Stochastic Tagging

Bayes Rule

Markov Chains

HMM Tagging

Viterbi Algorithm

B.Senthil Kumar, Asst. Prof

Natural Language Processing

B.E. (CSE) VII Sem

Topics

- Probability
- Conditional Probability
- Bayes Rule
- HMM tagging
- Markov Chains
- Hidden Markov Models

Introduction to Probability

- Experiment (trial)
  - Repeatable procedure with well-defined possible outcomes
- Sample Space (S)
    - the set of all possible outcomes
    - *finite or infinite*
  - Example
    - coin toss experiment
    - possible outcomes: S = {heads, tails}
  - Example
    - die toss experiment
    - possible outcomes: S = {1,2,3,4,5,6}

Introduction to Probability

- Definition of sample space depends on what we are asking
  - Sample Space (S): the set of all possible outcomes
  - Example
    - die toss experiment for whether the number is even or odd
    - possible outcomes: {even,odd}
    - *not* {1,2,3,4,5,6}

More definitions

- Events
  - an ***event*** is any subset of outcomes from the ***sample space***
- Example
  - die toss experiment
  - let A represent the event such that the outcome of the die toss experiment is divisible by 3
  - A = {3,6}
  - A is a subset of the sample space S= {1,2,3,4,5,6}

Definition of Probability

- The probability law assigns to an event a nonnegative number
- Called P(A)
- Also called the probability A
- That encodes our knowledge or belief about the collective likelihood of all the elements of A
- Probability law must satisfy certain properties

Probability Axioms

- Nonnegativity
  - P(A) >= 0, for every event A
- Additivity
  - If A and B are two disjoint events, then the probability of their union satisfies:
  - P(A U B) = P(A) + P(B)
- Normalization
  - The probability of the entire sample space S is equal to 1, i.e. P(S) = 1.

An example

- An experiment involving a single coin toss

  There are two possible outcomes, H and T

  Sample space S is {H,T}

  If coin is fair, should assign equal probabilities to {H,T} outcomes

  Since they have to sum to 1

  P({H}) = 0.5 and P({T}) = 0.5

  P({H,T}) = P({H})+P({T}) = 1.0

Another example

- Experiment involving 3 coin tosses
- Outcome is a 3-long string of H or T
- S ={HHH,HHT,HTH,HTT,THH,THT,TTH,TTT}
- Assume each outcome is equiprobable
  - “Uniform distribution”
- What is probability of the event that exactly 2 heads occur?

  A = {HHT,HTH,THH} 3 events

  P(A) = P({HHT})+P({HTH})+P({THH}) union of the prob of individual events

  = 1/8 + 1/8 + 1/8

  = 3/8

Probability definitions

- In summary:

$$P(E) = \frac{\text{number of outcomes corresponding to event E}}{\text{total number of outcomes}}$$

Probability of drawing a spade from 52 well-shuffled playing cards:

$$\frac{13}{52} = \frac{1}{4} = 0.25$$

Probability and part of speech tags

- What's the probability of a random word (from a random dictionary page) being a verb?

$$P(drawingaverb) = \frac{ofwaystogetaverb}{allwords}$$

- How to compute each of these?

  All words = just count all the words in the dictionary

  # of ways to get a verb: # of words which are verbs!

  If a dictionary has 50,000 entries, and 10,000 are verbs….

  P(V) is 10000/50000 = 1/5 = .20

Conditional Probability

- A way to reason about the outcome of an experiment based on partial information
  - In a word guessing game the first letter for the word is a “t”. What is the likelihood that the second letter is an “h”?
  - How likely is it that a person has a disease given that a medical test was negative?
  - A spot shows up on a radar screen. How likely is it that it corresponds to an aircraft?

An intuition

- Let’s say A is “it’s raining”.
- Let’s say P(A) in dry Florida is .01
- Let’s say B is “it was sunny ten minutes ago”
- P(A|B) means “what is the probability of it raining now if it was sunny 10 minutes ago”
- P(A|B) is probably way less than P(A)
- Perhaps P(A|B) is .0001
- Intuition: The knowledge about B should change our estimate of the probability of A.

More precisely

- Given an experiment, a corresponding sample space S, and a probability law
- Suppose we know that the outcome is some event B
- We want to quantify the likelihood that the outcome also belongs to some other event A
- We need a new probability law that gives us the conditional probability of A given B
- P(A|B)

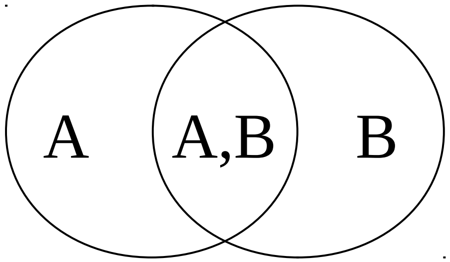
Conditional Probability

- let A and B be events in the sample space
- P(A|B) = the conditional *probability* of event A *occurring given* some fixed event B *occurring*
- *definition:* $P(A|B) = P(A \cap B) / P(B)$

Conditional probability

- $P(A|B) = P(A \cap B)/P(B)$
- Or

$$P(A|B) = \frac{P(A,B)}{P(B)}$$



Note: $P(A,B) = P(A|B) \cdot P(B)$

Also: $P(A,B) = P(B,A)$

Independence

- What is $P(A,B)$ if A and B are independent?
- $P(A,B)=P(A) \cdot P(B)$ iff A,B independent.

