Natural Language Processing

Tutorial 4: POS tagging and HMM

Dr. Sun Aixin

Question 1

- Find one tagging error in each of the following sentences that are tagged with the Penn Treebank tagset:

1. How/WRB do/VBP I/PRP get/VB to/TO Singapore/NN
2. Do/VBP you/PRP have/VB any/DT vacancies/NN
3. This/DT room/NN is/VBZ too/JJ noisy/JJ
4. Can/VB you/PRP give/VB me/PRP another/DT room/NN

Penn TreeBank POS Tagset

Review

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	"to"	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past participle	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your, one's*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	+,%, &	WRB	wh-adverb	*how, where*

Answer 1

- How/WRB do/VBP I/PRP get/VB to/TO Singapore/NN
  - **Singapore/NNP**
- Do/VBP you/PRP have/VB any/DT vacancies/NN
  - **vacancies/NNS**
- This/DT room/NN is/VBZ too/JJ noisy/JJ
  - **too/RB**
- Can/VB you/PRP give/VB me/PRP another/DT room/NN
  - **Can/MD**

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	"to"	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past participle	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your, one's*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	+,%, &	WRB	wh-adverb	*how, where*

Question 2

- Compute the best tag sequence for "I want to race" using the Viterbi algorithm with the provided HMM parameters, i.e., the transition probability and the word likelihood probabilities

	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

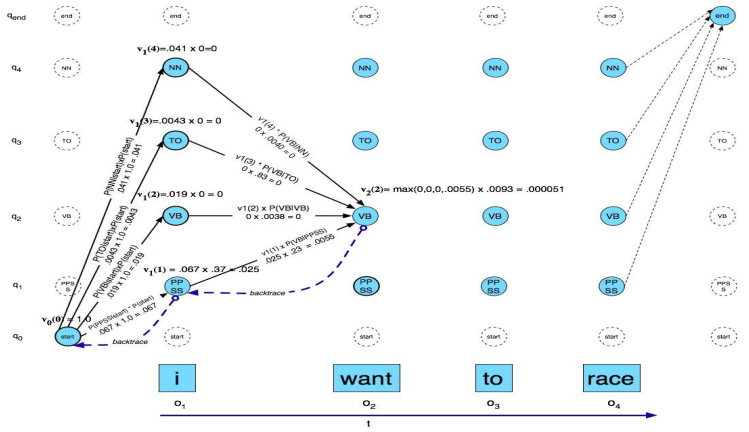
	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Main Idea

