Word classes and part of speech tagging

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

Word Classes: Tag Sets

- Vary in number of tags: a dozen to over 200
- Size of tag sets depends on language, objectives and purpose

Parts of Speech

- Perhaps starting with Aristotle in the West (384–322 BCE), there was the idea of having parts of speech
  - a.k.a lexical categories, word classes, “tags”, POS
- It comes from Dionysius Thrax of Alexandria (c. 100 BCE) the idea that is still with us that there are 8 parts of speech
  - But actually his 8 aren’t exactly the ones we are taught today
    - Thrax: noun, verb, article, adverb, preposition, conjunction, participle, pronoun
    - School grammar: noun, verb, adjective, adverb, preposition, conjunction, pronoun, interjection

Open class (lexical) words

Nouns

Proper
IBM
Italy

Common
cat / cats
snow

Verbs

Main
see
registered

Modals
can
had

Adjectives *old older oldest*

Adverbs *slowly*

Numbers
122,312
one

… more

Closed class (functional)

Determiners *the some*

Conjunctions *and or*

Pronouns *he its*

Prepositions *to with*

Particles *off up*

Interjections *Ow Eh*

… more

Open vs. Closed classes

Open vs. Closed classes

- Closed:
  - determiners: ***a, an, the***
  - pronouns: ***she, he, I***
  - prepositions: ***on, under, over, near, by, …***
  - Why "closed"?
- Open:
  - Nouns, Verbs, Adjectives, Adverbs.

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

Input: Plays well with others

Ambiguity: NNS/VBZ UH/JJ/NN/RB IN NNS

Output: Plays/VBZ well/RB with/IN others/NNS

Uses:

- Text-to-speech (how do we pronounce "lead"?)
- Can write regexps like (Det) Adj* N+ over the output for phrases, etc.
- As input to or to speed up a full parser
- If you know the tag, you can back off to it in other tasks

Penn Treebank POS tags

POS tagging performance

How many tags are correct? (Tag accuracy)

- About 97% currently
- But baseline is already 90%
  - Baseline is performance of stupidest possible method
    - Tag every word with its most frequent tag
    - Tag unknown words as nouns
- Partly easy because
  - Many words are unambiguous
  - You get points for them (*the*, *a*, etc.) and for punctuation marks!

Deciding on the correct part of speech can be difficult even for people

Mrs/NNP Shaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG

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

Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

How difficult is POS tagging?

About 11% of the word types in the Brown corpus are ambiguous with regard to part of speech