  $P(heads,tails) = P(heads) \cdot P(tails) = 0.5 \cdot 0.5 = 0.25$

  Note: $P(A|B)=P(A)$ iff A,B independent

  Also: $P(B|A)=P(B)$ iff A,B independent

Bayes Theorem

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

- Idea: The probability of an A conditional on another event B is generally different from the probability of B conditional on A. There is a definite relationship between the two.

Deriving Bayes Rule

The probability of event A given event B is

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

Deriving Bayes Rule

The probability of event B given event A is

$$P(B|A) = \frac{P(A,B)}{P(A)}$$

Deriving Bayes Rule

$$P(A|B)=\frac{P(A,B)}{P(B)} \qquad P(B|A)=\frac{P(A,B)}{P(A)}$$

$$P(A|B)P(B)=P(A,B) \qquad P(B|A)P(A)=P(A,B)$$

Deriving Bayes Rule

$$P(A|B)=\frac{P(A,B)}{P(B)} \qquad P(B|A)=\frac{P(A,B)}{P(A)}$$

$$P(A|B)P(B)=P(A,B) \qquad P(B|A)P(A)=P(A,B)$$

$$P(A|B)P(B)=P(B|A)P(A)$$

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

Bayes Rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

the theorem may be paraphrased as:

Conditional/Posterior probability =

(LIKELIHOOD multiplied by PRIOR) divided by NORMALIZING CONSTANT

Hidden Markov Model (HMM) Tagging

- Using an HMM to do POS tagging
- HMM is a special case of Bayesian inference
- It is also related to the "noisy channel" model in ASR (Automatic Speech Recognition)

POS tagging as a sequence classification task

- Given a sentence (an “observation” or “sequence of observations”)
  - *Secretariat is expected to race tomorrow*
  - sequence of n words w1…wn.
- What is the best sequence of tags which corresponds to this sequence of observations?
- Probabilistic/Bayesian 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 w1…wn.

Getting to HMM

- Let T = $t_1,t_2,\ldots,t_n$
- Let W = $w_1,w_2,\ldots,w_n$
- Goal: Out of all sequences of tags $t_1\ldots t_n$, get the the most probable sequence of POS tags T underlying the observed sequence of words $w_1,w_2,\ldots,w_n$

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

- Hat ^ means "our estimate of the best = the most probable tag sequence"
- Argmax$_x$ f(x) means "the x such that f(x) is maximized"

  it maximizes our estimate of the best tag sequence

Getting to HMM

- This equation is guaranteed to 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 do we make it operational? How do we compute this value?
- Intuition of Bayesian classification:
  - Use Bayes rule to transform it into a set of other probabilities that are easier to compute
  - Thomas Bayes: British mathematician (1702-1761)

Bayes Rule

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

Breaks down any conditional probability P(x|y) into three other probabilities

P(x|y): The conditional probability of an event x assuming that y has occurred

Bayes Rule

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

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

We can drop the denominator: it does not change for each tag sequence; we are looking for the best tag sequence for the same observation, for the same fixed set of words

Bayes Rule

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

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

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

Likelihood and prior

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

Likelihood and prior

Further Simplifications

1. the probability of a word appearing depends only on its own POS tag, i.e, independent of other words around it

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

2. **BIGRAM** assumption: the probability of a tag appearing depends only on the previous tag

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

3. The most probable tag sequence estimated by the bigram tagger

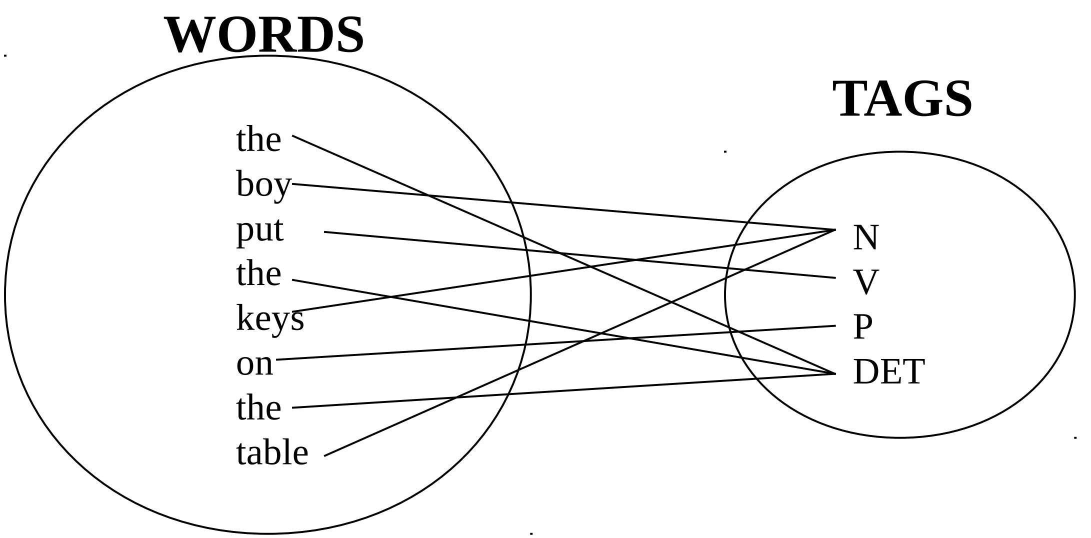
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

Likelihood ratio

Further Simplifications

1. the probability of a word appearing depends only on its own POS tag, i.e, independent of other words around it

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



Prior probability

Further Simplifications

2. **BIGRAM** assumption: the probability of a tag appearing depends only on the previous tag

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

Bigrams are groups of two written letters, two syllables, or two words; they are a special case of N-gram.

Bigrams are used as the basis for simple statistical analysis of text

The bigram assumption is related to the first-order Markov assumption

Likelihood and prior

Further Simplifications

3. The most probable tag sequence estimated by the bigram tagger

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

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

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

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

bigram assumption

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
    - But P(DT|JJ) to be:?

