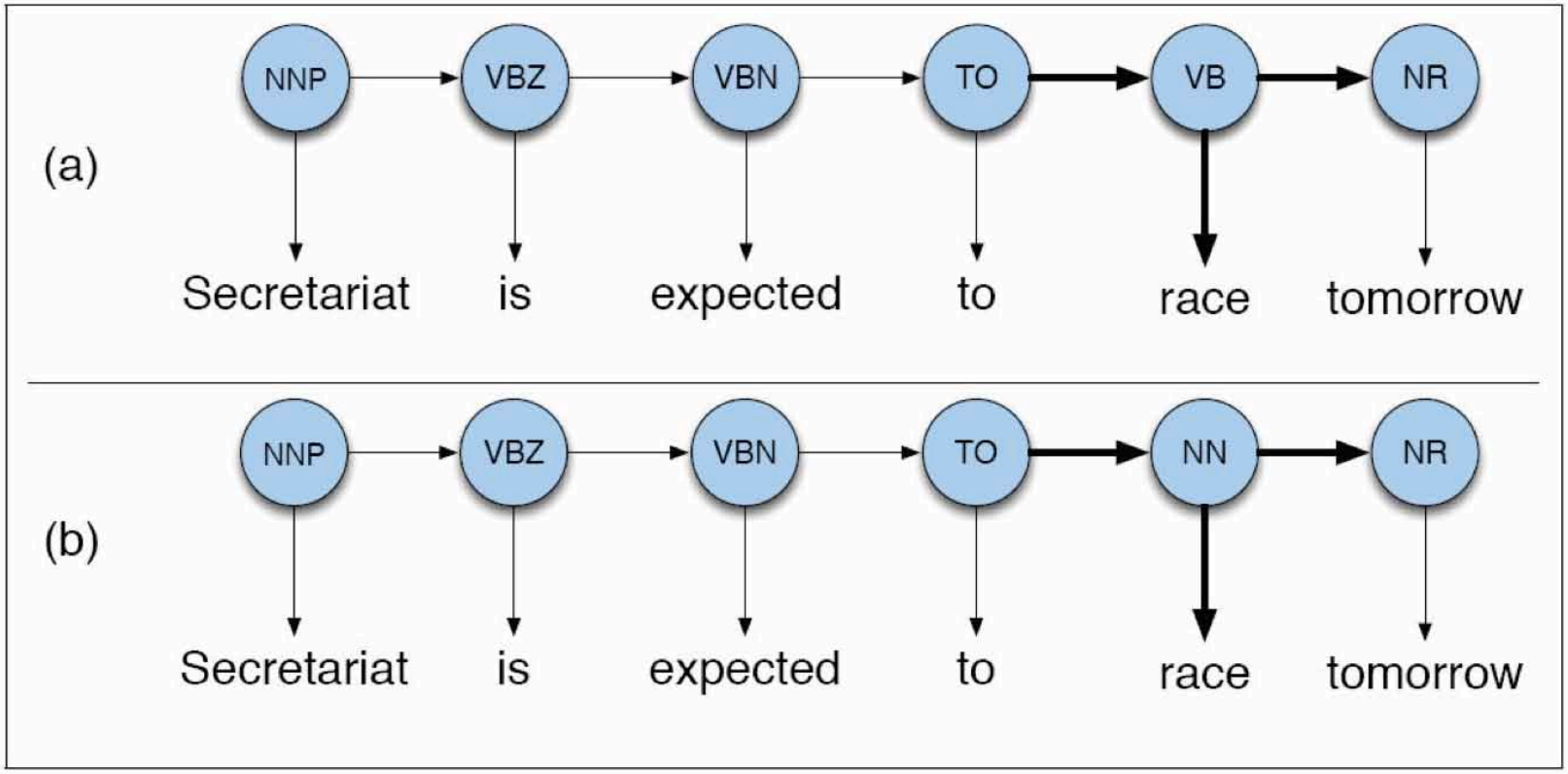
    Secretariat/NNP is/VBZ expected/VBN to/TO **race/VB** tomorrow/NR