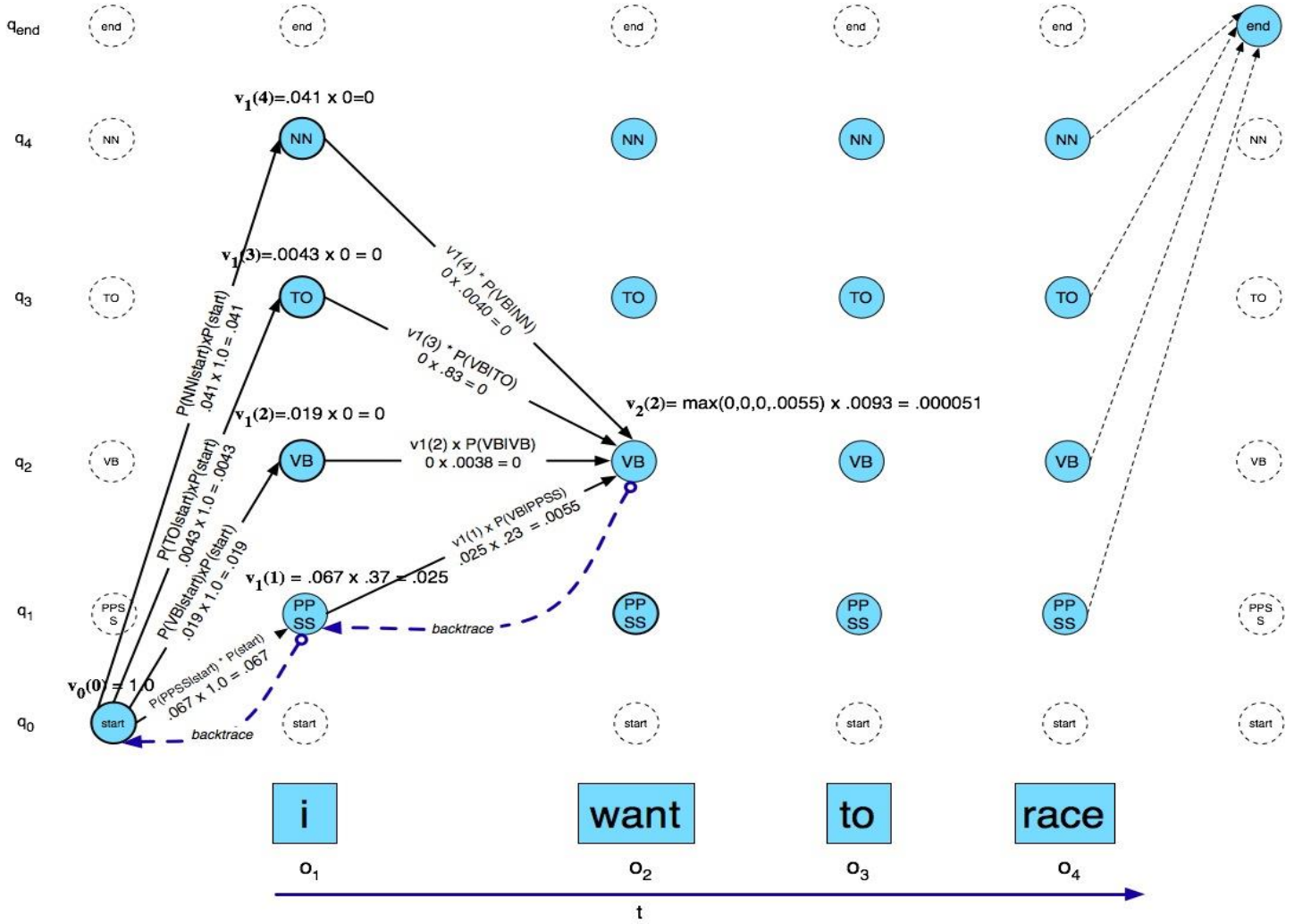
Review

- We also have a matrix.
  - Each column– a time 't' (observation)
  - Each row – a state 'i'
  - For each cell $v_t[i]$, we compute the probability of the **best path** to the cell