Two kinds of probabilities (1)

- Tag transition probabilities $p(t_i|t_{i-1})$
  - Compute P(NN|DT) by counting in a labeled corpus:

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

\# of times DT is followed by NN

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

Two kinds of probabilities (2)

- Word likelihood probabilities $p(w_i|t_i)$
  - P(is|VBZ) = probability of VBZ (3sg pres verb) being "is"

    If we were expecting a third person singular verb, how likely is it that this verb would be *is*?

  - Compute P(is|VBZ) by counting in a labeled corpus:

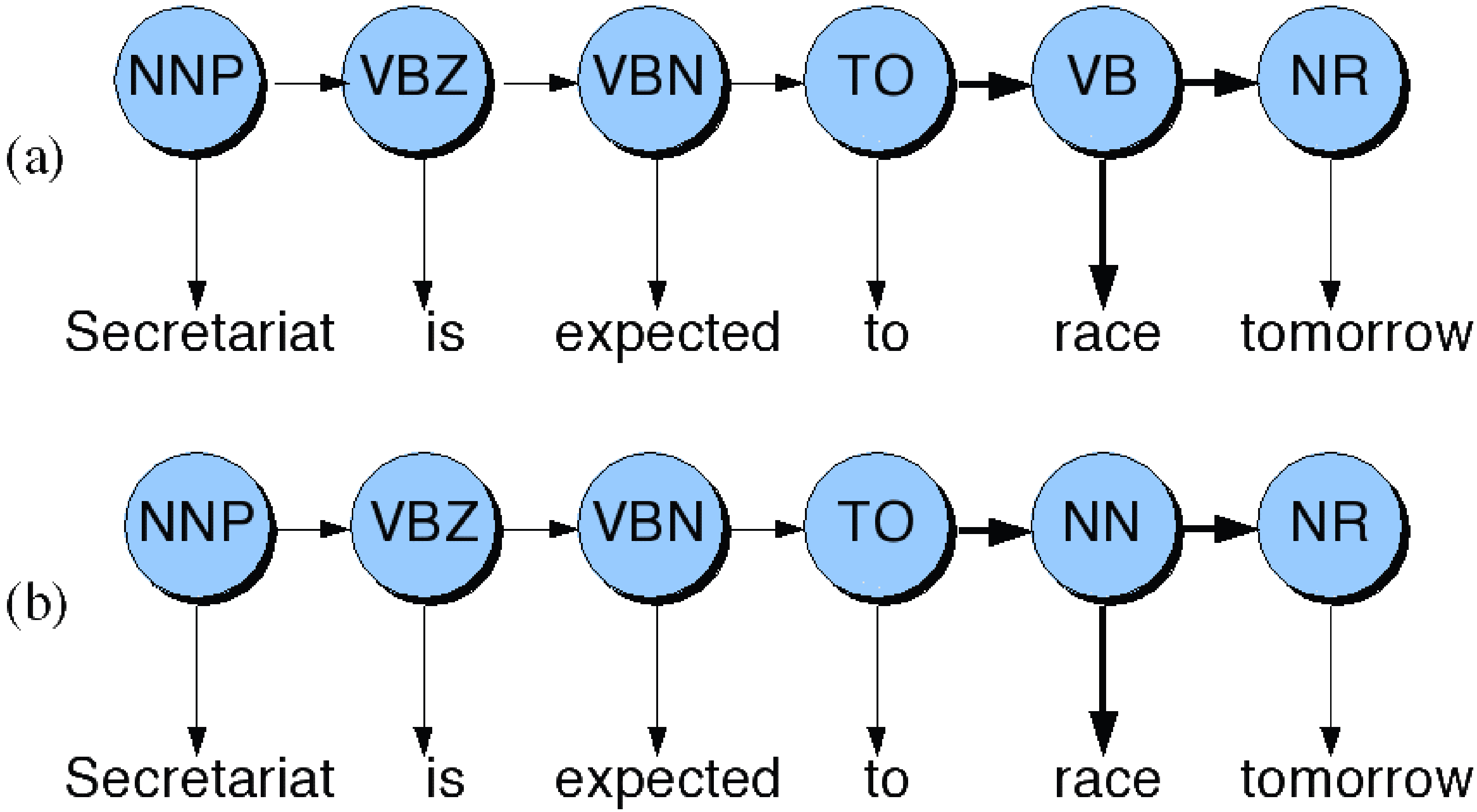
$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

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

An 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”

- P(NN|TO) = .00047

  P(VB|TO) = .83

  The tag transition probabilities P(NN|TO) and P(VB|TO):

  ‘How likely are we to expect verb/noun given the previous tag TO?’
- P(race|NN) = .00057

  P(race|VB) = .00012

  Lexical likelihoods from the Brown corpus for ‘race’ given a POS tag NN or VB.

Disambiguating “race”

- P(NR|VB) = .0027

  P(NR|NN) = .0012

  tag sequence probability of an adverb occurring given the previous

  tag verb / noun

P(VB|TO)P(NR|VB)P(race|VB) = .83 x .0027 x .00012 = .00000027

P(NN|TO)P(NR|NN)P(race|NN)= .

00047x.0012x.00057=.00000000032

Multiply the lexical likelihoods with the tag sequence probabilities:

Hence the *race* is tagged as **verb** !

Hidden Markov Models

- What we've described with these two kinds of probabilities is a Hidden Markov Model (HMM)
- Let's just spend a bit of time tying this into the model
- In order to define HMM, we will first introduce the Markov Chain, or observable Markov Model.

