Outline

- What is part of speech tagging?
- Markov chains
- Hidden Markov models
- Viterbi algorithm
- Example
- Coding assignment!

What is part of speech?

Why	not	learn	something	?
adverb	adverb	verb	noun	punctuation mark, sentence closer

Part of speech (POS) tagging

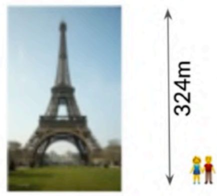
Part of speech tags:

lexical term	tag	example
noun	NN	something, nothing
verb	VB	learn, study
determiner	DT	the, a
w-adverb	WRB	why, where
...	...	

Why	not	learn	something	?
WRB	**RB**	**VB**	**NN**	**.**

Applications of POS tagging

Named entities



Co-reference resolution

Speech recognition

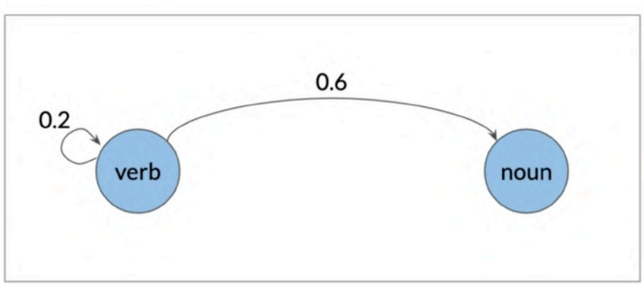
Part of Speech Dependencies

Markov Chains

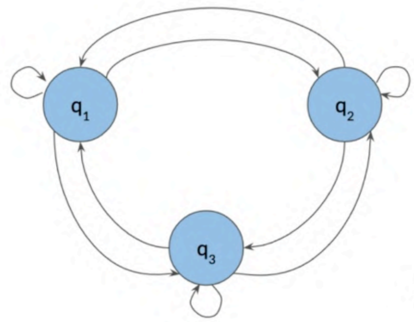
Why not learn ...

verb → verb? noun? ... ?

Visual Representation

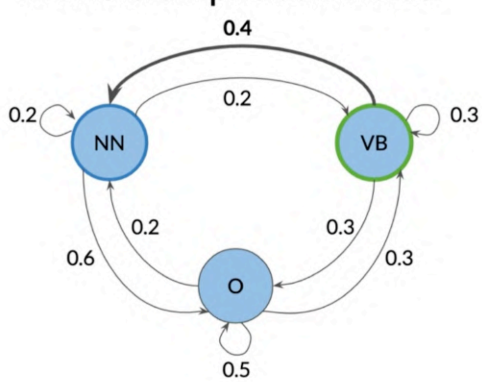


States



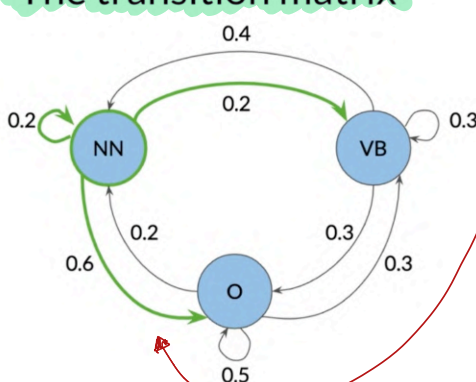
$$Q = \{q_1, q_2, q_3\}$$

Transition probabilities



Why not learn something?

The transition matrix



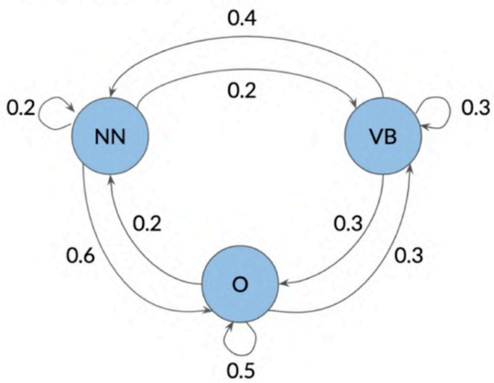
NN to O → 0.6

$A =$

	NN	VB	O
NN (noun)	0.2	0.2	0.6
VB (verb)	0.4	0.3	0.3
O (other)	0.2	0.3	0.5

$$\sum_{j=1}^{N} a_{ij} = 1$$

The first word



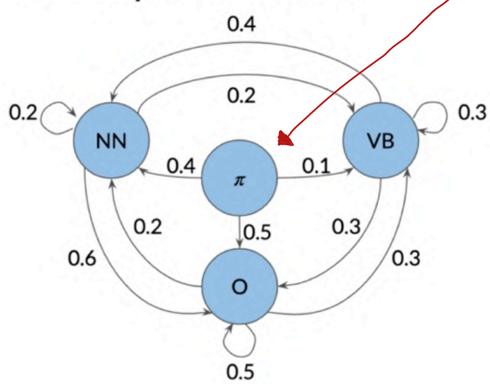
Why not learn something ?