But they tend to be very common words. E.g., *that*

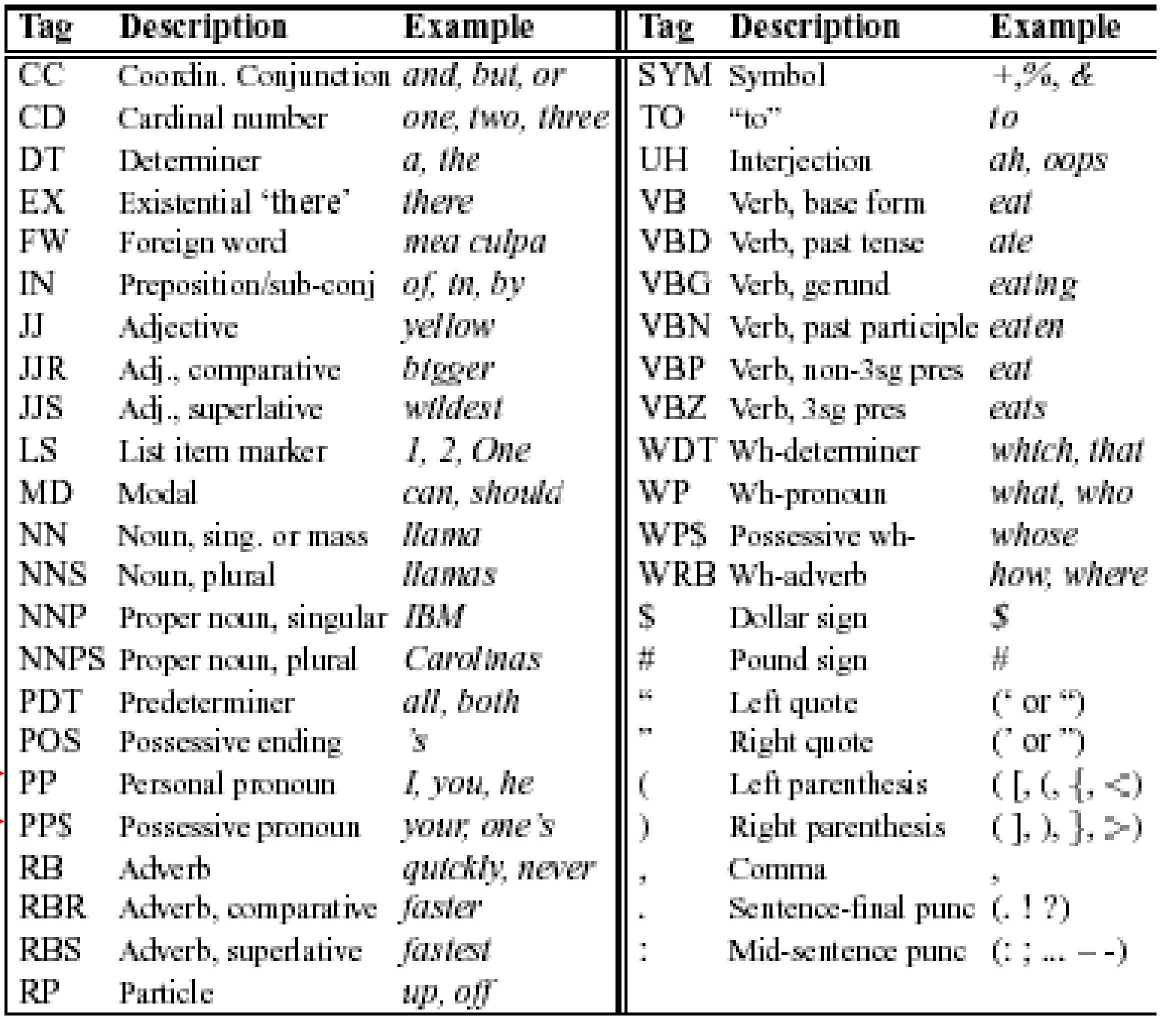
I know *that* he is honest = IN

Yes, *that* play was nice = DT

You can’t go *that* far = RB

40% of the word tokens are ambiguous

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 ?
```

See http://www.infogistics.com/posdemo.htm

Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo

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

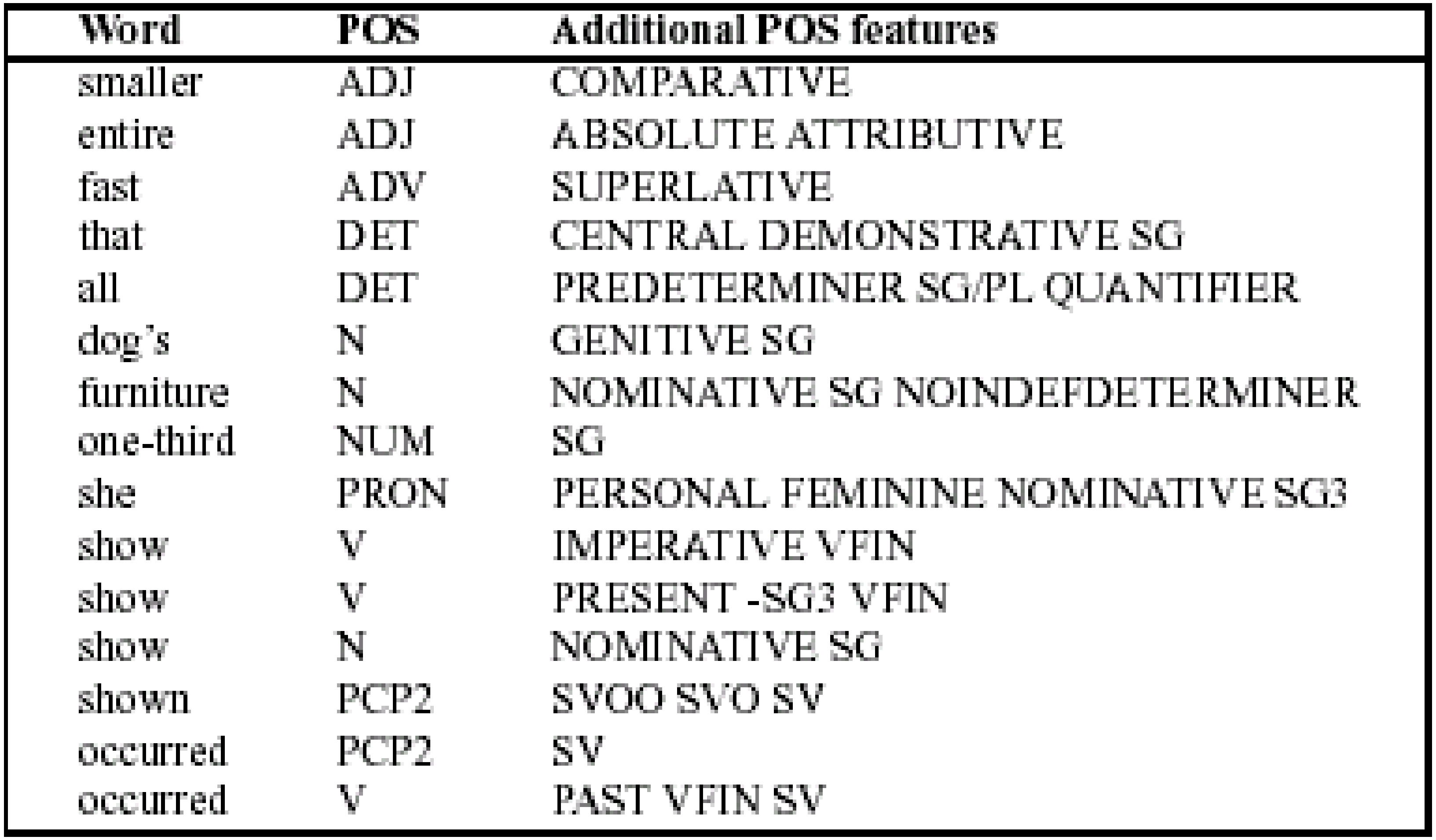
(Derose, 1988)

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

Sample ENGTWOL Lexicon

Demo: http://www2.lingsoft.fi/cgi-bin/engtwol



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 a 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
- Requires a training corpus
- No probabilities for words not in corpus.

Stochastic Tagging (cont.)

- Simple Method: Choose most frequent tag in training text for each word!
  - Result: 90% accuracy
  - Baseline
  - Others will do better
  - HMM is an example

HMM Tagger

- Intuition: Pick the most likely tag for this word.
- 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.

Toward a Bigram-HMM Tagger

$\mathrm{argmax}_T\ P(T|W)$

$\mathrm{argmax}_T P(T)P(W|T)$