  - Compare:

    People/NNS continue/VB to/TO inquire/VB the/AT reason/NN for/IN the/AT **race/NN** for/IN outer/JJ space/NN

Disambiguating "race"

- Two possible sequences of tags for our running example



source: J&M, Fig. 5.12

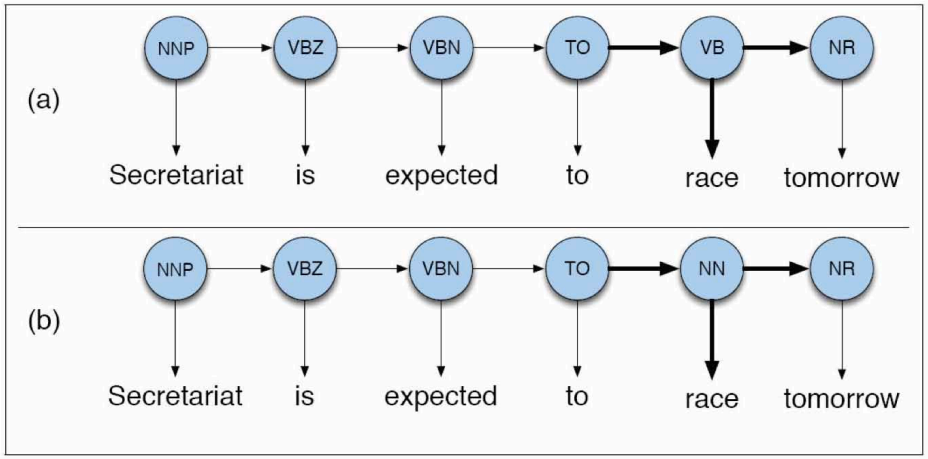
Disambiguating "race"

- From the Brown corpus:

  P(NN | TO) = .00047
  P(VB | TO) = .83

  P(race | NN) = .00057
  P(race | VB) = .00012

  P(NR | VB) = .0027
  P(NR | NN) = .0012



- Computing the probabilities surrounding "race", we have:

  P(VB | TO) P(race | VB) P(NR | VB) = .00000027
  P(NN | TO) P(race | NN) P(NR | NN) = .00000000032

- So, we (correctly) choose VB as the tag for "race"

Markov Chains

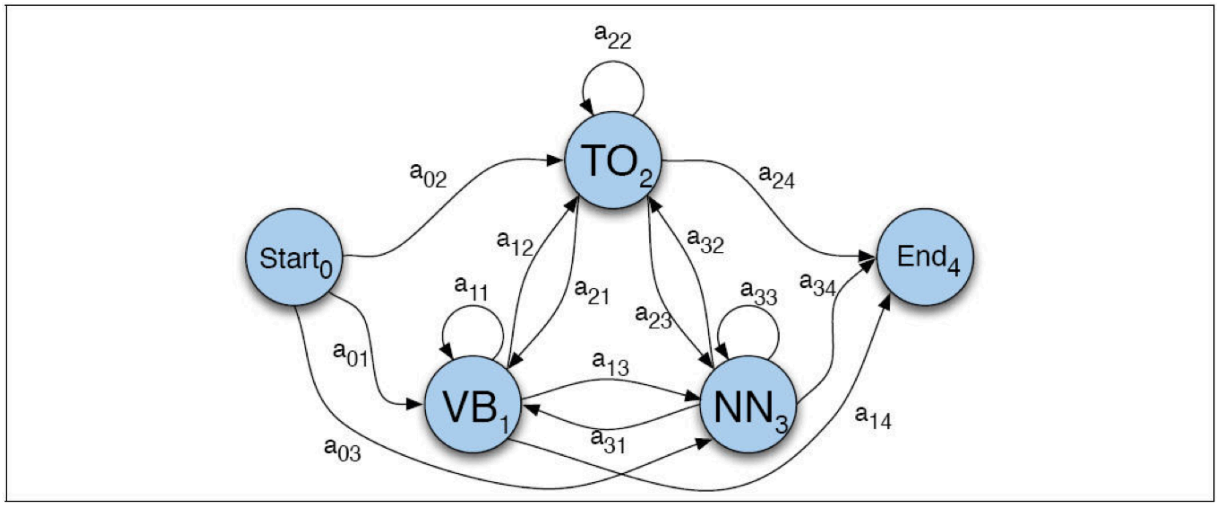
- To formalize what we have just illustrated, we will examine
  - Markov chains
  - Hidden Markov Models

