RICE SOCIAL SCIENCES
Department of Linguistics

hidden Markov model & Viterbi

LING 430 Computational Linguistics Fall 2026
lecture 9b
Oct 22 2026

Siyu Liang
siyu.liang@rice.edu

last time

- part-of-speech
  - open and closed classes
- tagsets
  - the universal tagset
  - the Penn Treebank tagset
- part-of-speech tagging

agenda

- Markov chains
- hidden Markov models
- the Viterbi algorithm
- HMM tagging

today's notebook

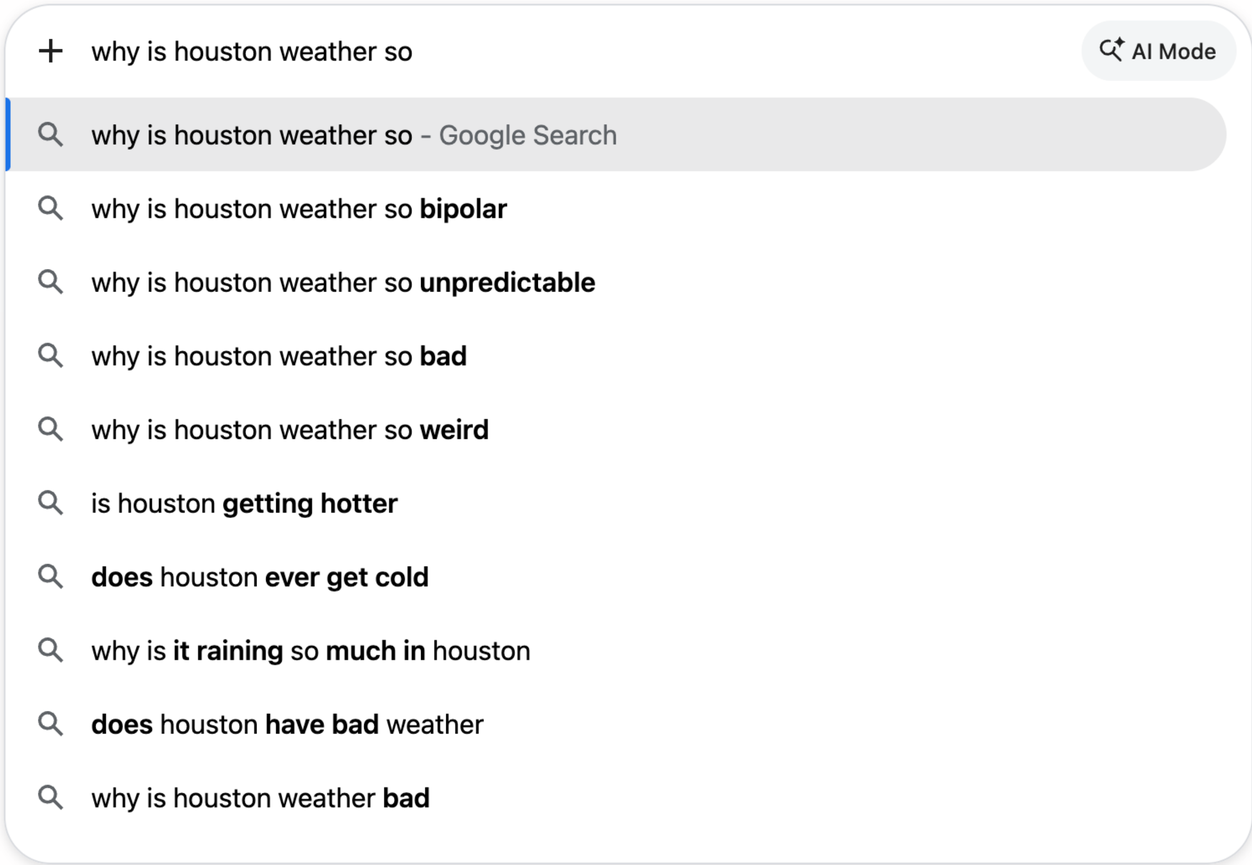
class09b.ipynb

- course website → week 9 → activity
  - **File → Save a copy in Drive**

Markov chains

Markov chains

- if today is hot
  - what about tomorrow?
  - hot, cold or warm?