$\mathrm{argmax}_t P(t_1 \ldots t_n)P(w_1 \ldots w_n|t_1 \ldots t_n)$

$\mathrm{argmax}_t [P(t_1)P(t_2|t_1) \ldots P(t_n|t_{n-1})][P(w_1|t_1)P(w_2|t_2) \ldots P(w_n|t_n)]$

To tag a single word: $t_i = \mathrm{argmax}_j\ P(t_j|t_{i-1})P(w_i|t_j)$

How do we compute $P(t_i|t_{i-1})$?

- $c(t_{i-1}t_i)/c(t_{i-1})$

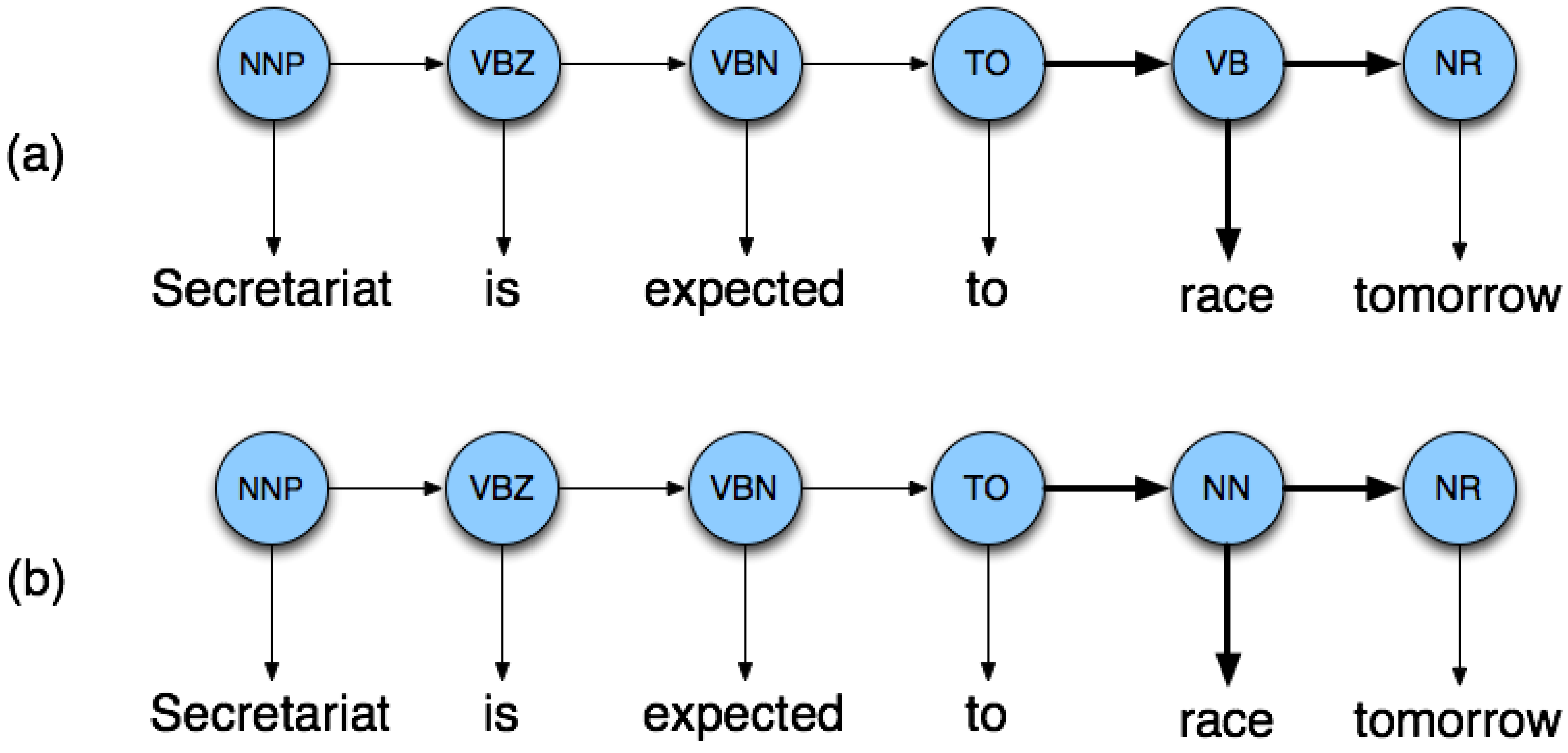
How do we compute $P(w_i|t_i)$?

- $c(w_i,t_i)/c(t_i)$

How do we compute the most probable tag sequence?

- Viterbi

Disambiguating “race”



Example

P(NN|TO) = .00047

P(VB|TO) = .83

P(race|NN) = .00057

P(race|VB) = .00012

P(NR|VB) = .0027

P(NR|NN) = .0012

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 reading,

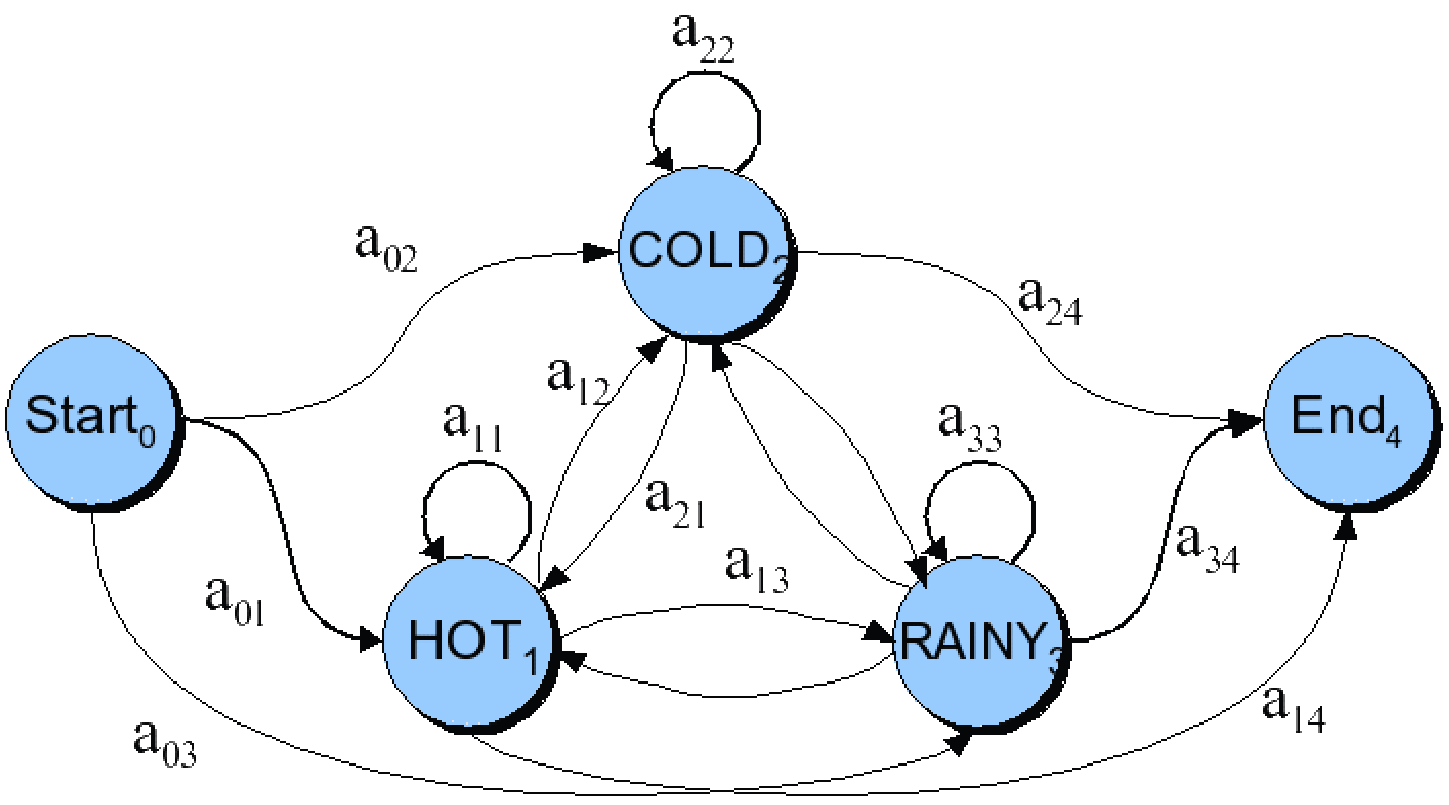
Hidden Markov Models

What we’ve described with these two kinds of probabilities is a Hidden Markov Model (HMM)

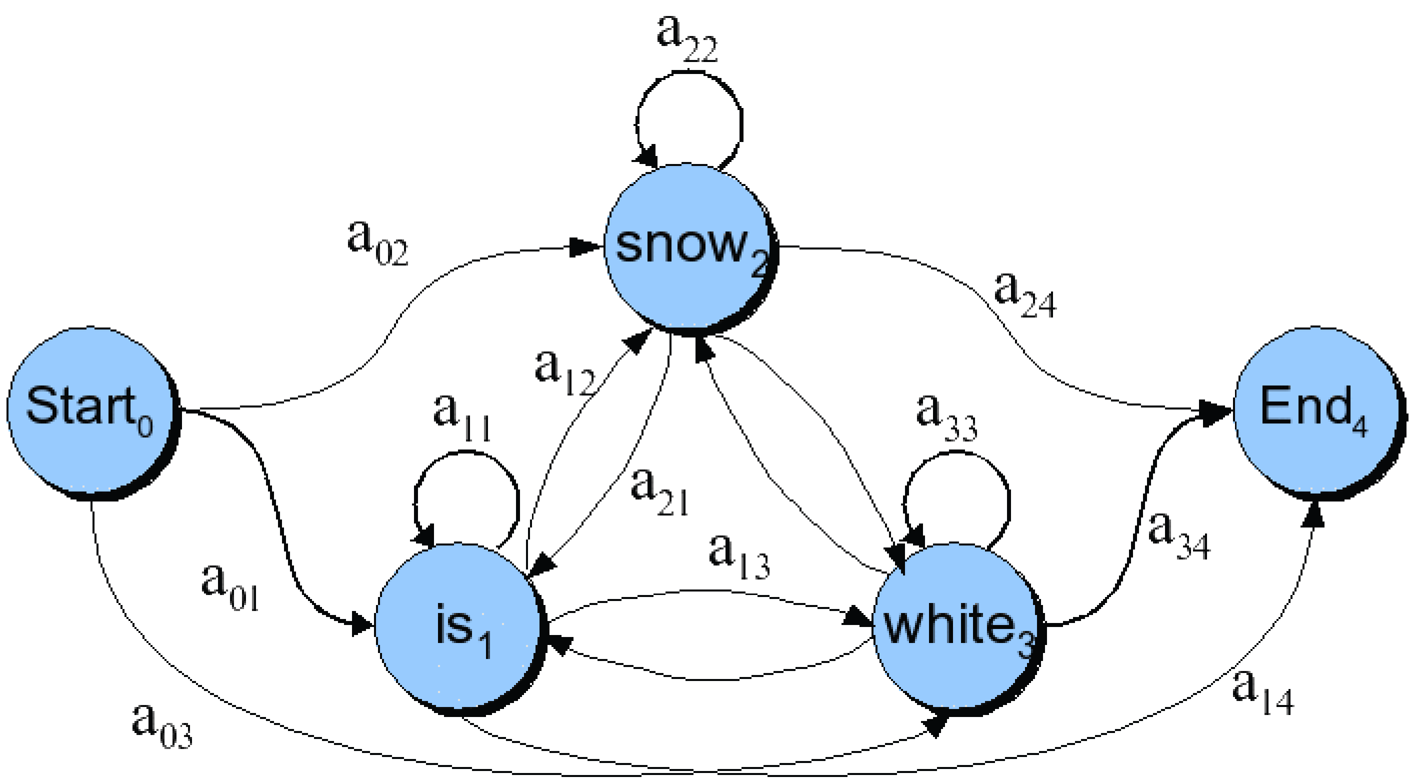
Definitions

- A weighted finite-state automaton adds probabilities to the arcs
  - The sum of the probabilities on arcs leaving a node must sum to one
- A Markov chain is a special case of a WFST in which the input sequence uniquely determines which states the automaton will go through
- Markov chains can't represent ambiguous problems