- **Markov Chain: "First-order observable Markov Model"**

- A set of n **states** $Q = q_1, q_2, \ldots, q_N$
- A **transition probability matrix** $A = a_{01}, a_{02}, \ldots, a_{nn}$ where $\sum_{j=1}^{N} a_{ij} = 1, \forall i$
- **Start state** and **end (final) state** q_0, q_f

Example: Markov Chain

- Markov chain for a short tag sequence:



source: J&M, Fig. 5.13

- Useful for computing probabilities of state sequences
  - e.g., P(Start, TO, VB, NN, End) = (a02) · (a21) · (a13) · (a34)
- But cannot represent inherently ambiguous problems

Hidden Markov Model

- **Hidden Markov Model**
- A set of n **states** — $Q = q_1, q_2, \ldots, q_N$
- A **transition probability matrix** — $A = a_{01}, a_{02}, \ldots, a_{nn}$ where $\sum_{j=1}^{N} a_{ij} = 1, \forall i$

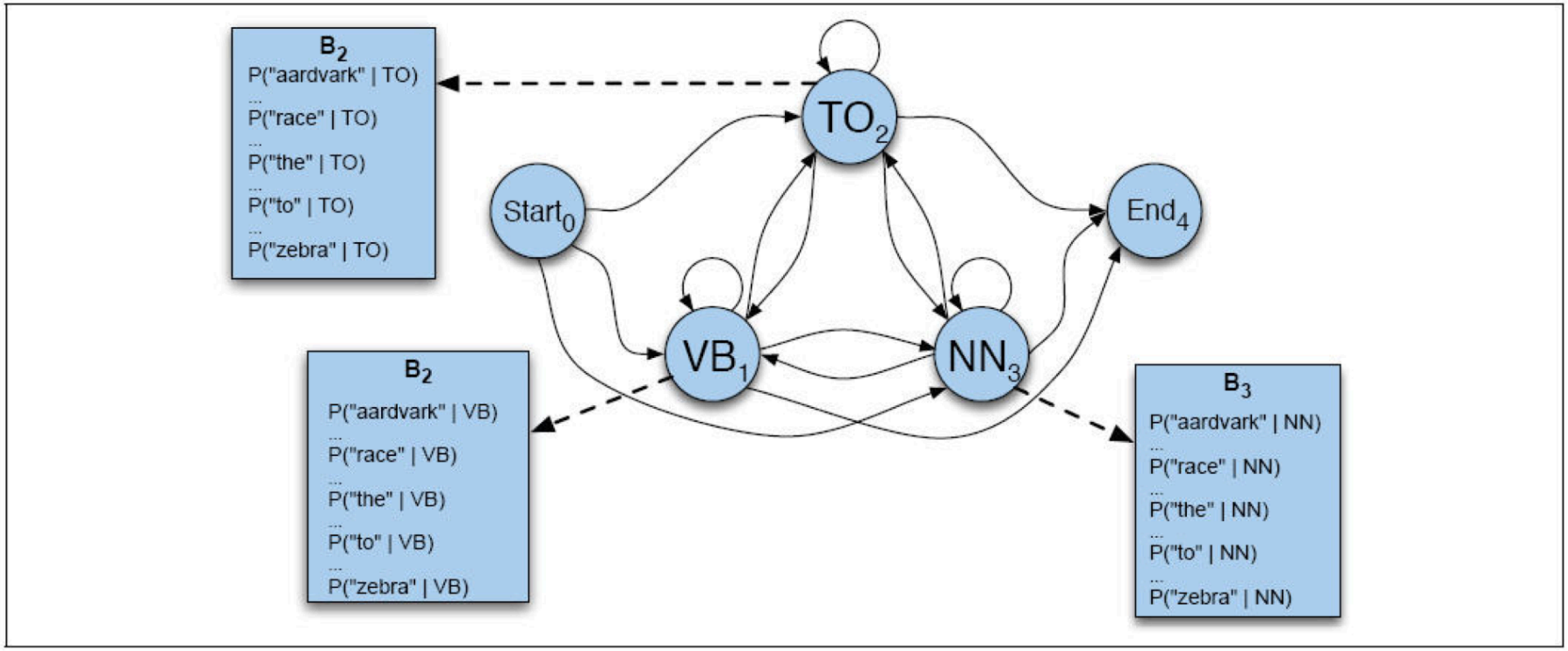
  and where $a_{ij} = P(q_t \mid q_{t-1})$
- A sequence of **observations** — $O = o_1, o_2, \ldots, o_T$
- A set of **observation likelihoods** — $B = \{ b_i(k) \}$ where $b_i(k) = P(o_k \mid q_i)$