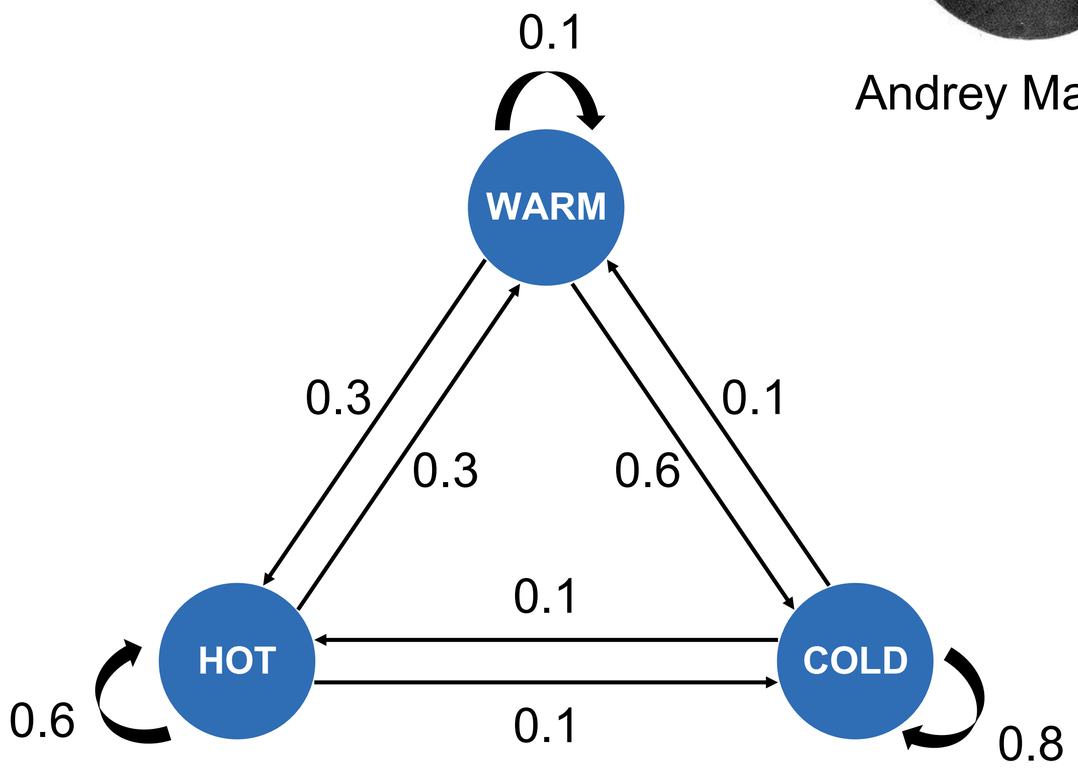


Markov chains



Andrey Markov

- a **Markov chain** is a set of states and the probability of moving from each state to the next
  - three states here: from HOT →
    - HOT 0.6
    - WARM 0.3
    - COLD 0.1
- the arrows leaving a state sum to 1



Markov chains

the Markov assumption

- tomorrow depends on today only, not on the whole history

$$P(q_i = a | q_1 \dots q_{i-1}) = P(q_i = a | q_{i-1})$$

- q_i is the state at step i
- a bigram language model is a Markov chain whose states are words

Markov chains

- a Markov chain is fully specified by two tables
- a set of states $Q = q_1 \ldots q_N$
- an **initial distribution** π: the probability of each state at step 1
- a **transition matrix** A: a_{ij} is the probability of going from state i to state j