  - Useful for assigning probabilities to unambiguous sequences

Markov Chain for Weather



Markov Chain for Words



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

Current state only depends on previous state

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

Hidden Markov Model

For Markov chains, the symbols are the same as the states.

- See **hot** weather: we' re in state **hot**

But in part-of-speech tagging

- The output symbols are **words**
- But the hidden states are **part-of-speech tags**

A Hidden Markov Model is an extension of a Markov chain in which the input symbols are not the same as the states.

This means we don' t know which state we are in.

Hidden Markov Models

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 $A = \{a_{ij}\}$

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

Special initial probability vector π

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

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

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

Eisner Task

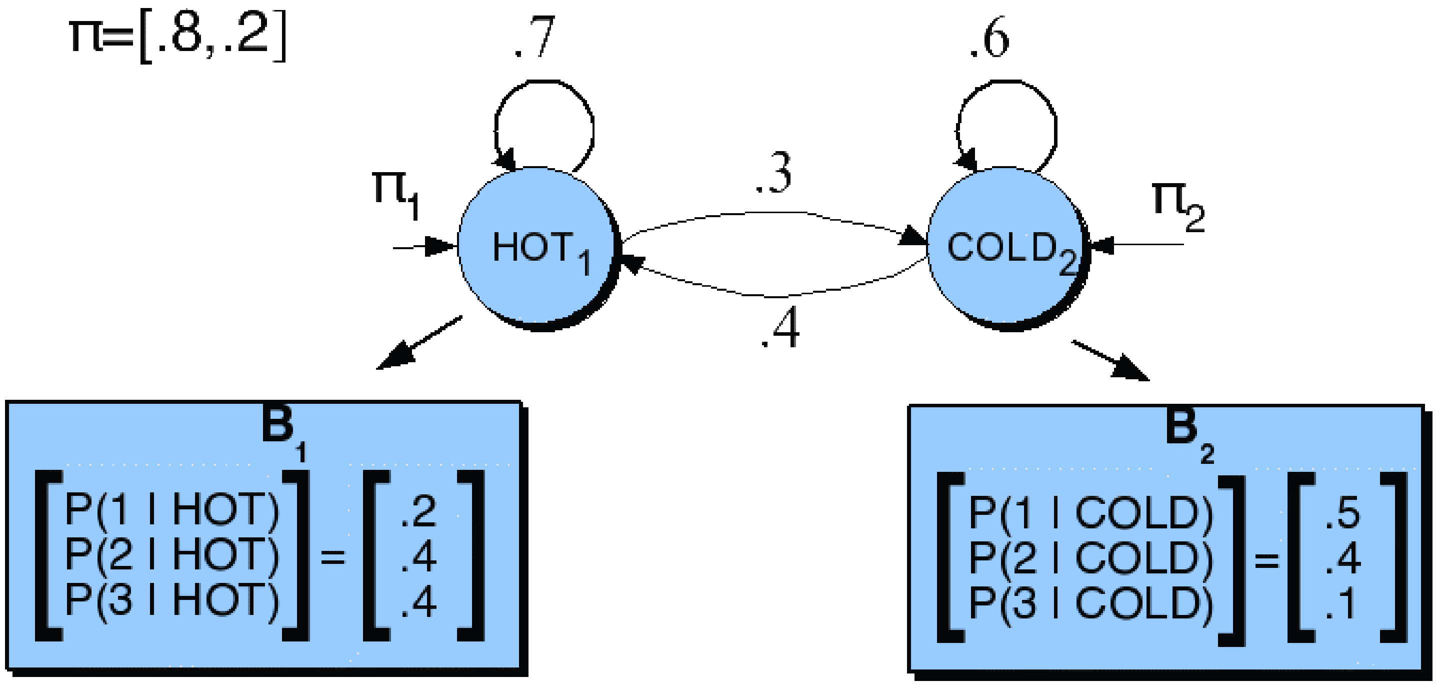
Given

- Ice Cream Observation Sequence: 1,2,3,2,2,2,3…

Produce:

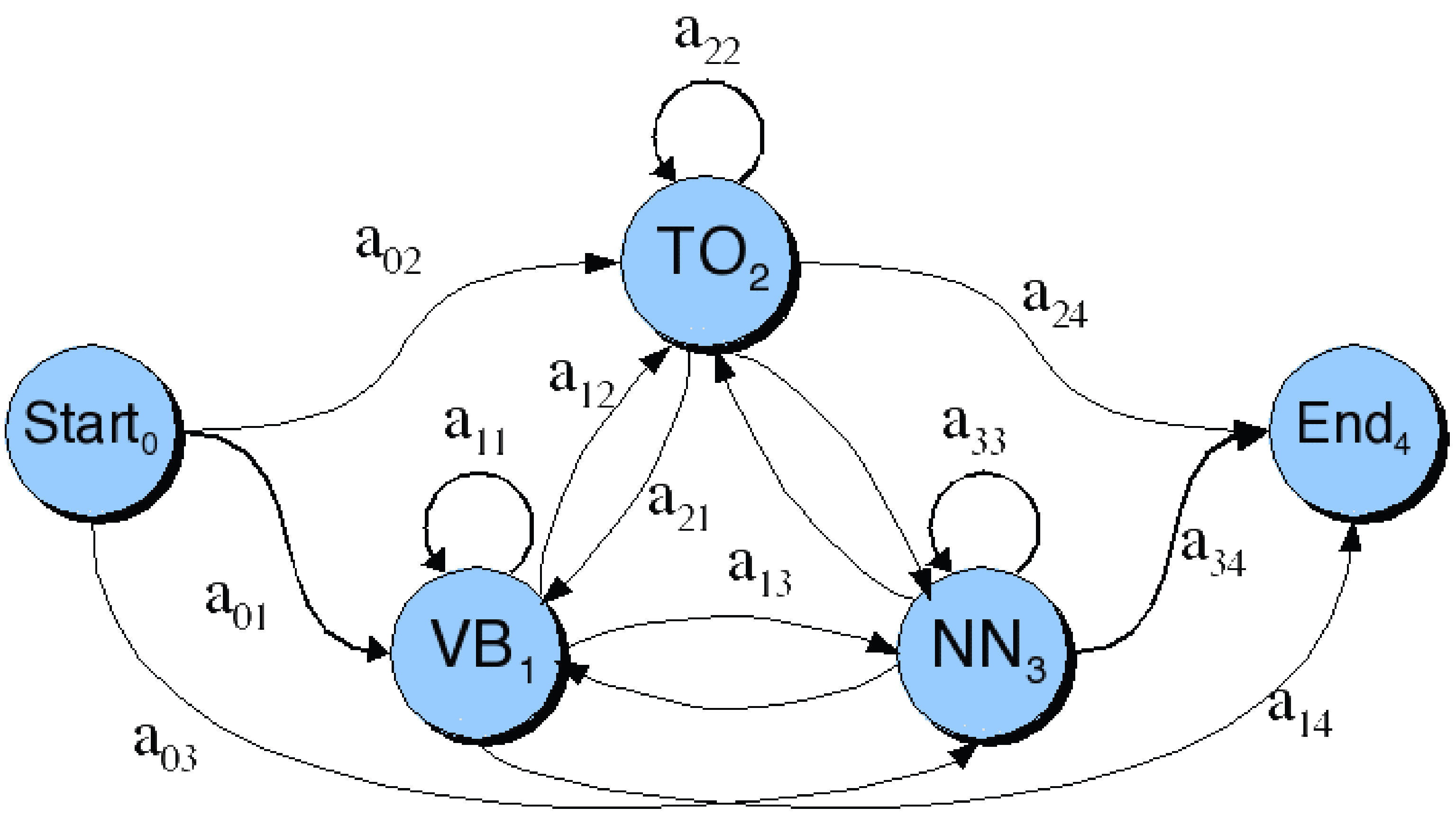