  where B is a matrix; each row a distribution
- **Start state** and **end (final) state** — q_0, q_f, that are not associated with observations, together with transition probabilities $\{ a_{0i} \}$ out of q_0 and $\{ a_{iF} \}$ and into q_f

Note: An alternative to $\{ a_{0i} \}$ is an **initial probability distribution** over the set of states

Example: HMM

- HMM version of our short tag sequence example:



source: J&M, Fig. 5.14

Today

Parts of speech (POS)

Tagsets

POS Tagging

Rule-based tagging

Hidden Markov Models

The Viterbi algorithm

POS Tagging Using an HMM

- **most likely sequence**
  - also often called the "decoding" problem
  - want most likely tag sequence $t_1, \ldots, t_k$ for a given sentence $w_1, \ldots, w_k$
- exhaustive enumeration
  - not practical for documents
  - e.g., consider a document of 1,000 words
    - assume 40 % are ambiguous
    - assume 2.5 average branching factor
    - then, number of possibilities is 2.5^{400}
- Viterbi algorithm
  - an efficient dynamic programming method
  - basic idea: keep track of only the best path so far to each possible prior state

Viterbi Algorithm

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*

create a path probability matrix *viterbi*$[N+2,T]$
for each state s **from** 1 **to** N **do** ; initialization step
$viterbi[s,1] \leftarrow a_{0,s} * b_s(o_1)$
$backpointer[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ; recursion step
for each state s **from** 1 **to** N **do**
$viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
$backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s}$
$viterbi[q_F,T] \leftarrow \max_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$; termination step
$backpointer[q_F,T] \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$; termination step
return the backtrace path by following backpointers to states back in time from $backpointer[q_F,T]$