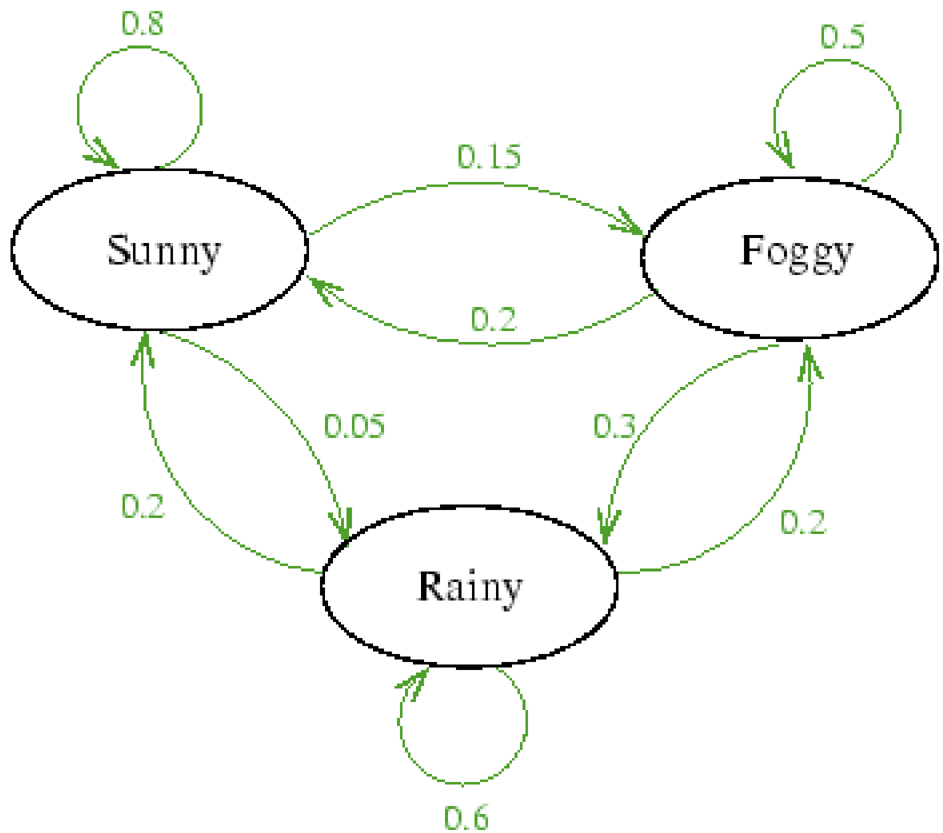
Definitions

- A weighted finite-state automaton adds probabilities to the arcs
  - The sum of the probabilities leaving any arc 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 inherently ambiguous problems
  - Useful for assigning probabilities to unambiguous sequences

Markov chain = “First-order observable Markov Model”

Table 1: Probabilities $p(q_{n+1}|q_n)$ of tomorrow's weather based on today's weather

	Tomorrow's weather		
Today's weather			
	0.8	0.05	0.15
	0.2	0.6	0.2
	0.2	0.3	0.5



Hidden Markov Model

- For Markov chains, the output symbols are the same as the states.
  - See **rainy** weather: we're in state **rainy**
- But in part-of-speech tagging (and other things)
  - The output symbols are **words**
  - But the hidden states are **part-of-speech tags**
- So we need an extension!
- A Hidden Markov Model is an extension of a Markov chain that allows both observed events (like words as input) and hidden events (like pos tags)

Hidden Markov Model

- 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 (prior)
  - Transition probability matrix $A = \{a_{ij}\}$. Probability of moving from state *i* to state *j*

- Observation likelihoods (likelihood)
  - probability matrix $B=\{b_i(o_t)\}$

    a sequence of observation likelihoods, each expressing the probability of an observation o_t being generated from a state *i*.