- Weather Sequence: H,C,H,H,H,C…

HMM for Ice Cream

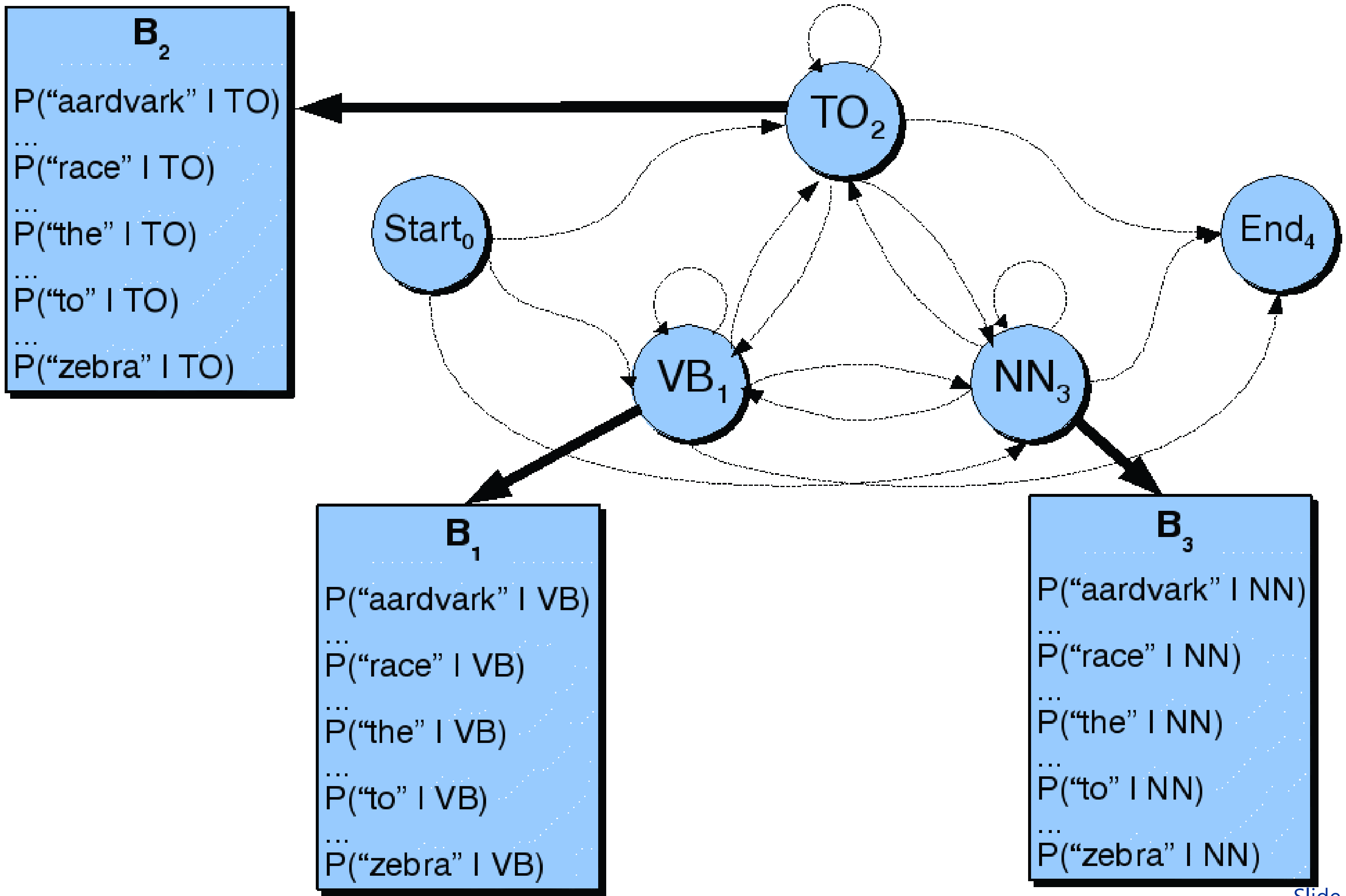


$\pi = [.8, .2]$
.7
.6
.3
.4
π_1
π_2
HOT_1
$COLD_2$
B_1
$\begin{bmatrix} P(1 \mid HOT) \\ P(2 \mid HOT) \\ P(3 \mid HOT) \end{bmatrix} = \begin{bmatrix} .2 \\ .4 \\ .4 \end{bmatrix}$
B_2
$\begin{bmatrix} P(1 \mid COLD) \\ P(2 \mid COLD) \\ P(3 \mid COLD) \end{bmatrix} = \begin{bmatrix} .5 \\ .4 \\ .1 \end{bmatrix}$

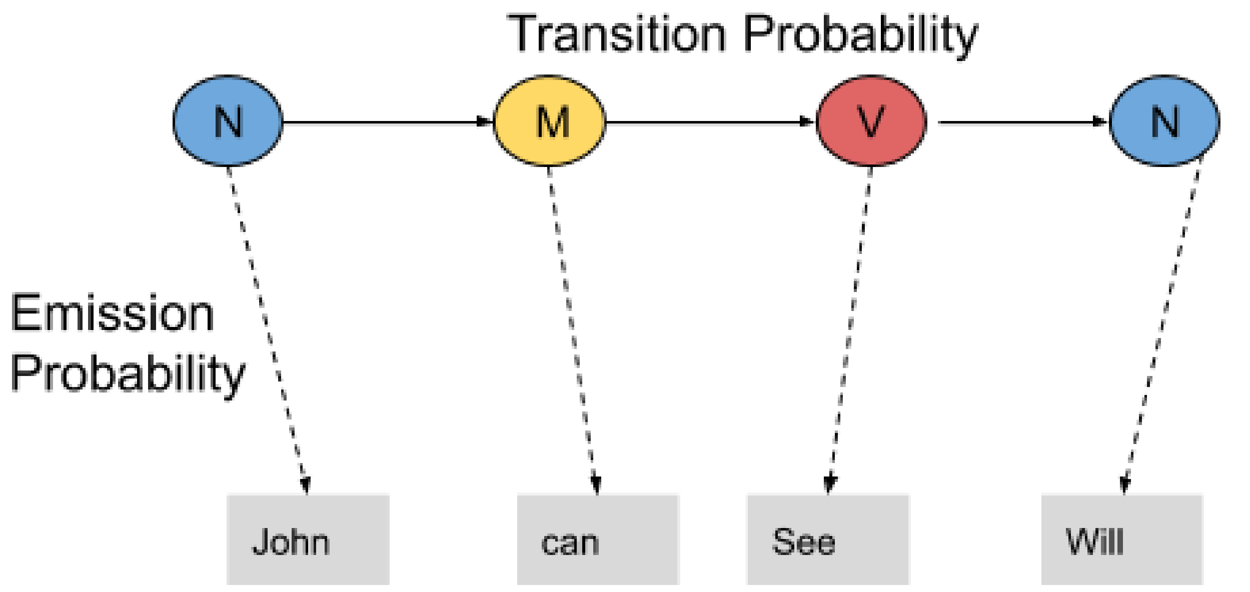
Transition Probabilities



Observation Likelihoods



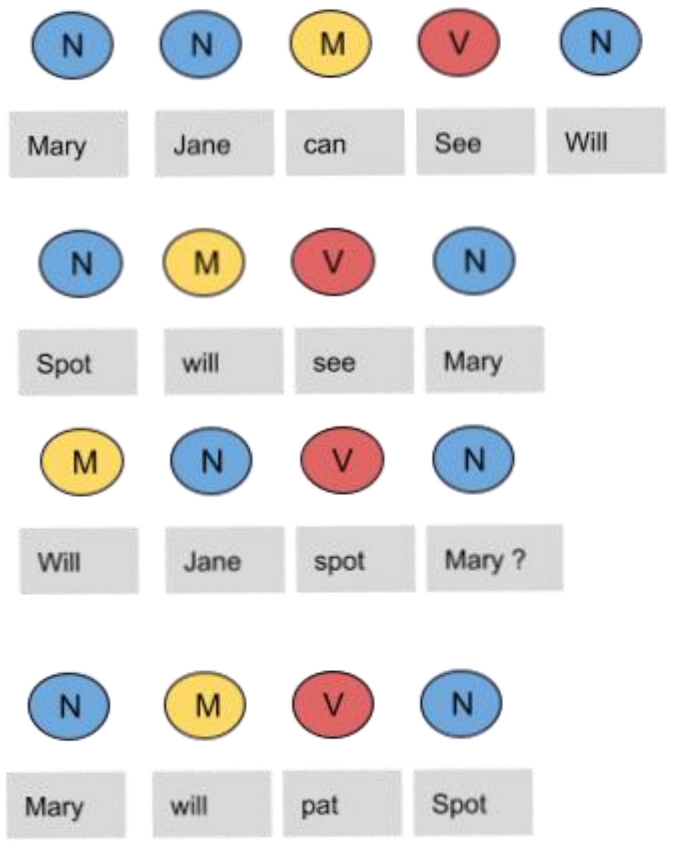
HMM Example


Transition Probability
N
M
V
N
Emission Probability
John
can
See
Will

Example Data

Mary Jane can see Will
Spot will see Mary
Will Jane spot Mary?
Mary will pat Spot

Representation in Tagged Form



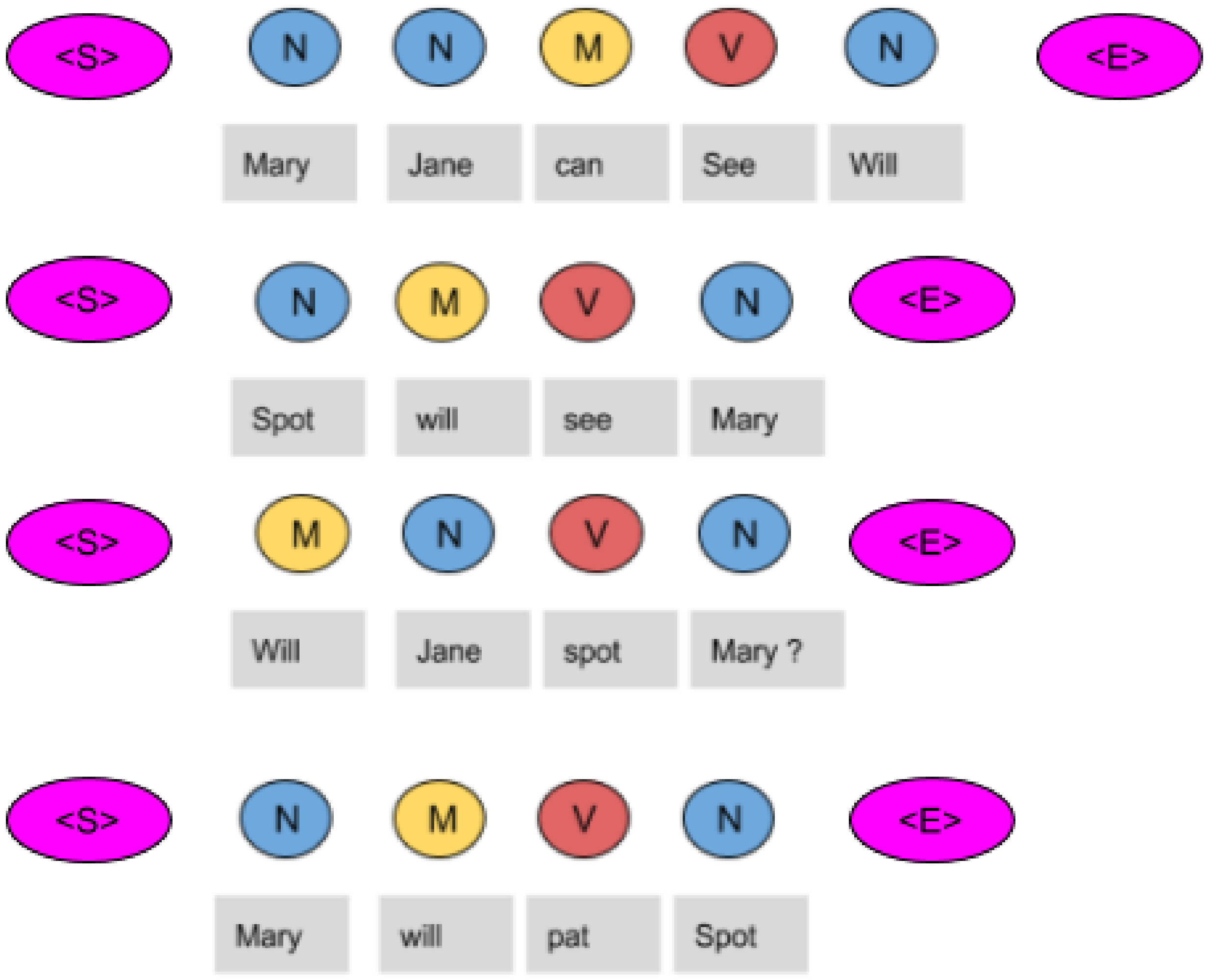
Emission Probabilities Computation

Words	Noun	Model	Verb
Mary	4	0	0
Jane	2	0	0
Will	1	3	0
Spot	2	0	1
Can	0	1	0
See	0	0	2
pat	0	0	1

Emission Probabilities Computation

Words	**Noun**	**Model**	**Verb**
Mary	4/9	0	0
Jane	2/9	0	0
Will	1/9	3/4	0
Spot	2/9	0	1/4
Can	0	1/4	0
See	0	0	2/4
pat	0	0	1

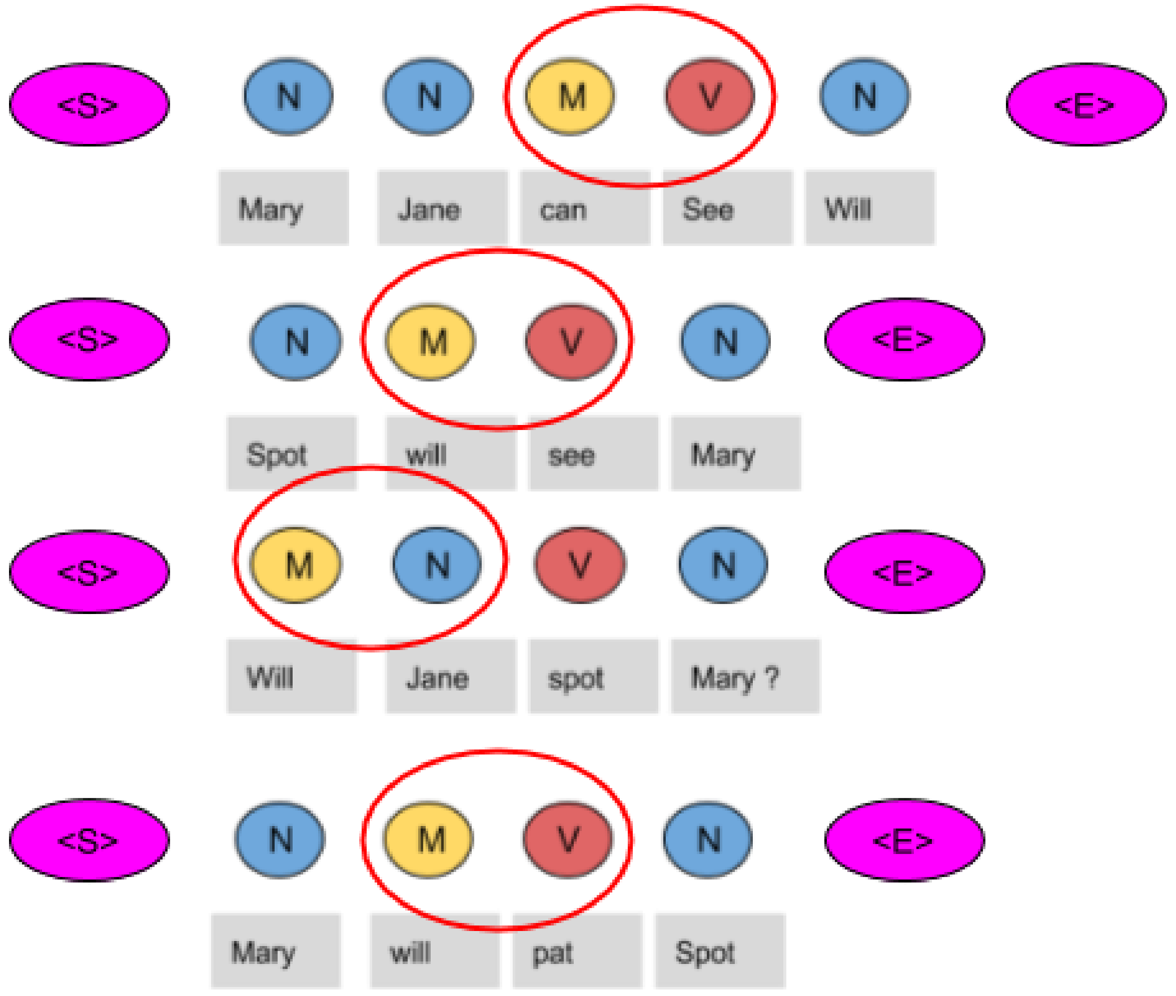
Transition Probabilities



Tag Co-occurrence Matrix

	N	M	V	<E>