π	HOT	COLD	WARM
	0.5	0.2	0.3

A	HOT	COLD	WARM
HOT	0.6	0.1	0.3
COLD	0.1	0.8	0.1
WARM	0.3	0.6	0.1

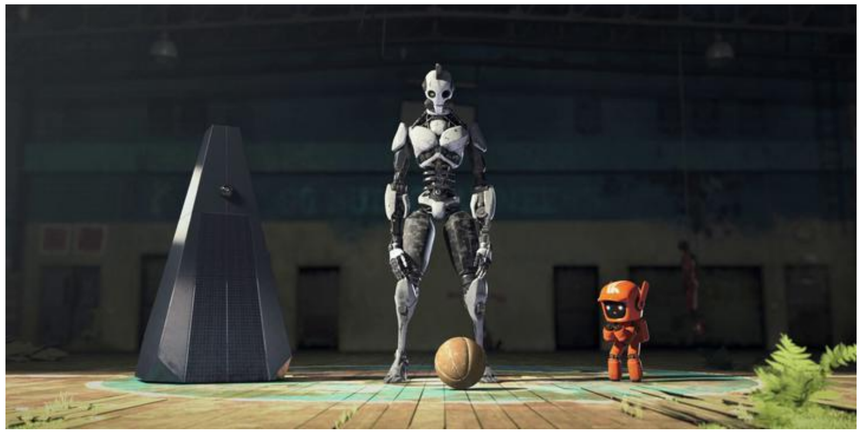
Markov chains

- with the table we can score any sequence of states
- starting from HOT, which is more likely
  - HOT HOT COLD or HOT COLD COLD?
- multiply along the arrows
  $P(\text{HOT COLD}|\text{HOT}) = a_{\text{HH}} \cdot a_{\text{HC}} = 0.6 \times 0.1 = 0.06$
  $P(\text{COLD COLD}|\text{HOT}) = a_{\text{HC}} \cdot a_{\text{CC}} = 0.1 \times 0.8 = 0.08$
- i.e. cold days tend to stay cold

A	HOT	COLD	WARM
HOT	0.6	0.1	0.3
COLD	0.1	0.8	0.1
WARM	0.3	0.6	0.1

hidden Markov models

hidden Markov models



Love Death + Robots (2019)

- in a Markov chain we see the states
- now suppose we cannot
- suppose something catastrophic happens to humanity…
- you are a climatologist in the year 2799, and you want to know what was the weather in Houston in 2026?
- no weather records survive, but we have a diary from a human
  - I ate 3 ice cream today… I ate 1 ice cream today… I ate 3 ice cream today.
- now we have to figure out: which days were hot and which were cold (no warm for simplicity)?

https://www.imdb.com/title/tt9788484/

hidden Markov models

hidden states

- a hidden Markov model is a Markov chain whose states we cannot see, only observations that depend on them
- the states are **hidden**: we never see HOT or COLD
- we see **observations** that depend on the state: 1, 2 or 3 ice creams
- so we need one more table
  - π: the initial distribution
  - A: transitions between hidden states
  - B: **emission probabilities**, $b_i(o) = P(o \mid q_i)$, how likely each observation is in each state
    - e.g. *P*(🍦🍦🍦| HOT)

hidden Markov models

the ice cream HMM

- π initial distribution
- table A transition probabilities
- table B emission probabilities
  - on a hot day one eats three ice creams with probability 0.4, on a cold day 0.1

π	**HOT**	**COLD**
	0.8	0.2

A	**HOT**	**COLD**
HOT	0.6	0.4
COLD	0.5	0.5

B	🍦	🍦🍦	🍦🍦🍦
HOT	0.2	0.4	0.4
COLD	0.5	0.4	0.1

hidden Markov models

two assumptions

- the real world would be too difficult to model without some assumptions for simplification
- the **Markov assumption**: the next state depends only on the current state

$$P(q_i|q_1 \dots q_{i-1}) = P(q_i|q_{i-1})$$

- **output independence**: an observation depends only on the state that produced it

$$P(o_i|q_1 \dots q_T, o_1 \dots o_T) = P(o_i|q_i)$$

hidden Markov models

- observations O = 3 1 3, and one guess at the hidden states, Q = HOT COLD HOT
- walk the path: start, emit, move, emit, move, emit

$$P(Q,O) = \pi_H \cdot b_H(3) \cdot a_{\mathrm{HC}} \cdot b_C(1) \cdot a_{\mathrm{CH}} \cdot b_H(3)$$
$$= 0.8{\times}0.4{\times}0.4{\times}0.5{\times}0.5{\times}0.4 = 0.0128$$