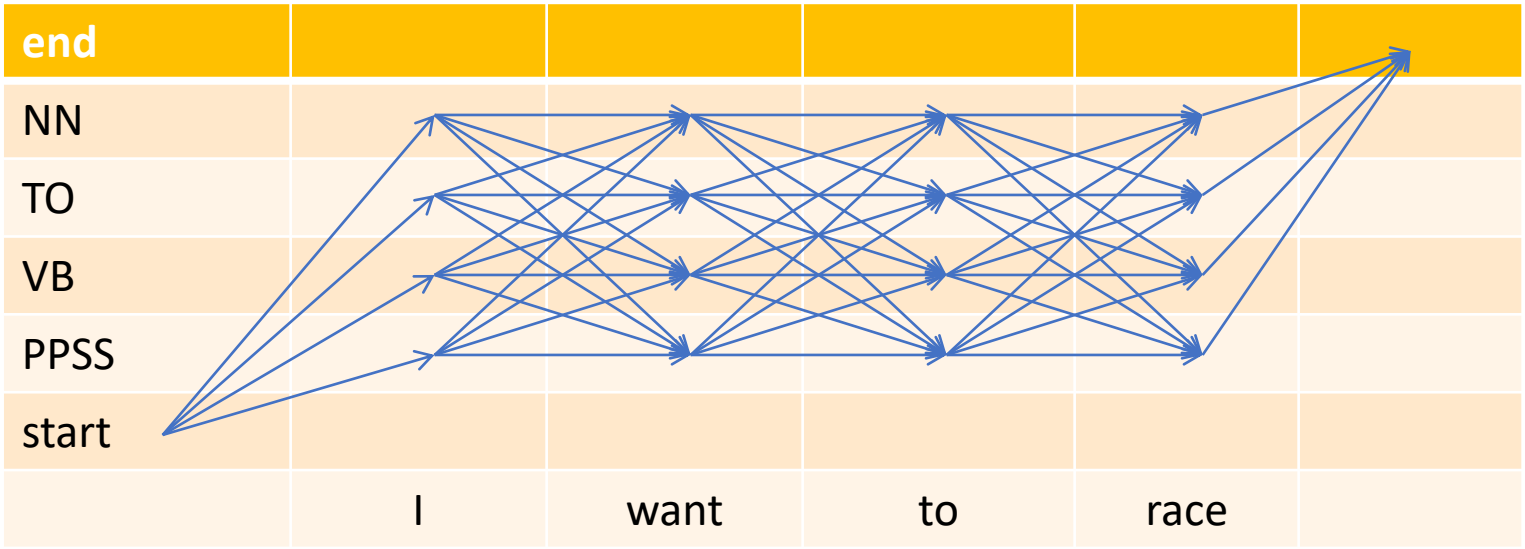
- the **Viterbi path probability** at time t for state i
  - there are $|Q|$ number of paths from $t-1$ to $v_t[i]$
  - if we know **the best path** to each cell in $t-1$, or $v_{t-1}[j]$
  - $\arg\max_j v_{t-1}[j] \times P(i|j) \times P(s_t|i)$



Viterbi Example



Required computations



(This figure does not show the backtrace pointers)

Answer 2

	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

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

					end
NN	$p(NN\| < s >) * p(I\|NN) = 0$				
TO	0				
VB	0				
PPSS	$p(PPSS\| < s >) * p(I\|PPSS) = 0.067 * 0.37 = 0.02479$				
start					
	I	want	to	race	

Answer 2

	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

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

					end
NN	0	$.02479 \times p(NN\|PPSS) * p(want\|NN) = .02479 \times .0012 \times .000054 = 0.00000000160639$			
TO	0	0			
VB	0	$.02479 \times p(VB\|PPSS) \times p(want\|VB) = .02479 \times .23 \times .0093 = 0.00005302581$			
PPSS	0.02479	0			
start					
	I	want	to	race	

Answer 2

	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

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

					end
NN	0	1.6×10^{-9}	0		
TO	0	0	$\max\left(1.6 \times 10^{-9} \times p(TO\|NN), 5.3 \times 10^{-5} \times p(TO\|VB)\right) * p(to\|TO) = \max(1.6 \times 10^{-9} \times .016, 5.3 \times 10^{-5} \times .035) * .99 = 1.84 \times 10^{-6}$		
VB	0	5.3×10^{-5}	0		
PPSS	0.02479	0	0		
start					
	I	want	to	race	

Answer 2

	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

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

					end
NN	0	1.6×10^{-9}	0	$1.84 \times 10^{-6} \times p(NN\|TO) \times p(race\|NN) = 1.84 \times 10^{-6} \times .00047 \times .00057 = 4.92 \times 10^{-14}$	
TO	0	0	1.84×10^{-6}	0	
VB	0	5.3×10^{-5}	0	$1.84 \times 10^{-6} \times p(VB\|TO) \times p(race\|VB) = 1.84 \times 10^{-6} \times .83 \times .00012 = 1.83 \times 10^{-10}$	
PPSS	0.02479	0	0	0	
start					
	I	want	to	race	