NN?
VB?
O?

?

What do you do when there is ~~no previous word~~ as in the case when beginning a sentence. To handle this you can introduce what is known as an initial state by you include these probabilities in the Table A. So now it has dimensions n plus 1 by n.

Initial probabilities



$A =$

	NN	VB	O
π (initial)	0.4	0.1	0.5
NN (noun)	0.2	0.2	0.6
VB (verb)	0.4	0.3	0.3
O (other)	0.2	0.3	0.5

Transition table and matrix

$A =$

	NN	VB	O
π (initial)	0.4	0.1	0.5
NN (noun)	0.2	0.2	0.6
VB (verb)	0.4	0.3	0.3
O (other)	0.2	0.3	0.5

$$A = \begin{pmatrix} 0.4 & 0.1 & 0.5 \\ 0.2 & 0.2 & 0.6 \\ 0.4 & 0.3 & 0.3 \\ 0.2 & 0.3 & 0.5 \end{pmatrix}$$

(n+1, n)

Transition Matrix

Summary

States

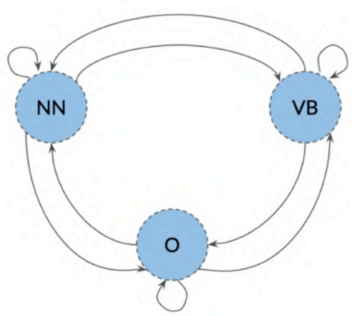
$$Q = \{q_1, \ldots, q_N\}$$

Transition matrix

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,N} \\ \vdots & \ddots & \vdots \\ a_{N+1,1} & \cdots & a_{N+1,N} \end{pmatrix}$$

States are hidden
or not directly
observable

Hidden Markov Model



hidden states

you

jump = verb
run = verb
fly = verb

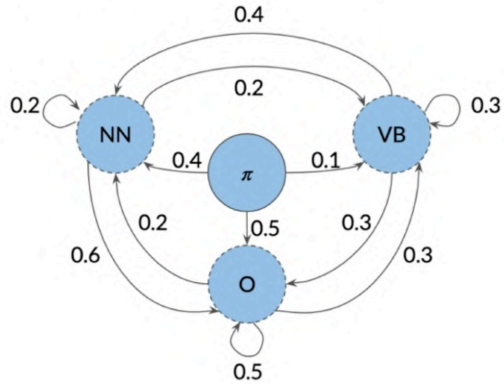
machine

jump
run
fly
*

For a machine looking at the text data, what it's going to observe are the actual words, such as jump, run, and fly. These words are said to be observable because they can be seen by the machine.

*observable

Transition probabilities



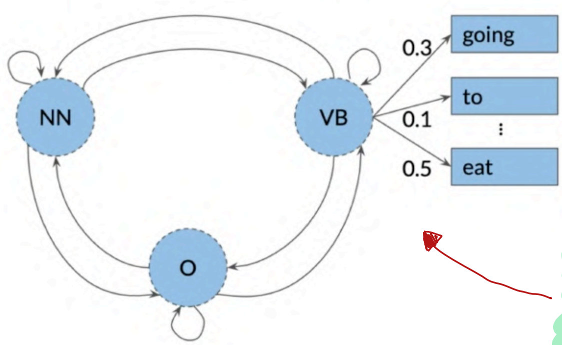
$A =$

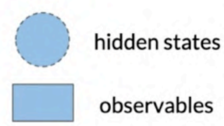
	NN	VB	O
π (initial)	0.4	0.1	0.5
NN (noun)	0.2	0.2	0.6
VB (verb)	0.4	0.3	0.3
O (other)	0.2	0.3	0.5

(n+1, n)

n = number of hidden states

Emission probabilities

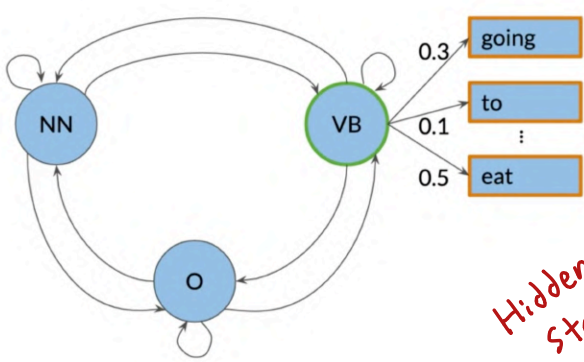




The hidden Markov model also has additional probabilities known as <u>emission probabilities</u>. These describe the transition from the hidden states of your hidden Markov model, which are parts of speech seen here as circles for noun, verb, and other, to the observables or the words of your corpus, shown here inside rectangles.