- but there are 2^3 = 8 possible paths for three days
  - which is the best?

hidden Markov models

decoding

- **decoding**: given the observations, find the most probable sequence of hidden states

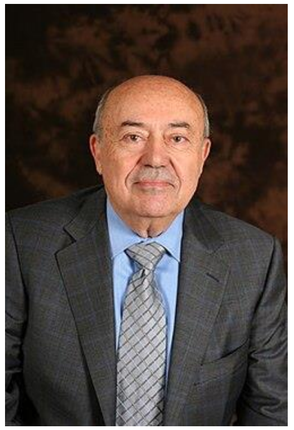
$$\hat{Q} = \text{argmax}_Q P(Q|O) = \text{argmax}_Q P(O|Q) \cdot P(Q)$$

- Bayes' theorem applied again!
  - and $P(O)$ is the same for every Q, so we dropped it
- but the paths to find the most probably sequence can be many
  - 2 states and 3 days is 8 paths; 45 tags and a 20-word sentence is 45^{20}, about 10^{33}!
- we don't want to do this for natural language data!

the Viterbi algorithm

the Viterbi algorithm

- the **Viterbi algorithm** finds the most probable path without trying every path
  - for each state, keep only the **best path** that ends there, and remember where it came from
- why is that safe?
  - if the overall best path is COLD on day 2, its first two days must be the best way to reach COLD on day 2
  - any other way to reach COLD on day 2 could be swapped in for a better total
- dynamic programming (also minimum edit distance)



Andrew J. Viterbi

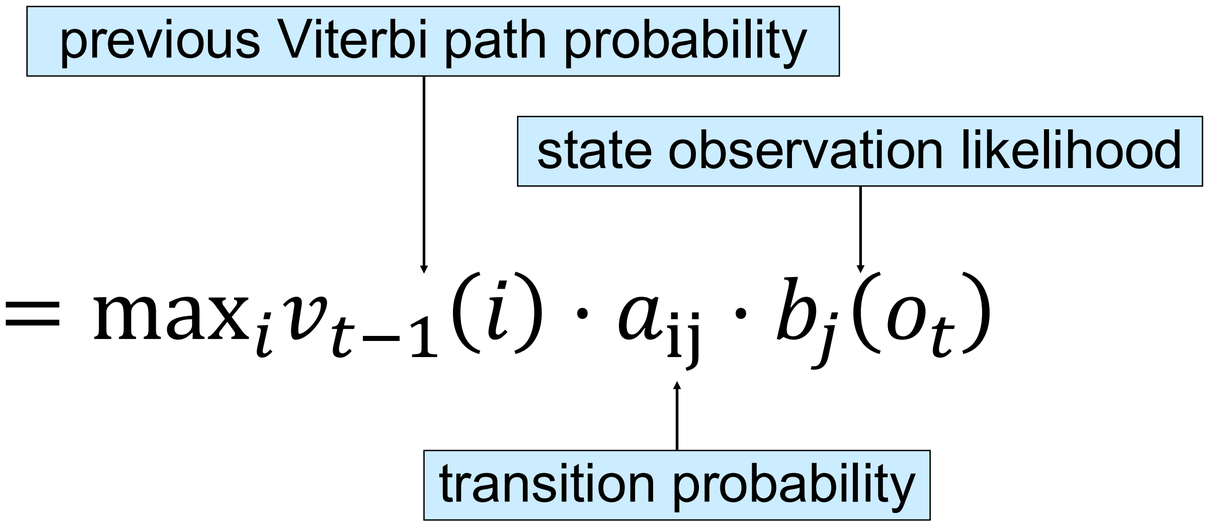
the Viterbi algorithm

notations

- $v_t(j)$: the probability of the best path that ends in state j at time t
- day 1: start in j and emit the first observation

$$v_1(j) = \pi_j \cdot b_j(o_1)$$

- later days: the best way into some state i yesterday, then move to j, then emit



the Viterbi algorithm

1. day 1, three ice creams 🍦🍦🍦

$$v_1(\text{HOT}) = \pi_H \cdot b_H(3) = 0.8 \times 0.4 = 0.32$$
$$v_1(\text{COLD}) = \pi_C \cdot b_C(3) = 0.2 \times 0.1 = 0.02$$