- A special start and end state

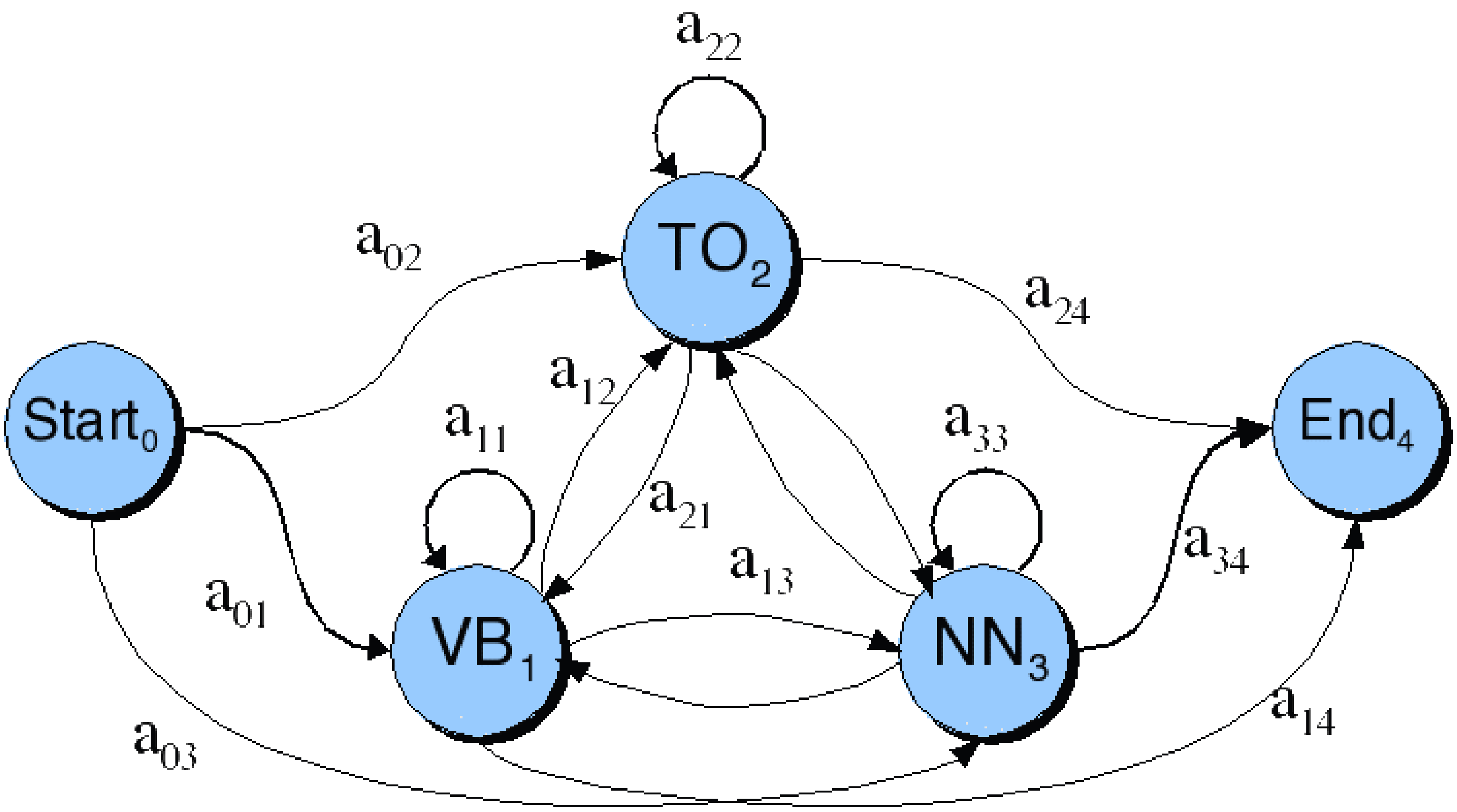
HMM Taggers

- An HMM has two kinds of probabilities
  - A transition probabilities (PRIOR) (slide 35)
  - B observation likelihoods (LIKELIHOOD) (slide 35)
- HMM Taggers choose the tag sequence which maximizes the product of word likelihood and tag sequence probability

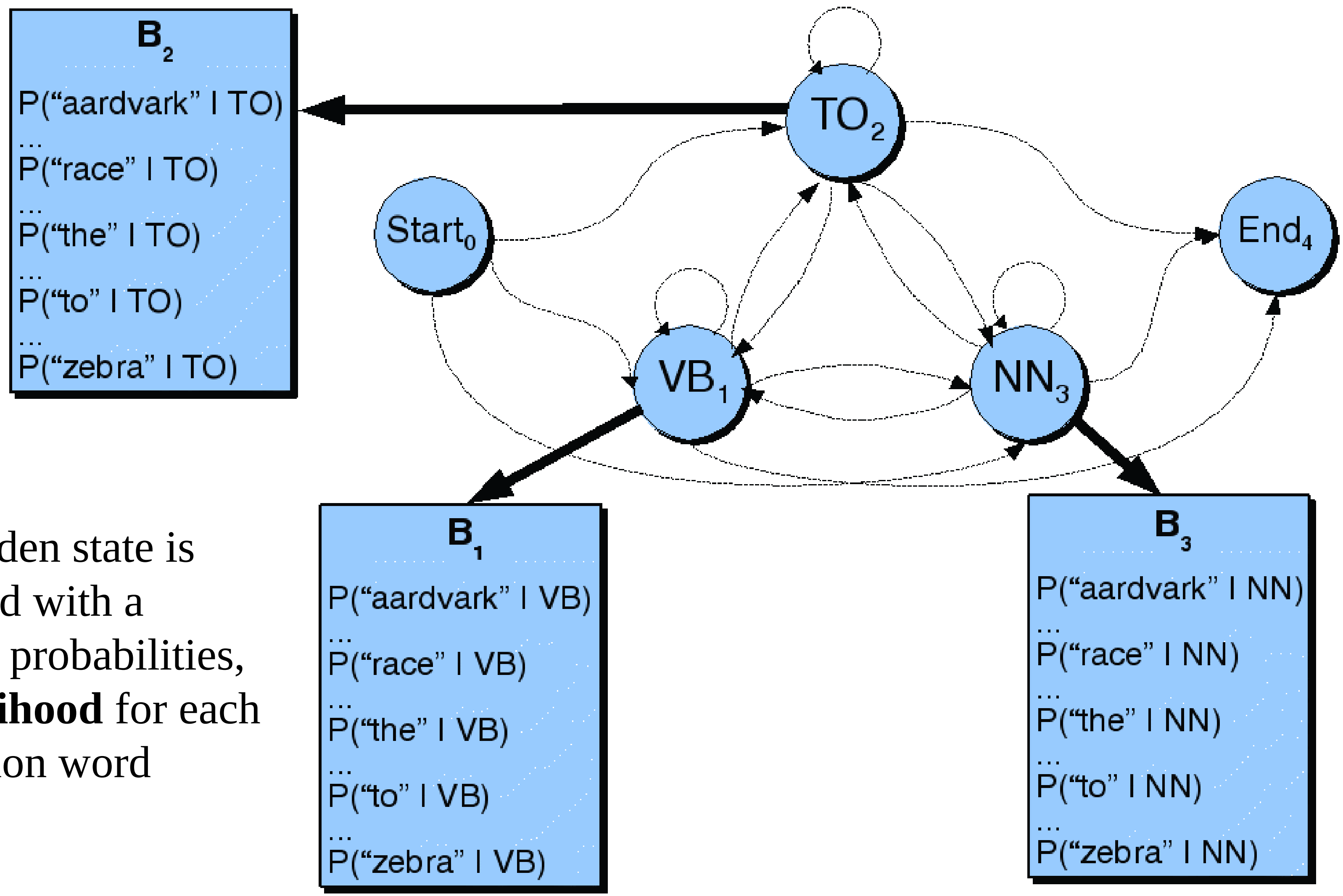
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \overbrace{P(w_1^n|t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