source: J&M, Fig. 5.14

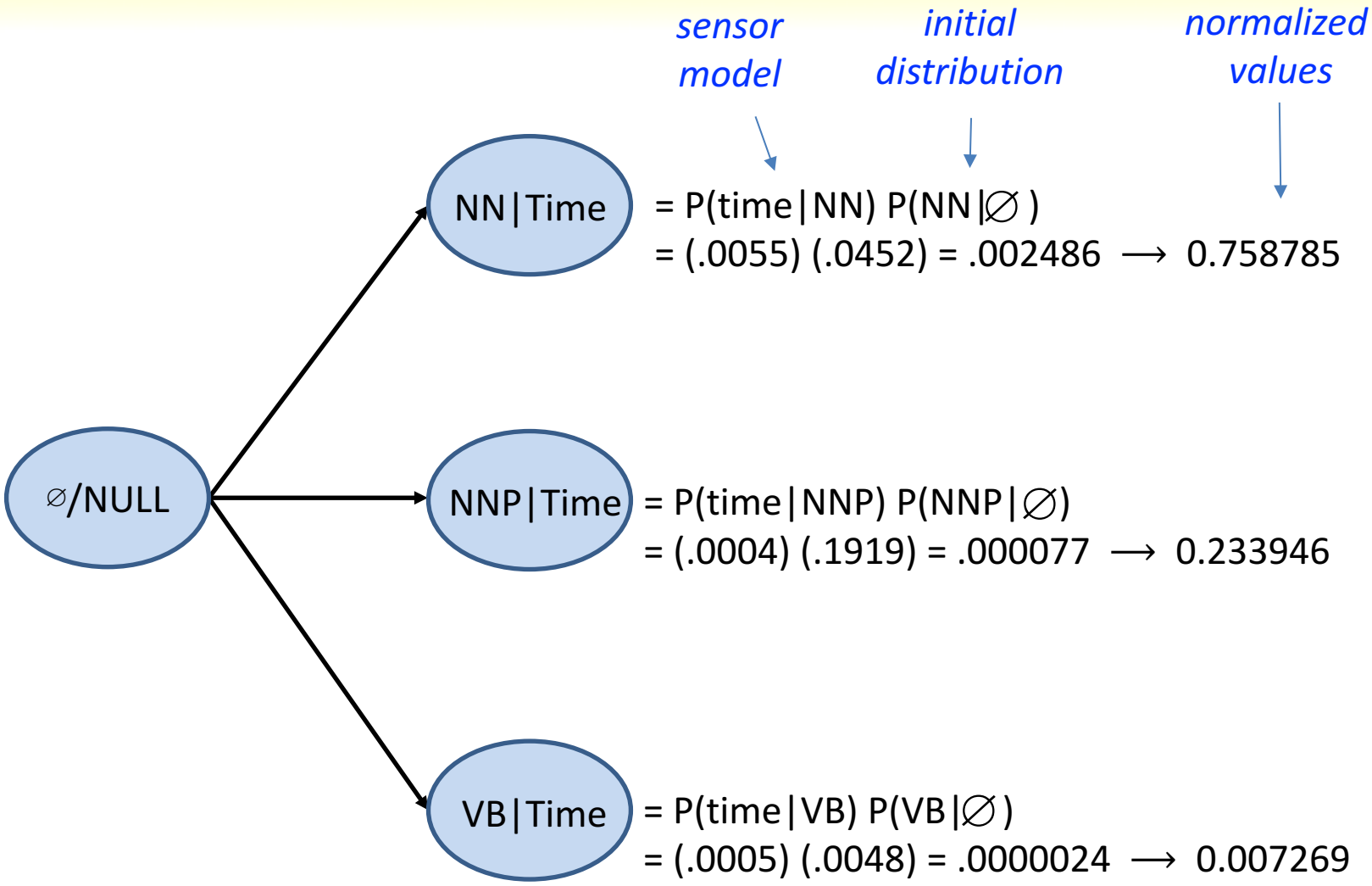
POS Tagging: "Time travel will work"

- CoNLL 2000 sample corpus
  - tagged and chunked data from a Wall Street Journal corpus
  - chunk tags were removed since not needed for POS tagging
- Penn Treebank Tagset