Emission probabilities

Observables



$B =$

	going	to	eat	...
NN (noun)	0.5	0.1	0.02	
VB (verb)	0.3	0.1	0.5	
O (other)	0.3	0.5	0.68	

Hidden states

The value, 0.5, is the emission probability of going from the state verb to emitting the observable eat.

The emission matrix

$B =$

	going	to	eat	...
NN (noun)	0.5	0.1	0.02	
VB (verb)	0.3	0.1	0.5	
O (other)	0.3	0.5	0.68	

$$\sum_{j=1}^{V} b_{ij} = 1$$

He lay on his back.

I'll be back.

The emission matrix represents the probabilities for the transition of your n hidden states representing your parts of speech tags to the n words in your corpus.

Summary

States

$$Q = \{q_1, \ldots, q_N\}$$

Transition matrix

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,N} \\ \vdots & \ddots & \vdots \\ a_{N+1,1} & \cdots & a_{N+1,N} \end{pmatrix}$$

$(n+1, n)$

Emission matrix

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1V} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & b_{NV} \end{pmatrix}$$

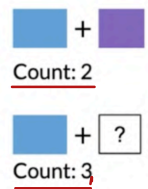
Transition probabilities

You eat
The oatmeal
You eat

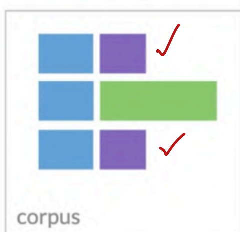
corpus

Transition probabilities

blue transitioning to purple

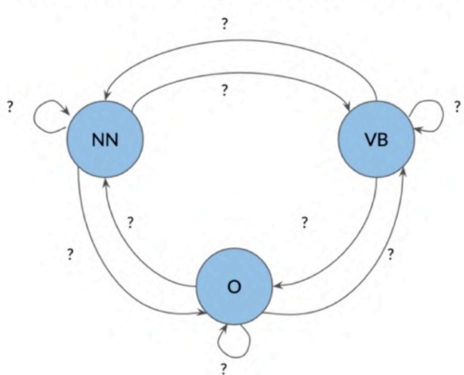


Transition probabilities



transition probability:  = ⅔

Transition probabilities



1. Count occurrences of tag pairs

$$C(t_{i-1}, t_i)$$ ← $C(t_{i-1} | t_i)$

2. Calculate probabilities using the counts

$$\boxed{P(t_i|t_{i-1})} = \frac{C(t_{i-1}, t_i)}{\sum_{j=1}^{N} C(t_{i-1}, t_j)}$$

total

The corpus

In a Station of the Metro
The apparition of these faces in the crowd :
Petals on a wet , black bough .

Ezra Pound – 1913

Preparation of the corpus

First, add the start token to each line or sentence in order to be able to calculate the initial probabilities using the previous defined formula.

<s> In a Station of the Metro
<s> The apparition of these faces in the crowd :
<s> Petals on a wet , black bough .

Ezra Pound – 1913

Preparation of the corpus

Then transform all words in the corpus to lowercase. So the model becomes case insensitive. The punctuation you should leave intact because it doesn't make a difference for a toy model and there aren't tags for different punctuation included here.

<s> in a station of the metro
<s> the apparition of these faces in the crowd :
<s> petals on a wet , black bough .

Ezra Pound – 1913

Populating the transition matrix

next states

$A =$

	NN	VB	O
π	1		
NN (noun)	C(NN,NN)		
VB (verb)	C(VB,NN)		
O (other)	C(O,NN)		

current states

Transition probabilities

<s> in a station of the metro
<s> the apparition of these faces in the crowd :
<s> petals on a wet , black bough .

Ezra Pound – 1913

Populating the transition matrix

$A =$

	NN	VB	O
π	1		
NN (noun)	0		
VB (verb)	0		
O (other)	6		

<s> in a station of the metro
<s> the apparition of these faces in the crowd :
<s> petals on a wet , black bough .