	🍦🍦🍦		
HOT	0.32		
COLD	0.02		

π	HOT	COLD
	0.8	0.2

A	HOT	COLD
HOT	0.6	0.4
COLD	0.5	0.5

B	🍦	🍦🍦	🍦🍦🍦
HOT	0.2	0.4	0.4
COLD	0.5	0.4	0.1

the Viterbi algorithm

2. day 2, one ice cream 🍦

- two ways into HOT
  - from HOT 0.32 × 0.6 = 0.192 👈 keep this
  - from COLD 0.02 × 0.5 = 0.01

  $v_2(\text{HOT}) = 0.192 \times b_H(1) = 0.192 \times 0.2 = 0.0384$
- two ways into COLD
  - from HOT 0.32 × 0.4 = 0.128 👈 keep this
  - from COLD 0.02 × 0.5 = 0.01

  $v_2(\text{COLD}) = 0.128 \times b_C(1) = 0.128 \times 0.5 = 0.064$

	🍦🍦🍦	🍦
HOT	0.32	0.0384 ← HOT
COLD	0.02	0.064 ← HOT

A	**HOT**	**COLD**
HOT	0.6	0.4
COLD	0.5	0.5

B	🍦	🍦🍦	🍦🍦🍦
HOT	0.2	0.4	0.4
COLD	0.5	0.4	0.1

the Viterbi algorithm

3. day 3, three ice creams

- into HOT
  - from HOT 0.0384 × 0.6 = 0.0230
  - from COLD 0.064 × 0.5 = 0.032 👈 keep this

$v_3(\text{HOT}) = 0.032 \times b_H(3) = 0.032 \times 0.4 = 0.0128$

- into COLD
  - from HOT 0.0384 × 0.4 = 0.0154
  - from COLD 0.064 × 0.5 = 0.032 👈 keep this

$v_3(\text{COLD}) = 0.032 \times b_C(3) = 0.032 \times 0.1 = 0.0032$

	🍦🍦🍦	🍦	🍦🍦🍦
HOT	0.32	0.0384 ← HOT	0.0128 ← COLD
COLD	0.02	0.064 ← HOT	0.0032 ← COLD

A	HOT	COLD
HOT	0.6	0.4
COLD	0.5	0.5

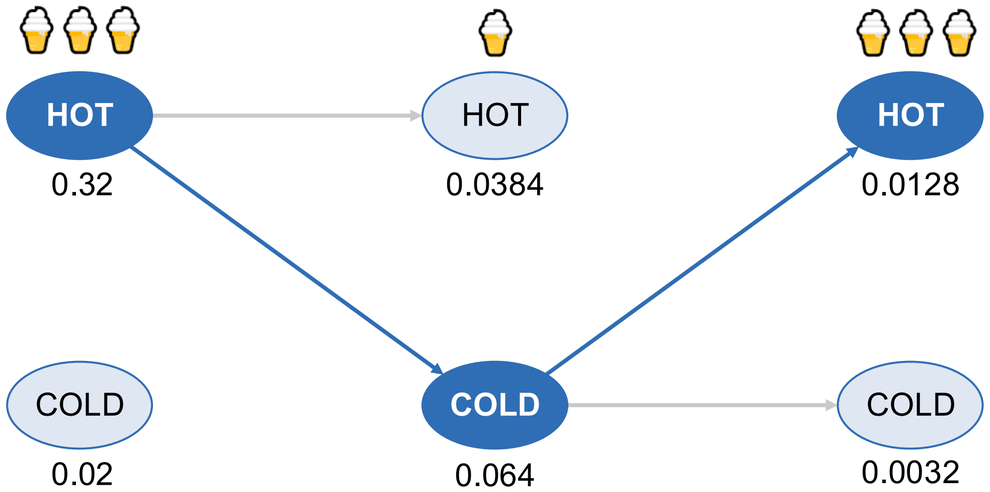
B	🍦	🍦🍦	🍦🍦🍦
HOT	0.2	0.4	0.4
COLD	0.5	0.4	0.1

the Viterbi algorithm

4. read off the answer

- the best final cell is HOT, 0.0128
- follow the arrows back
  - HOT on day 3 came from COLD, COLD on day 2 came from HOT
  - 3 1 3 was most likely HOT COLD HOT