Markov chain corresponding to hidden states of HMM, showing A probs

Transition probabilities are used to compute **prior probability**



B observation likelihoods for HMM



Each hidden state is associated with a vector of probabilities, one **likelihood** for each observation word

Viterbi Algorithm

- Any model that contains hidden variables, the task of determining which sequence of variables is the underlying source of some sequence of observations is called the **decoding task**.
- The Viterbi algorithm is the most common decoding algorithm used for HMMs.
- Let HMM be defined by two tables: Fig 5.15, Fig 5.16

Viterbi Algorithm

- Fig 5.15 – the *transition probabilities* between hidden states (pos tags) – a_{ij} probability

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

Figure 5.15 Tag transition probabilities (the a array, $p(t_i|t_{i-1})$) computed from the 87-tag Brown corpus without smoothing. The rows are labeled with the conditioning event; thus $P(PPSS|VB)$ is .0070. The symbol <s> is the start-of-sentence symbol.

Viterbi Algorithm

- Fig 5.16 – the observation likelihoods of words given tags (i.e., $b_i(O_t)$ probabilities)

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Figure 5.16 Observation likelihoods (the *b* array) computed from the 87-tag Brown corpus without smoothing.

Viterbi Algorithm

- Viterbi sets up a probability matrix – one column for each observation t and one row for each state in the state graph.
- The algorithm first creates N state columns.
- Begin in first column and move on column by column.
- For each cell, *viterbi[s,t]* is computed by taking the maximum over the extensions of all the paths that lead to the current cell.

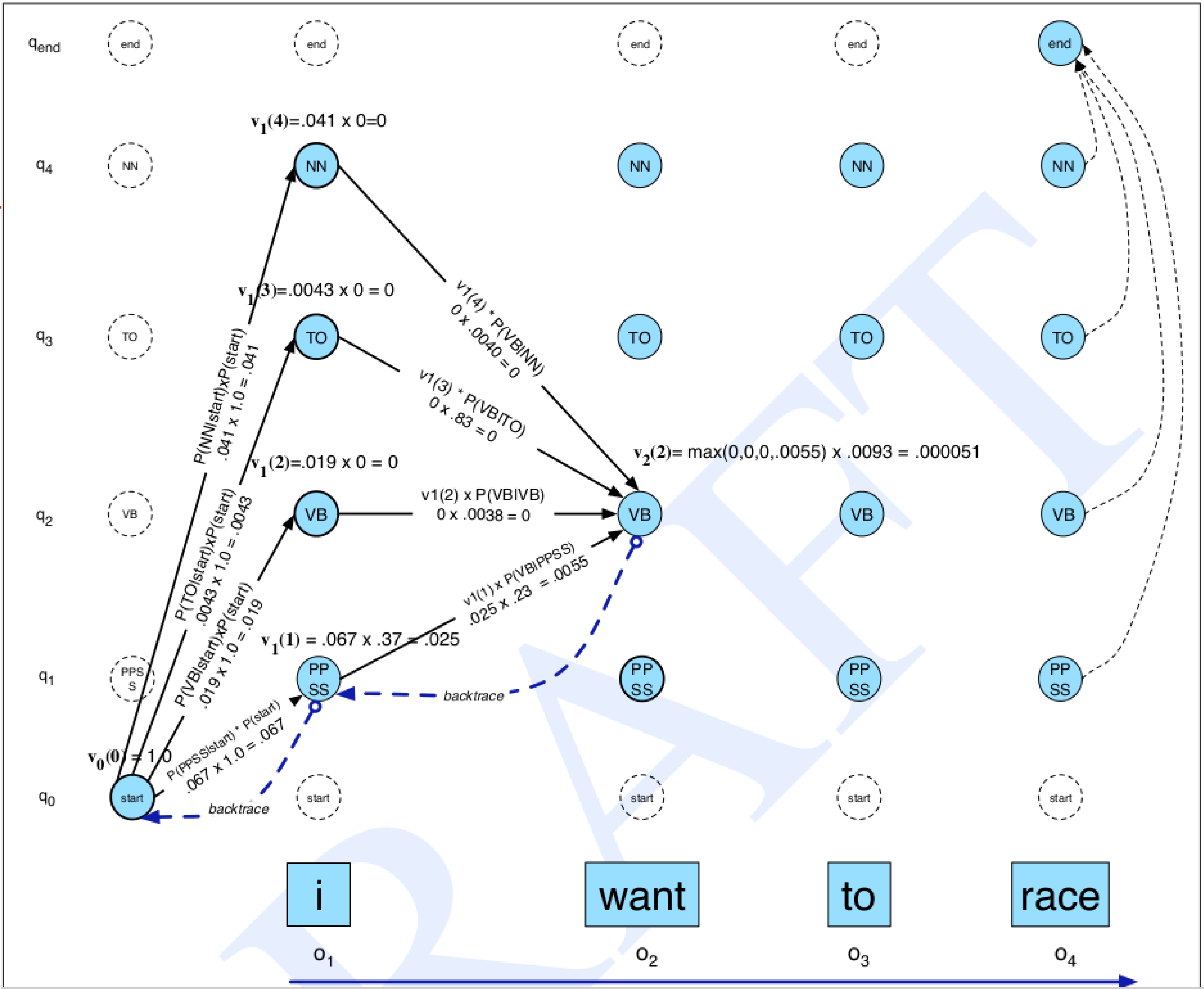
$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

Viterbi Algorithm

- For each cell, *viterbi[s,t]* is computed by taking the maximum over the extensions of all the paths that lead to the current cell.