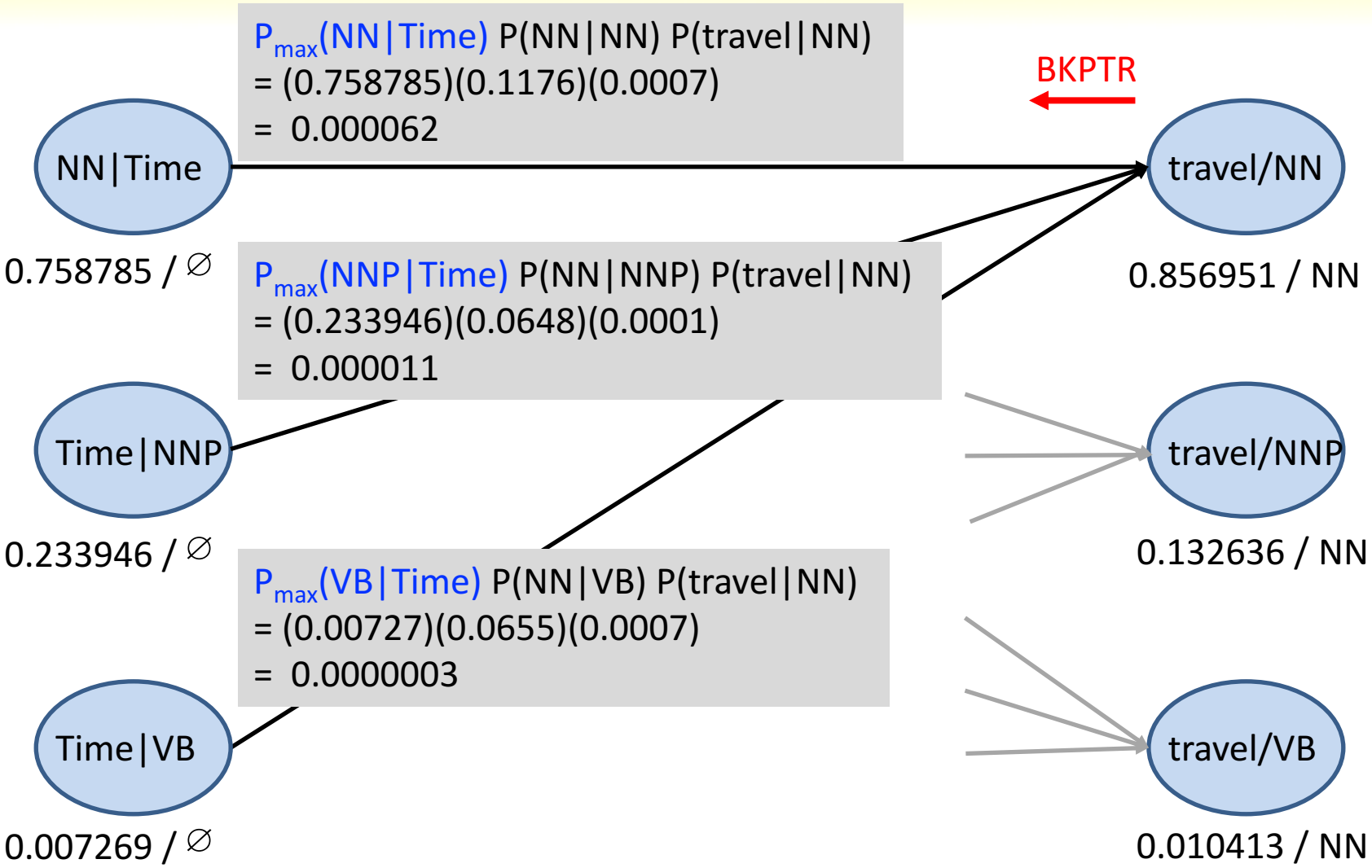
Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

```
Confidence NN
in IN
the DT
pound NN
is VBZ
widely RB
expected VBN
to TO
take VB
another DT
sharp JJ
dive NN
if IN
trade NN
figures NNS
for IN
September NNP
, ,
due JJ
for IN
release NN
tomorrow NN
, ,
fail VB
to TO
show VB
a DT
substantial JJ
improvement NN
from IN
July NNP
and CC
August NNP
's POS
near-record JJ
deficits NNS
. .
```