Answer 2

	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

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

					end
NN	0	1.6×10^{-9}	0	$1.84 \times 10^{-6} \times p(NN\|TO) \times p(race\|NN) = 1.84 \times 10^{-6} \times .00047 \times .00057 = 4.92 \times 10^{-14}$	
TO	0	0	1.84×10^{-6}	0	
VB	0	5.3×10^{-5}	0	$1.84 \times 10^{-6} \times p(VB\|TO) \times p(race\|VB) = 1.84 \times 10^{-6} \times .83 \times .00012 = 1.83 \times 10^{-10}$	
PPSS	0.02479	0	0	0	
start					
	I	want	to	race	

Answer 2

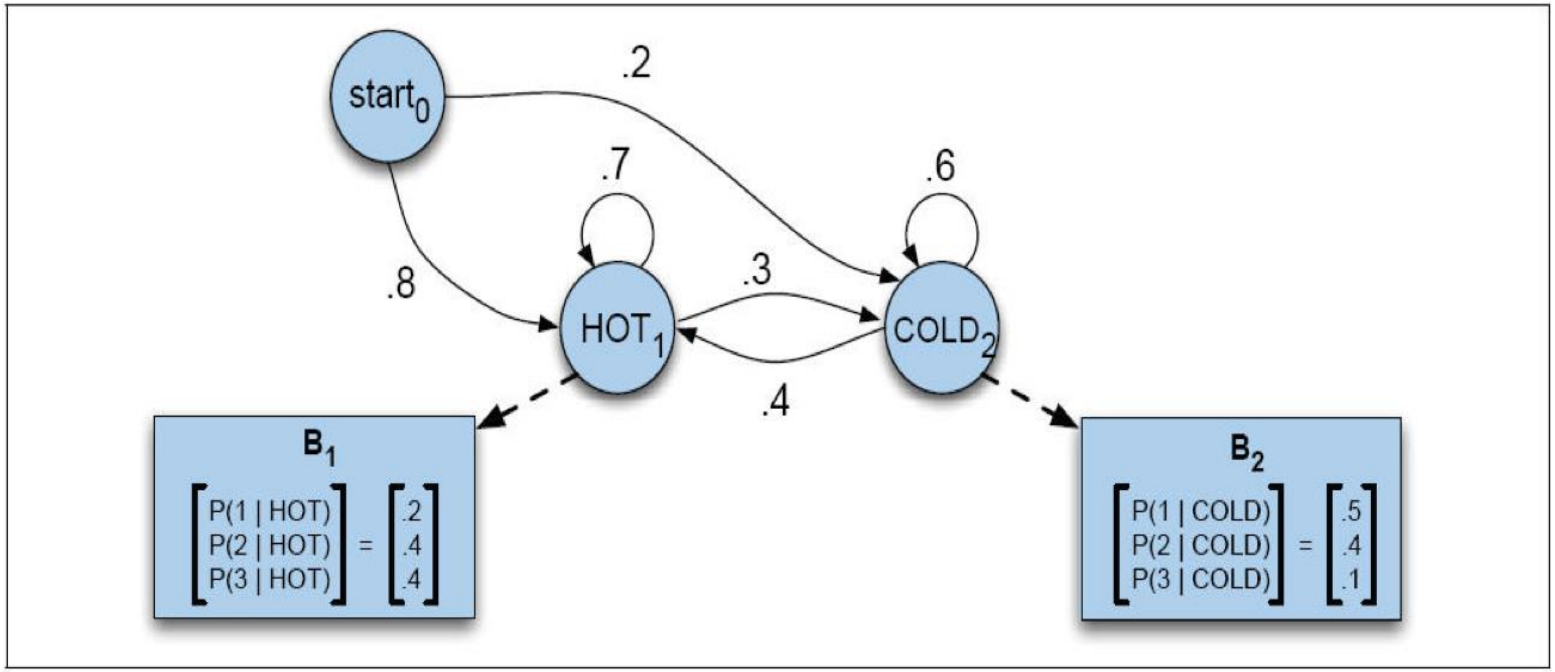
	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