	🍦🍦🍦	🍦	🍦🍦🍦
HOT	0.32	0.0384 ← HOT	0.0128 ← COLD
COLD	0.02	0.064 ← HOT	0.0032 ← COLD



activity 1

- we have a new diary ice cream entry!
  - 1 2 1
- in pairs, fill the table one day at a time
  - 1. day 1: π × emission
  - 2. and 3.: for each state, the better of the two ways in, × emission; note where it came from
  - 4. pick the best final cell and follow the arrows back
- what were the most likely weather for the three days?

π	HOT	COLD
	0.8	0.2

A	HOT	COLD
HOT	0.6	0.4
COLD	0.5	0.5

B	🍦	🍦🍦	🍦🍦🍦
HOT	0.2	0.4	0.4
COLD	0.5	0.4	0.1

HMM tagging

HMM tagging

- similar to the ice cream example
  - the words are what we observe (“ice cream”)
  - the POS tags are the hidden states that produced them (“NN NN”)
- A: $P(t_i \mid t_{i-1})$, how tags follow each other: the grammar
- B: $P(w_i \mid t_i)$, which words each tag produces: the lexicon
- insights from Tuesday
  - nouns usually follow a determiner and precede a verb

HMM tagging

transition & emission probabilities

- count in a tagged corpus, exactly as for a bigram model

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

- in the Wall Street Journal, the tag MD appears 13,124 times
  - 10,471 of those MD tags are followed by VB

$$P(\text{VB}|\text{MD}) = \frac{10471}{13124} = 0.80$$

- and 4,046 of those 13,124 MD tags are the word *will*

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

$$P(\text{will}|\text{MD}) = \frac{4046}{13124} = 0.31$$

HMM tagging

transition & emission probabilities

- we want the tag sequence that is most probable given the words
  $\hat{t}_{1:n} = \text{argmax}_{t_{1:n}} P(t_{1:n}|w_{1:n})$
- Bayes' theorem, drop $P(w_{1:n})$, then the two HMM assumptions
  $\hat{t}_{1:n} = \text{argmax}_{t_{1:n}} \prod_i P(w_i|t_i) \cdot P(t_i|t_{i-1})$
- get "emission" × "transition", then use Viterbi to find the best path

HMM tagging

example

- now let's revisit tagging for the sentence "Janet will back the bill"
- table A **transition probabilities** from actual WSJ data

	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

HMM tagging

example

- table B **emission probabilities**
  - blank cells are 0: the tag never produced that word

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

HMM tagging

1. Janet will back the bill

- only NNP can emit *Janet*, so every other cell is 0
  $v_1(\text{NNP}) = P(\text{NNP}|<s>) \cdot P(\text{Janet}|\text{NNP}) = 0.2767 \times 0.000032 = 8.85 \times 10^{-6}$

	Janet	will	back	the	bill
NNP	8.85×10^{-6}				
MD					
VB					
JJ					
NN					
RB					
DT					

A	from <s>
NNP	0.2767

B	Janet
NNP	0.000032

HMM tagging

2. Janet will back the bill

- three tags can emit *will*; the only way in is from NNP
  $v_2(\text{MD}) = 8.85\times10^{-6}\times0.0110\times0.308431 = 3.00\times10^{-8}$
  $v_2(\text{VB}) = 8.85\times10^{-6}\times0.0009\times0.000028 = 2.23\times10^{-13}$
  $v_2(\text{NN}) = 8.85\times10^{-6}\times0.0584\times0.000200 = 1.03\times10^{-10}$
- MD leads by a lot: *will* is almost always a modal

A	from NNP
MD	0.0110
VB	0.0009
NN	0.0584

B	will
MD	0.308431
VB	0.000028
NN	0.000200

	Janet	will	back	the	bill
NNP	8.85×10^{-6}				
MD		3.00×10^{-8} ← NNP			
VB		2.23×10^{-13} ← NNP			
JJ					
NN		1.03×10^{-10} ← NNP			
RB					
DT					