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

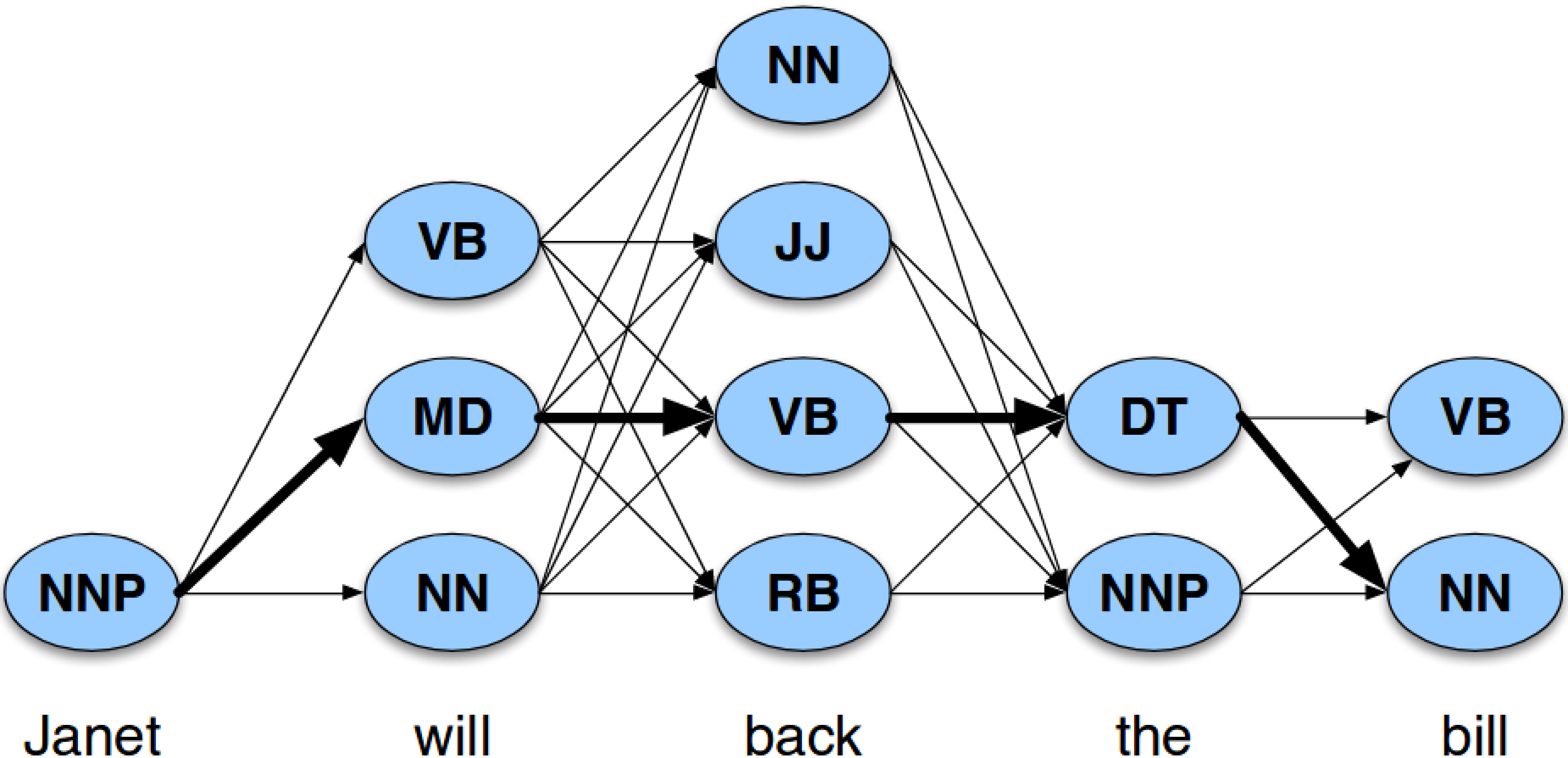
$v_{t-1}(i)$	the **previous Viterbi path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j



q_end
q4
q3
q2
q1
q0
end
NN
TO
VB
PPS S
start
v1(4)=.041 x 0=0
v1(3)=.0043 x 0 = 0
v1(2)=.019 x 0 = 0
v1(1) = .067 x .37 = .025
v0(0) = 1.0
P(NN|start)xP(start) .041 x 1.0 = .041
P(TO|start)xP(start) .0043 x 1.0 = .0043
P(VB|start)xP(start) .019 x 1.0 = .019
P(PPSS|start) * P(start) .067 x 1.0 = .067
v1(4) * P(VB|NN) 0 x .0040 = 0
v1(3) * P(VB|TO) 0 x .83 = 0
v1(2) x P(VB|VB) 0 x .0038 = 0
v1(1) x P(VB|PPSS) .025 x .23 = .0055
v2(2)= max(0,0,0,.0055) x .0093 = .000051
PP SS
backtrace
backtrace
i
want
to
race
o1
o2
o3
o4