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

					end
NN	0	1.6×10^{-9}	0	$1.84 \times 10^{-6} \times p(NN\|TO) \times p(race\|NN) = 1.84 \times 10^{-6} \times .00047 \times .00057 = 4.92 \times 10^{-14}$	
TO	0	0	1.84×10^{-6}	0	
VB	0	5.3×10^{-5}	0	$1.84 \times 10^{-6} \times p(VB\|TO) \times p(race\|VB) = 1.84 \times 10^{-6} \times .83 \times .00012 = 1.83 \times 10^{-10}$	
PPSS	0.02479	0	0	0	
start					
	I	want	to	race	

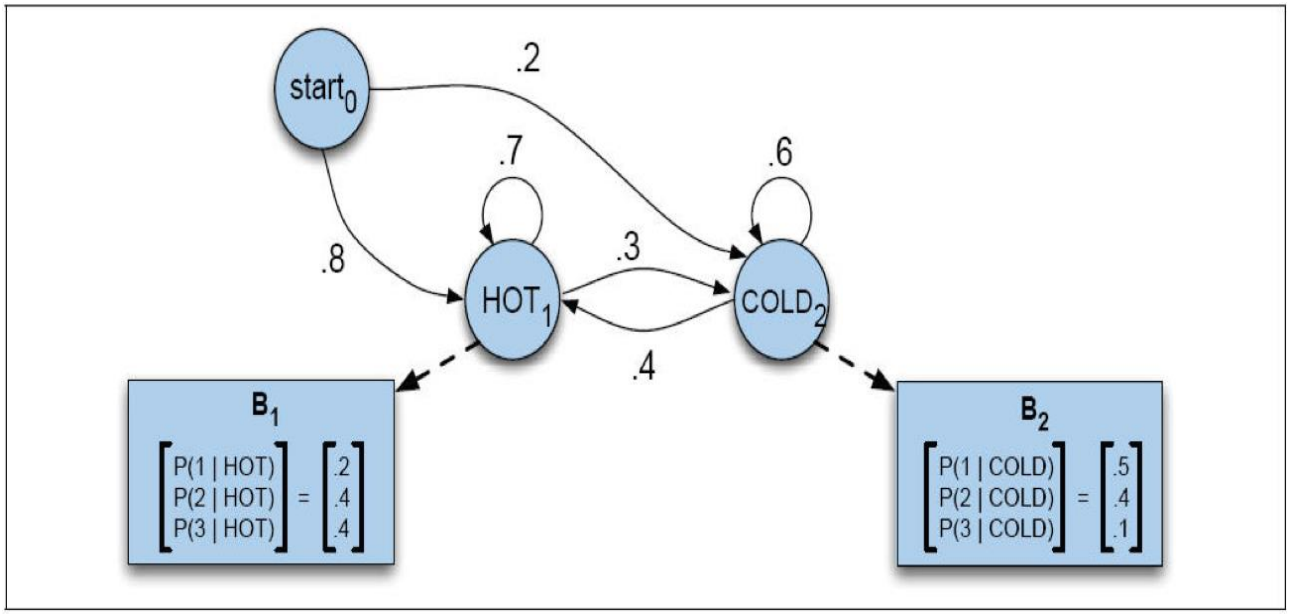
Question 3

- Run the Viterbi algorithm with the HMM below to compute the most likely weather sequences for each of the two observation sequences,
  - 312312312
  - 311233112.