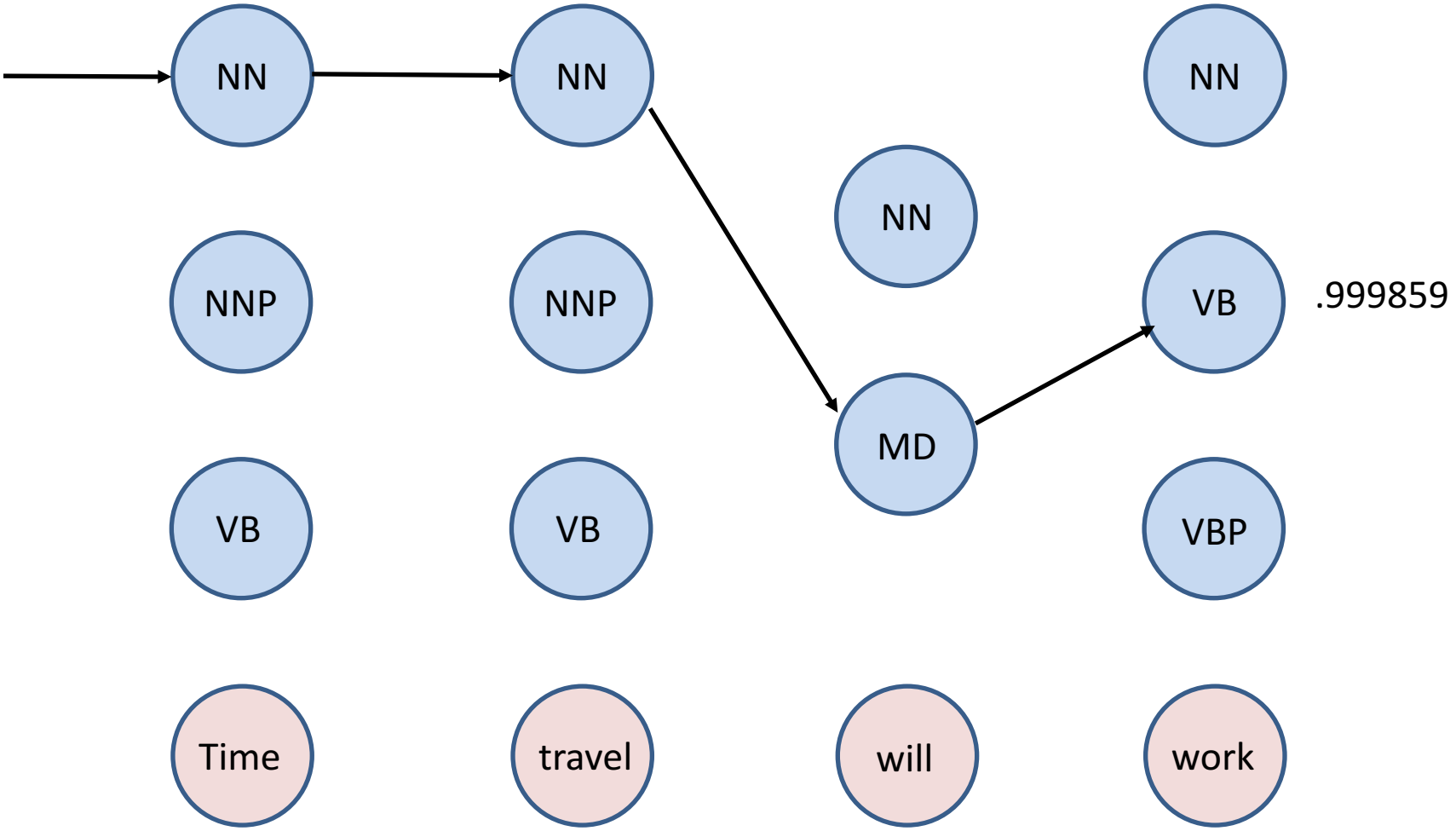
Time : NN (0.0055) NNP (0.0004) VB (0.0005)