<S>	3	1	0	0
N	1	3	1	4
M	1	0	3	0
V	4	0	0	0

Tag Co-occurrence Cont.



Tag Co-occurrence Matrix Final

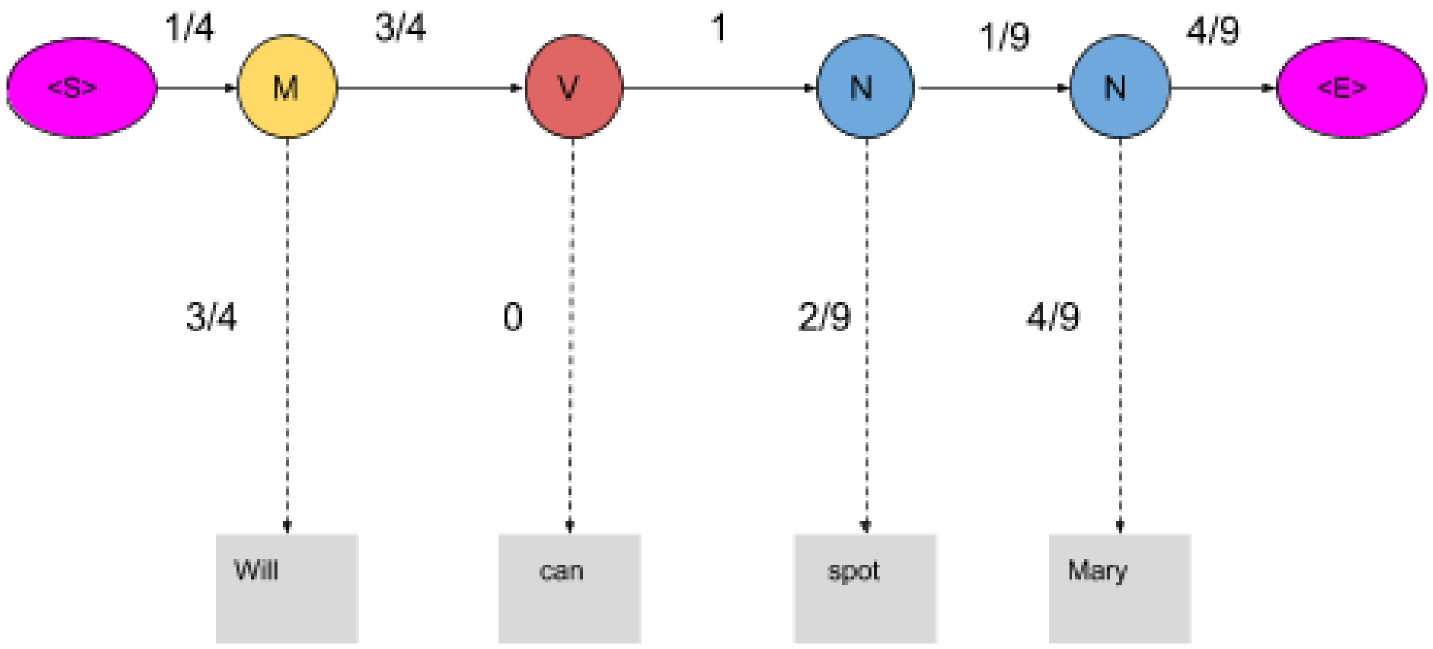
	N	**M**	**V**	**<E>**
<S>	3/4	1/4	0	0
N	1/9	3/9	1/9	4/9
M	1/4	0	3/4	0
V	4/4	0	0	0

Sentence to Tag

Let the sentence, ‘ Will can spot Mary’ be tagged as-

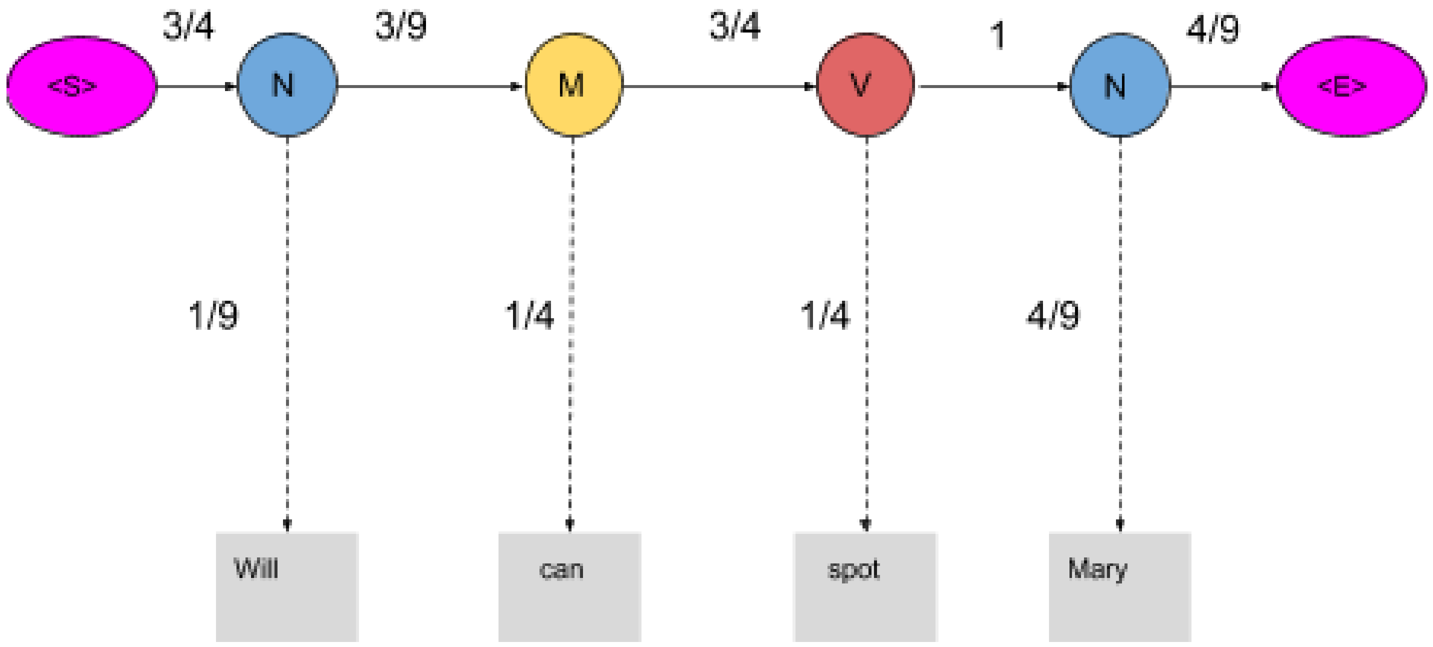
Will as a model
Can as a verb
Spot as a noun
Mary as a noun

Probability Calculation



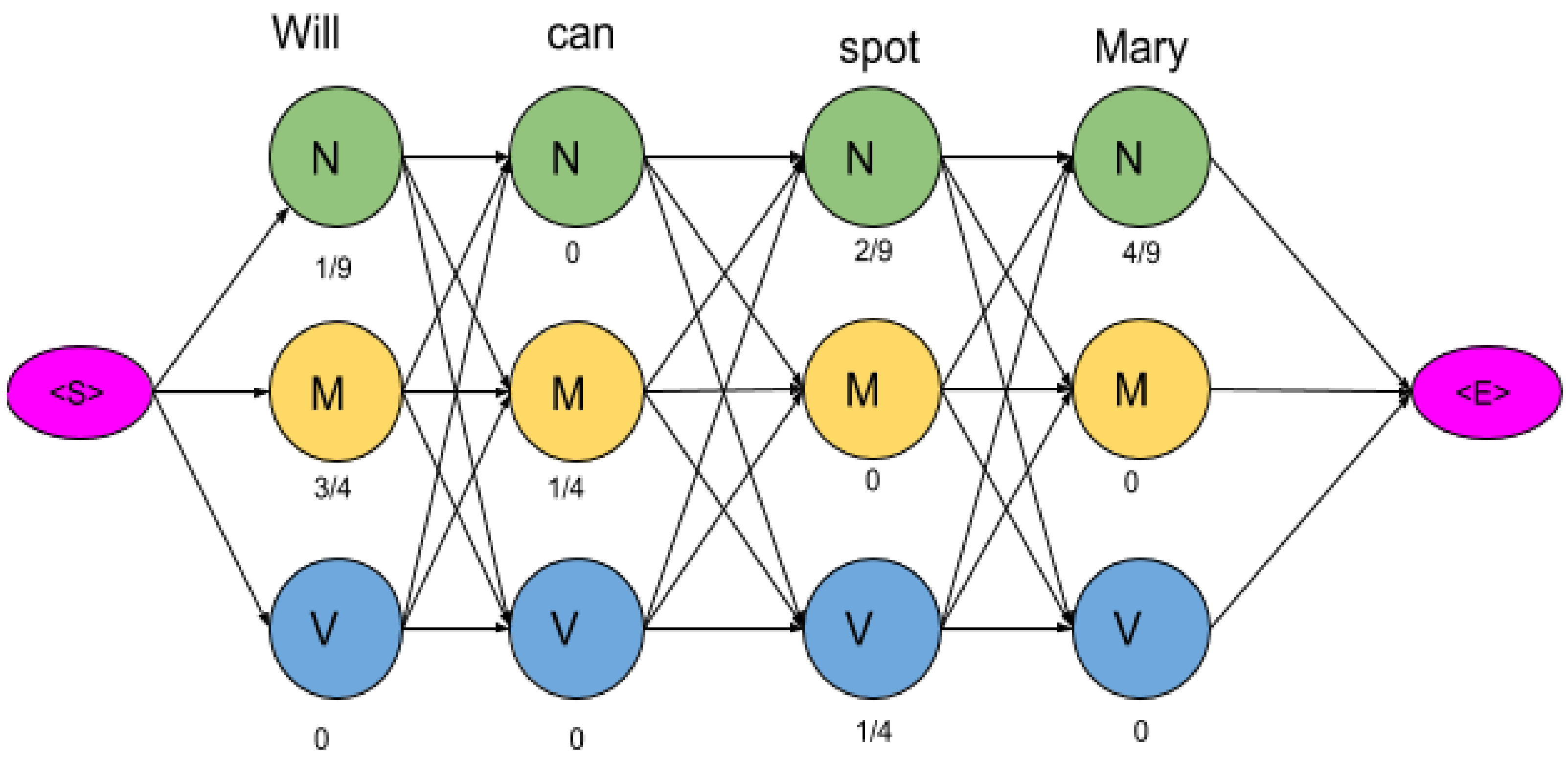
1/4*3/4*3/4*0*1*2/9*1/9*4/9*4/9=0

Alternate Tag Sequence

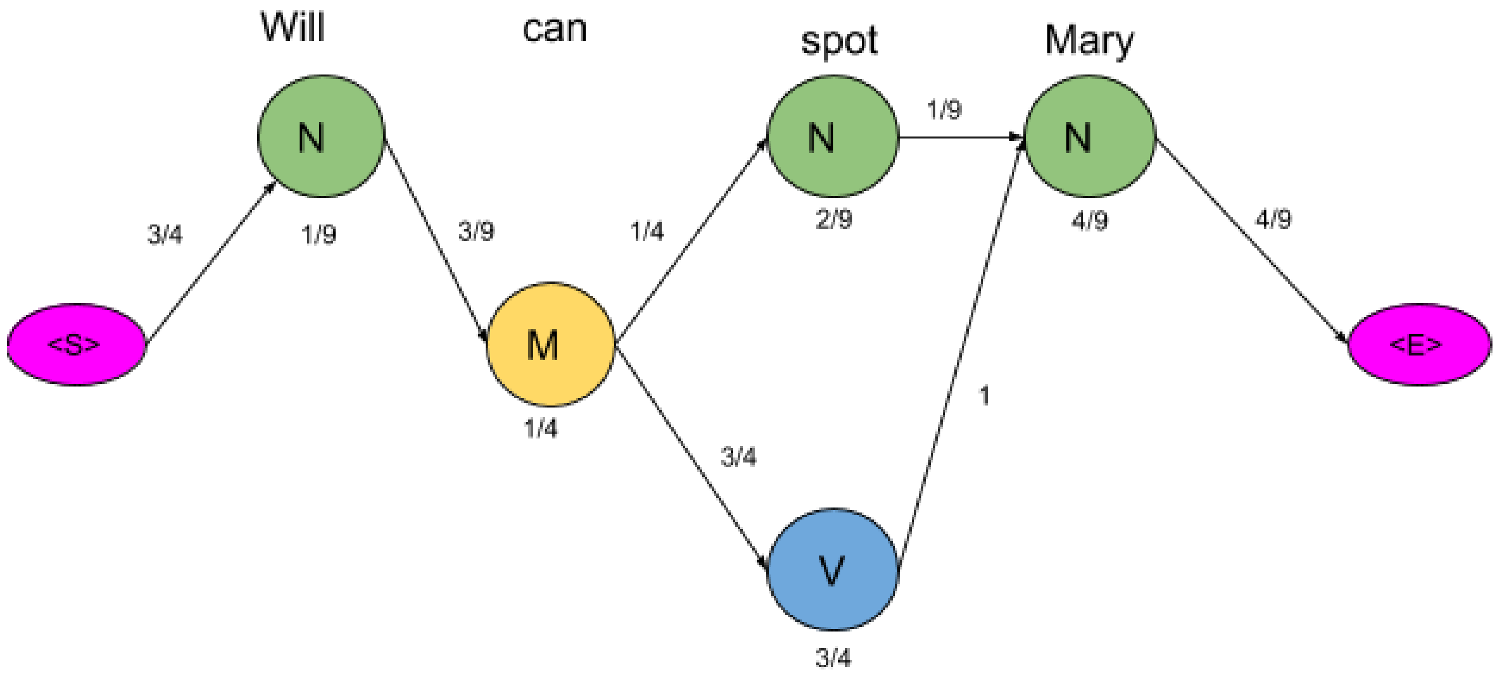


3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9=0.00025720164

All 81 Combinations



Removing 0 Edges



Two remaining sequences

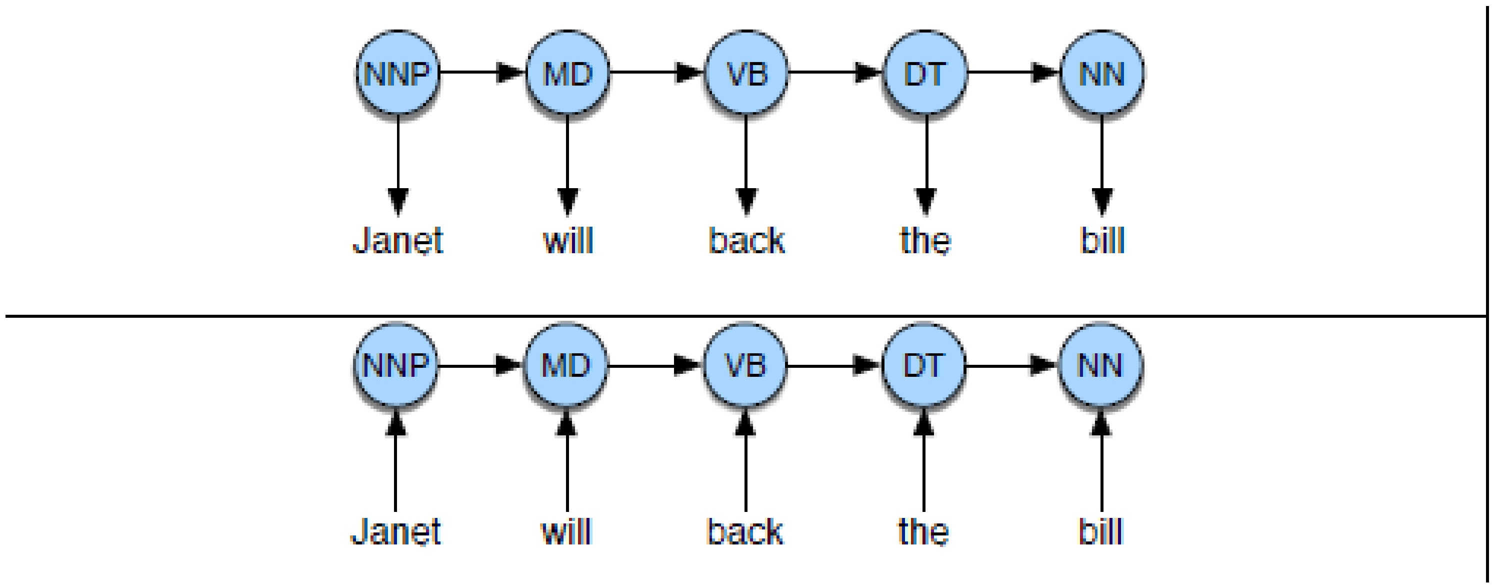
<S>→N→M→N→N→<E> =**3/4*1/9*3/9*1/4*1/4*2/9*1/9*4/9*4/9=0.00000846754**

<S>→N→M→N→V→<E>=**3/4*1/9*3/9*1/4*3/4*1/4*1*4/9*4/9= 0.00025720164**

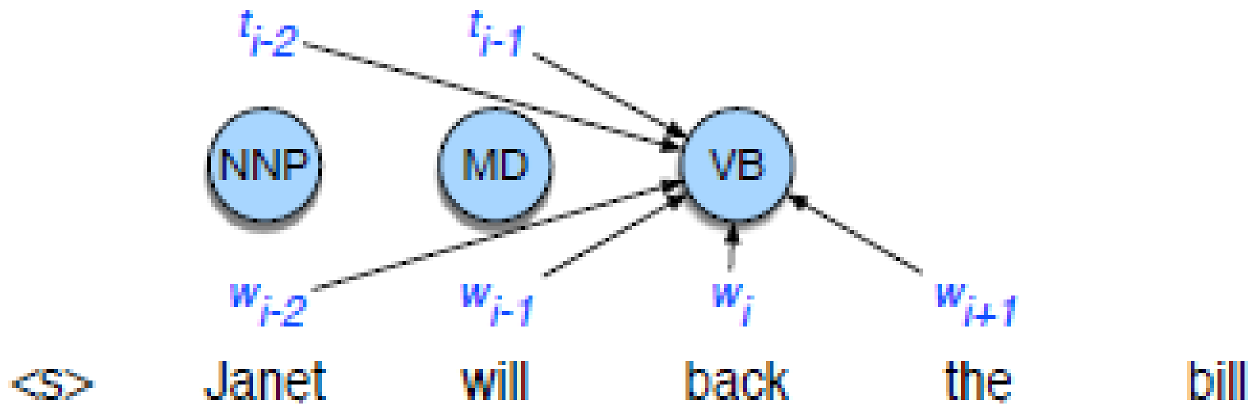
Maximum Entropy Markov Model MEMM

$$\begin{aligned}\hat{T} &= \underset{T}{\operatorname{argmax}}\, P(T|W) \\ &= \underset{T}{\operatorname{argmax}} \prod_i P(t_i|w_i, t_{i-1})\end{aligned}$$

HMM vs MEMM


NNP
MD
VB
DT
NN
Janet
will
back
the
bill
NNP
MD
VB
DT
NN
Janet
will