HMM tagging

3. Janet will back the bill

- four tags can emit *back*; the best way in is from MD
  $v_3(\text{VB}) = 3.00\times10^{-8}\times0.7968\times0.000672 = 1.61\times10^{-11}$
  $v_3(\text{RB}) = 3.00\times10^{-8}\times0.1698\times0.010446 = 5.33\times10^{-11}$
  $v_3(\text{JJ}) = 3.00\times10^{-8}\times0.0005\times0.000340 = 5.11\times10^{-15}$
  $v_3(\text{NN}) = 3.00\times10^{-8}\times0.0008\times0.000223 = 5.36\times10^{-15}$
- RB is ahead! *back* is usually an adverb

	Janet	will	back	the	bill
NNP	8.85×10^{-6}				
MD		$3.00 \times 10^{-8} \leftarrow$ NNP			
VB		$2.23 \times 10^{-13} \leftarrow$ NNP	$1.61 \times 10^{-11} \leftarrow$ MD		
JJ			$5.11 \times 10^{-15} \leftarrow$ MD		
NN		$1.03 \times 10^{-10} \leftarrow$ NNP	$5.36 \times 10^{-15} \leftarrow$ MD		
RB			$5.33 \times 10^{-11} \leftarrow$ MD		
DT					

A	from MD
VB	0.7968
RB	0.1698
JJ	0.0005
NN	0.0008

B	back
VB	0.000672
RB	0.010446
JJ	0.000340
NN	0.000223

HMM tagging

4. Janet will back the bill

A	from VB	from RB
DT	0.2231	0.0479
NNP	0.0322	0.0068

B	the
DT	0.506099
NNP	0.000048

- DT can come from VB or from RB; compare the two
  - from VB: $1.61\times10^{-11}\times0.2231\times0.506099 = 1.82\times10^{-12}$
  - from RB: $5.33\times10^{-11}\times0.0479\times0.506099 = 1.29\times10^{-12}$
- VB wins: determiners follow verbs far more than adverbs
- NNP can emit *the* too, but 2.49 × 10^{-17} is far behind

	Janet	will	back	the	bill
NNP	8.85 × 10^{-6}			2.49 × 10^{-17} ← VB	
MD		3.00 × 10^{-8} ← NNP			
VB		2.23 × 10^{-13} ← NNP	1.61 × 10^{-11} ← MD		
JJ			5.11 × 10^{-15} ← MD		
NN		1.03 × 10^{-10} ← NNP	5.36 × 10^{-15} ← MD		
RB			5.33 × 10^{-11} ← MD		
DT				1.82 × 10^{-12} ← VB	

HMM tagging

5. Janet will back the bill

$$v_5(\text{NN}) = 1.82\times10^{-12}\times0.4744\times0.002337 = 2.01\times10^{-15}$$
$$v_5(\text{VB}) = 1.82\times10^{-12}\times0.0002\times0.000028 = 1.02\times10^{-20}$$

- NN wins; follow the arrows back
- *Janet*/NNP *will*/MD *back*/VB *the*/DT *bill*/NN