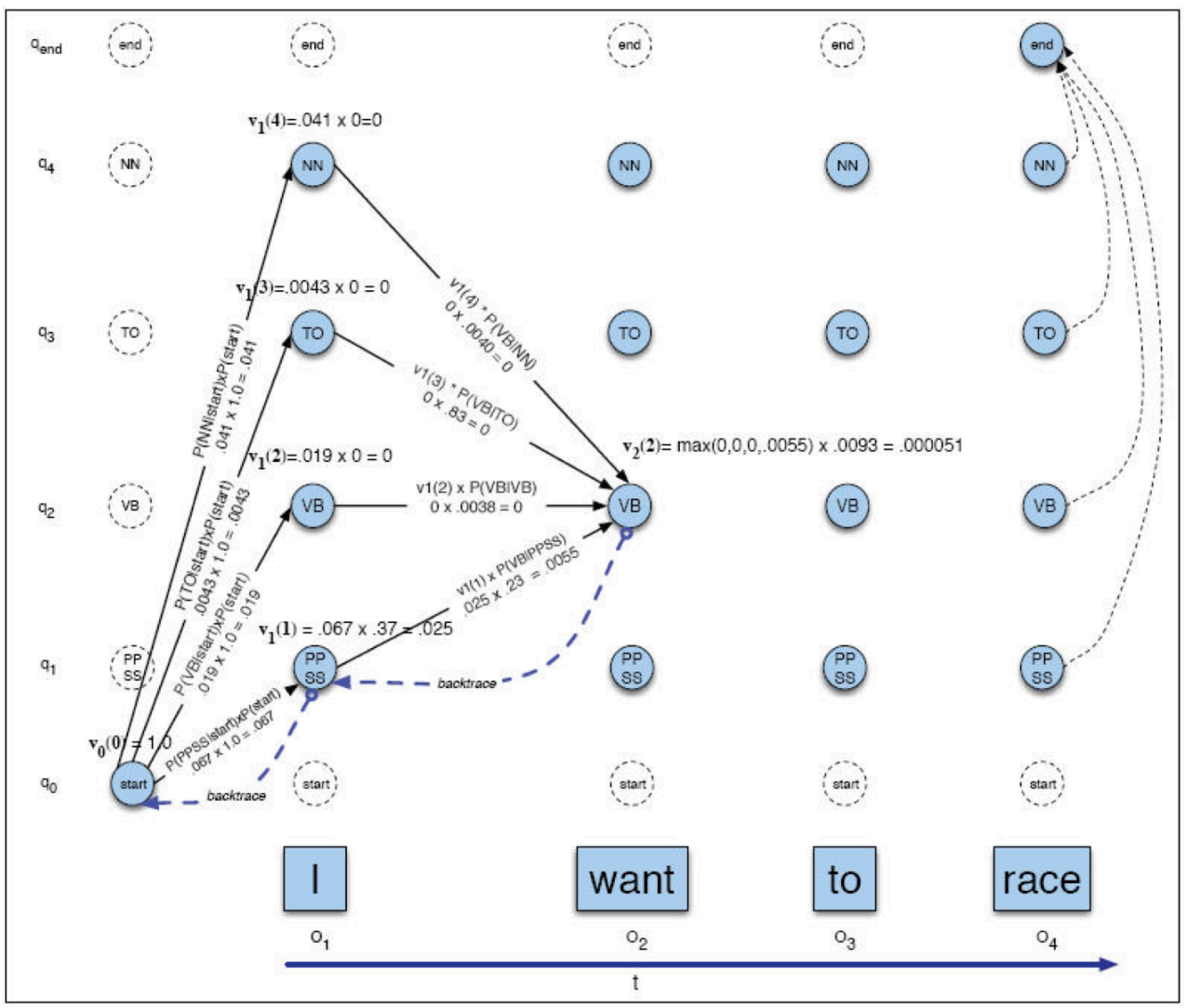
travel : NN (0.0007) NNP (0.0001) VB (0.0007)



Most Likely Path



Another Viterbi Example



source: J&M, Fig. 5.18

Viterbi Summary

- Use an array
  - columns are inputs
  - rows are possible states
- Sweep through the array in one pass
  - fill columns left-to-right
  - use transition and observation probabilities
- Backtrace
  - store only the MAX probability to each cell
  - not every path