Viterbi Algorithm

- Example 2: *Janet will back the bill*

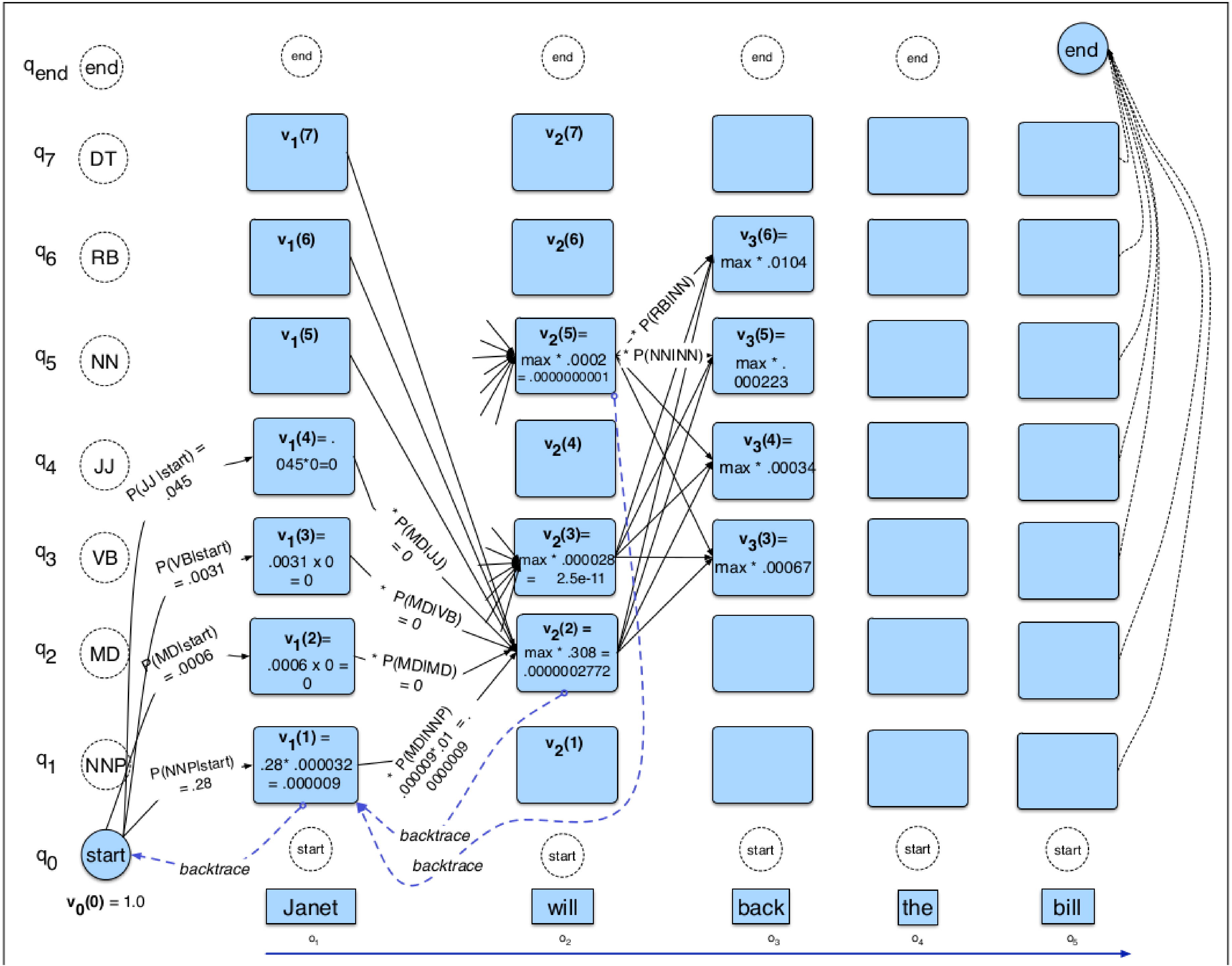


	NNP	MD	VB	JJ	NN	RB	DT
$< s >$	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

Figure 10.5 The A transition probabilities $P(t_i|t_{i-1})$ computed from the WSJ corpus without smoothing. Rows are labeled with the conditioning event; thus $P(VB|MD)$ is 0.7968.

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0.000097	0
NN	0	0.000200	0.000223	0.000006	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Figure 10.6 Observation likelihoods B computed from the WSJ corpus without smoothing.



q_end
end
q7
DT
q6
RB
q5
NN
q4
JJ
q3
VB
q2
MD
q1
NNP
q0
start
v0(0) = 1.0
v1(7)
v1(6)
v1(5)
v1(4)= .045*0=0
v1(3)= .0031 x 0 = 0
v1(2)= .0006 x 0 = 0
v1(1) = .28* .000032 = .000009
P(JJ |start) = .045
P(VB|start) = .0031
P(MD|start) = .0006
P(NNP|start) = .28
* P(MD|JJ) = 0
* P(MD|VB) = 0
* P(MD|MD) = 0
* P(MD|NNP) .000009*.01 = .0000009
v2(7)
v2(6)
v2(5)= max * .0002 = .0000000001
v2(4)
v2(3)= max * .000028 = 2.5e-11
v2(2) = max * .308 = .0000002772
v2(1)
* P(RB|NN)
* P(NN|NN)
v3(6)= max * .0104
v3(5)= max * .000223
v3(4)= max * .00034
v3(3)= max * .00067
backtrace
backtrace
backtrace
start
Janet
will
back
the
bill
o1
o2
o3
o4
o5

References

- Chapter 5. Part-of-Speech Tagging

  Speech and Language Processing by Jurafsky and Martin