back
the
bill

Feature Power in MEMM

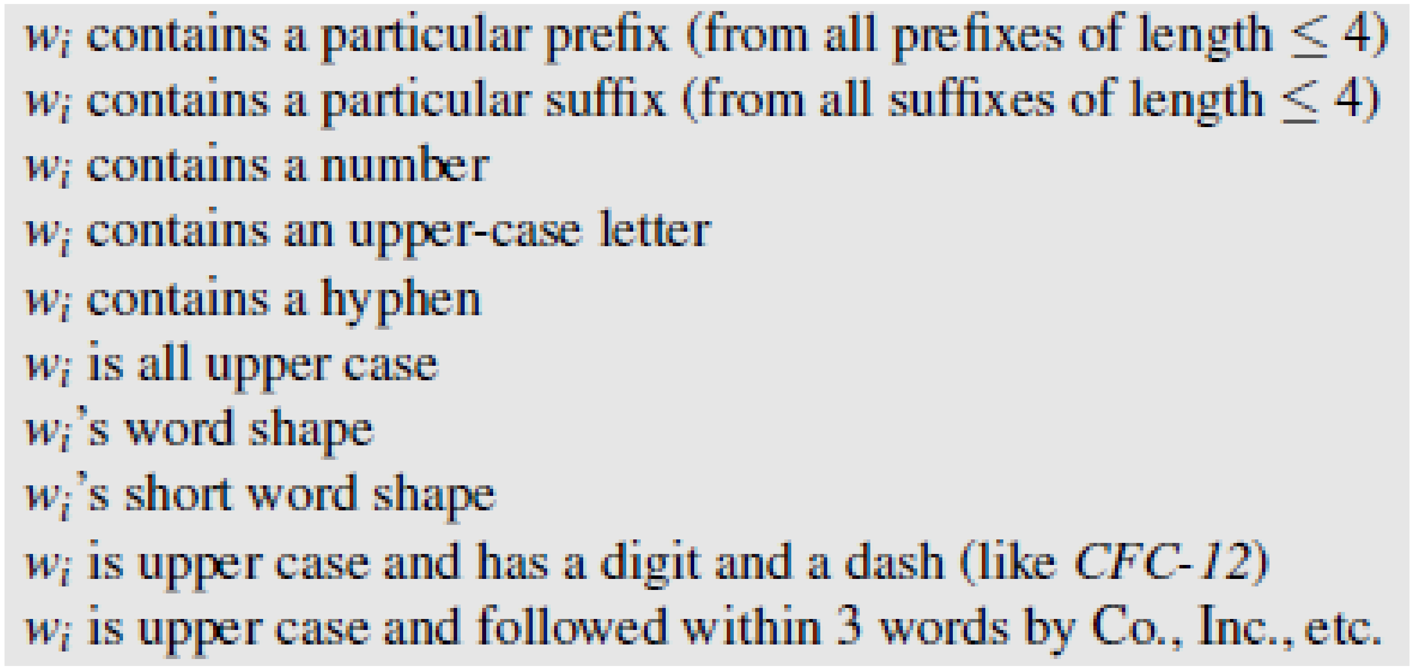


Features in MEMM

$$\langle t_i, w_{i-2}\rangle, \langle t_i, w_{i-1}\rangle, \langle t_i, w_i\rangle, \langle t_i, w_{i+1}\rangle, \langle t_i, w_{i+2}\rangle$$
$$\langle t_i, t_{i-1}\rangle, \langle t_i, t_{i-2}, t_{i-1}\rangle,$$
$$\langle t_i, t_{i-1}, w_i\rangle, \langle t_i, w_{i-1}, w_i\rangle \langle t_i, w_i, w_{i+1}\rangle,$$

t_i = VB and w_{i-2} = Janet
t_i = VB and w_{i-1} = will
t_i = VB and w_i = back
t_i = VB and w_{i+1} = the
t_i = VB and w_{i+2} = bill
t_i = VB and t_{i-1} = MD
t_i = VB and t_{i-1} = MD and t_{i-2} = NNP
t_i = VB and w_i = back and w_{i+1} = the

Word Spelling and Shape Features



w_i contains a particular prefix (from all prefixes of length ≤ 4)
w_i contains a particular suffix (from all suffixes of length ≤ 4)
w_i contains a number
w_i contains an upper-case letter
w_i contains a hyphen
w_i is all upper case
w_i's word shape
w_i's short word shape
w_i is upper case and has a digit and a dash (like *CFC-12*)
w_i is upper case and followed within 3 words by Co., Inc., etc.

Word shape and spelling features for "well-dressed"

$\text{prefix}(w_i) = \texttt{w}$