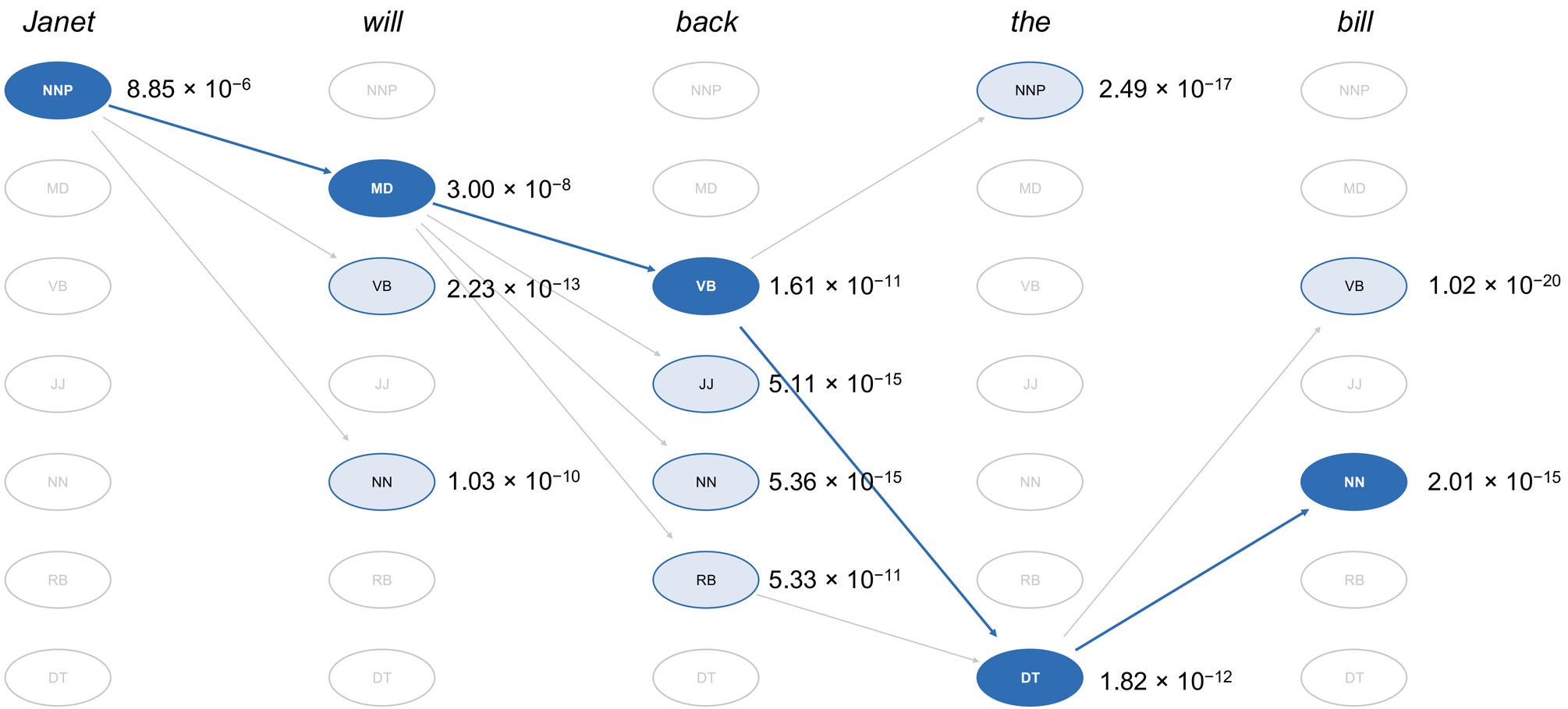
A	from DT
NN	0.4744
VB	0.0002

B	bill
NN	0.002337
VB	0.000028

	Janet	will	back	the	bill
NNP	8.85×10^{-6}			2.49×10^{-17} ← VB	
MD		3.00×10^{-8} ← NNP			
VB		2.23×10^{-13} ← NNP	1.61×10^{-11} ← MD		1.02×10^{-20} ← DT
JJ			5.11×10^{-15} ← MD		
NN		1.03×10^{-10} ← NNP	5.36×10^{-15} ← MD		2.01×10^{-15} ← DT
RB			5.33×10^{-11} ← MD		
DT				1.82×10^{-12} ← VB	

HMM tagging

the lattice



HMM tagging

the lexicon and the grammar

- why did VB beat RB?
- at *back*, the lexicon favored RB by three to one
- at *the*, the grammar overturned it: VB → DT is 0.2231, RB → DT is 0.0479
- the dummy tagger from Tuesday only uses the lexicon
- the HMM keeps several possible paths and the best local one
- Viterbi decides at the end

HW 5

- released today, **due Thursday, Oct 29, 11:59 pm**
- a written homework
- build an HMM tagger by hand from a tiny tagged corpus, then compare with nltk
- submit one PDF on Canvas

next time

- Tuesday, Oct 27
  - named entities, annotation and corpora
- Thursday
  - parsing