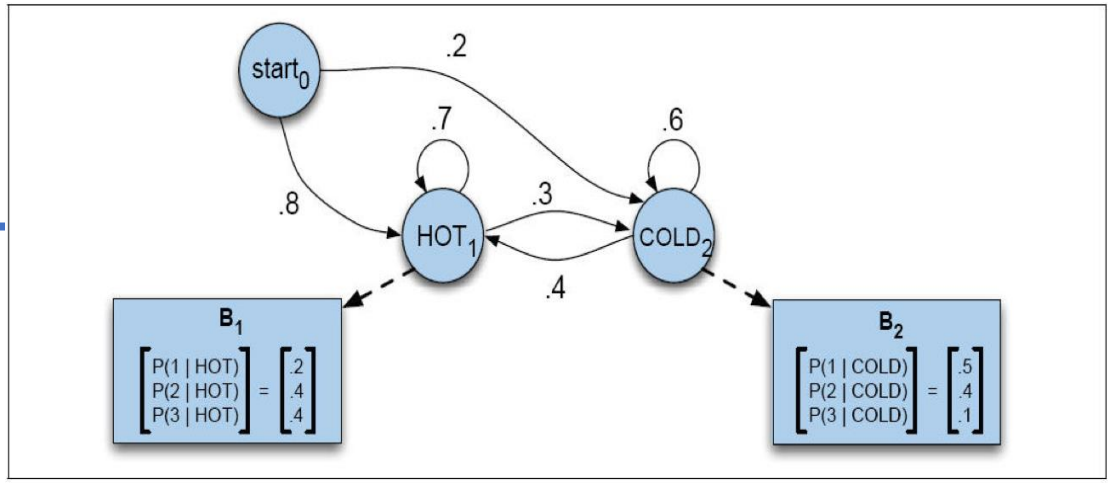


Hint 3

					end
H					
C					
start					
	3	1	2	3	…



Answer 3



- 3
  - H 0.8 *0.4 (P(3|H)) = 0.32
  - C 0.2 *0.1 (P (3|C)) = 0.02
- 1
  - H max (0.32*0.7*0.2, 0.02*0.4*0.2)
  - C max (0.32*0.3*0.5, 0.02*0.6*0.5)
- 2
- 3

Sequence 1: 3 1 2 3 1 2 3 1 2
Decoded states: -Hot--Hot--Hot--Hot--Hot--Hot--Hot--Hot--Hot-

Sequence 2: 3 1 1 2 3 3 1 1 2.
Decoded states: -Hot--Cold--Cold--Hot--Hot--Hot--Cold--Cold--Cold-

Question 4

- The Church tagger (1988) is different from the HMM tagger since it incorporates the probability of the tag given the word.
  - HMM: $p(word|tag) * p(tag|previous\ n\ tags)$
  - Church: $p(tag|word) * p(tag|previous\ n\ tags)$
- Interestingly, this use of a kind of "reverse likelihood" has proven to be useful in the modern log-linear approach to machine translation.

Question 4

- As a gedanken-experiment, construct a sentence, a set of tag transition probabilities, and a set of lexical tag probabilities that demonstrate a way in which the HMM tagger can produce a better answer than the Church tagger, and create another example in which the Church tagger is better.

  - Hint: The Church and HMM taggers will perform differently when, given two tags, tag_1 and tag_2:

    - $p(tag_1|word) > p(tag_2|word)$
    - $p(word|tag_1) < p(word|tag_2)$

Answer 4

- A word "manufacturing" is associated with the following probabilities (from a sample of text from Wall Street Journal).

  - $P(VBG|manufacturing) = 0.231$
  - $P(NN|manufacturing) = 0.769$
  - $P(manufacturing|VBG) = 0.004$
  - $P(manufacturing|NN) = 0.001$

  - So if we are looking at the words, we will expect this word to receive tag NN
  - If we are looking at the tags, we expect this word to be produced more often from VBG state than NN state

Answer 4

- Let's assume $P(NN| < s >) = P(VBG| < s >) = 0.5$
- Then HMM model will select VBG label
  - $P(manufacturing|NN) * P(NN| < s >) = 0.0005$
  - $P(manufacturing|VBG) * P(VBG| < s >) = 0.002$

- Church(1988) Tagger will select NN label
  - $P(NN|manufacturing) * P(NN| < s >) = 0.3845$
  - $P(VGB|manufacturing) * P(VBG| < s >) = 0.1155$