$\text{prefix}(w_i) = \texttt{we}$

$\text{prefix}(w_i) = \texttt{wel}$

$\text{prefix}(w_i) = \texttt{well}$

$\text{suffix}(w_i) = \text{ssed}$

$\text{suffix}(w_i) = \text{sed}$

$\text{suffix}(w_i) = \text{ed}$

$\text{suffix}(w_i) = \text{d}$

$\text{has-hyphen}(w_i)$

$\text{word-shape}(w_i) = \texttt{xxxx-xxxxxxx}$

$\text{short-word-shape}(w_i) = \texttt{x-x}$

Decoding MEMM

$$
\begin{aligned}
\hat{T} &= \operatorname*{argmax}_T P(T|W) \\
&= \operatorname*{argmax}_T \prod_i P(t_i|w_{i-l}^{i+l}, t_{i-k}^{i-1}) \\
&= \operatorname*{argmax}_T \prod_i \frac{\exp\left(\sum_j \theta_j f_j(t_i, w_{i-l}^{i+l}, t_{i-k}^{i-1})\right)}{\sum_{t' \in \text{tagset}} \exp\left(\sum_j \theta_j f_j(t', w_{i-l}^{i+l}, t_{i-k}^{i-1})\right)}
\end{aligned}
$$

Greedy Approach

```
for $i = 1$ to $length(W)$
    $\hat{t}_i = \underset{t' \in T}{\operatorname{argmax}} P(t' \mid w_{i-l}^{i+l}, t_{i-k}^{i-1})$
```

Viterbi

Basic $$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

HMM $$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, P(s_j|s_i)\, P(o_t|s_j) \quad 1 \leq j \leq N, 1 < t \leq T$$

MEMM $$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, P(s_j|s_i, o_t) \quad 1 \leq j \leq N, 1 < t \leq T$$