Ezra Pound – 1913

Populating the transition matrix

$A =$

	NN	VB	O
π	1	0	2
NN (noun)	0	0	
VB (verb)	0	0	0
O (other)	6	0	

<s> in a station of the metro
<s> the apparition of these faces in the crowd :
<s> petals on a wet , black bough .

Ezra Pound – 1913

Populating the transition matrix

$A =$

	NN	VB	O
π	1	0	2
NN (noun)	0	0	6
VB (verb)	0	0	0
O (other)	6	0	

<s> in a station of the metro
<s> the apparition of these faces in the crowd :
<s> petals on a wet , black bough .

Ezra Pound – 1913

Populating the transition matrix

$A =$

	NN	VB	O
π	1	0	2
NN (noun)	0	0	6
VB (verb)	0	0	0
O (other)	6	0	8

<s> in a station of the metro
<s> the apparition of these faces in the crowd :
<s> petals on a wet , black bough .

Ezra Pound – 1913

Populating the transition matrix

$A =$

	NN	VB	O	
π	1	0	2	3
NN	0	0	6	6
VB	0	0	0	0
O	6	0	8	14

$$P(\text{NN}|\pi) = \frac{\boxed{C(\pi, \text{NN})}}{\boxed{\sum_{j=1}^{N} C(\pi, t_j)}} = \frac{1}{3}$$

Populating the transition matrix

$A =$

	NN	VB	O	
π	1	0	2	3
NN	0	0	6	6
VB	0	0	0	0
O	6	0	8	14

$$P(\mathrm{NN}|\mathrm{O}) = \frac{\boxed{C(\mathrm{O},\mathrm{NN})}}{\boxed{\sum_{j=1}^{N} C(\mathrm{O},t_j)}} = \frac{6}{14}$$

Populating the transition matrix

$A =$

	NN	VB	O	
π	1	0	2	3
NN	0	0	6	6
VB	0	0	0	0
O	6	0	8	14

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1},t_i)}{\sum_{j=1}^{N} C(t_{i-1},t_j)}$$

You may have realized that there are two problems here. One is that the row sum of the VB tag is zero, which would lead to a division by zero using this formula. The other is that a lot of entries in the transition matrix are zero, meaning that these transitions will have probability zero. This won't work if you want the model to generalize to other equals, which might actually contain verbs. To handle this, change your formula slightly by adding a small value epsilon to each of the accounts in the numerator, and add N times epsilon to the divisor such that the row sum still adds up to one. This operation is also referred to as smoothing, which you might remember from previous lessons. So if you substitute the epsilon with a small value,

Smoothing

$A =$

	NN	VB	O	
π	1+ε	0+ε	2+ε	3+3*ε
NN	0+ε	0+ε	6+ε	6+3*ε
VB	0+ε	0+ε	0+ε	0+3*ε
O	6+ε	0+ε	8+ε	14+3*ε

Smoothing

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1},t_i) + \boxed{\epsilon}}{\sum_{j=1}^{N} C(t_{i-1},t_j) + \boxed{N} * \boxed{\epsilon}}$$

In the real-world example, you might not want to apply smoothing to the initial probabilities in the first row of the transition matrix. That's because if you apply smoothing to that row by adding a small value to possibly zeroed valued entries, you'll effectively allow a sentence to start with any parts of speech tag, including punctuation.

Smoothing

$A =$

	NN	VB	O
π	0.3333	0.0003	0.6663
NN	0.0001	0.0001	0.9996
VB	0.3333	0.3333	0.3333
O	0.4285	0.0000	0.5713

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i) + \epsilon}{\sum_{j=1}^{N} C(t_{i-1}, t_j) + N * \epsilon}$$

Emission probabilities

You eat
The oatmeal
You eat

corpus

You

Count: 2

Count: 3

Emission probabilities

You eat
The oatmeal
You eat

corpus

emission probability: You = ⅔

The emission matrix

$B =$

	in	a	...
NN (noun)	$C(\text{NN}, \text{in})$		
VB (verb)	$C(\text{VB}, \text{in})$		
O (other)	$C(\text{O}, \text{in})$		

<s> in a station of the metro
<s> the apparition of these faces in the crowd :
<s> petals on a wet , black bough .

Ezra Pound - 1913

The emission matrix

$B =$

	in	a	...
NN (noun)	0		
VB (verb)	0		
O (other)	2		

<s> in a station of the metro
<s> the apparition of these faces in the crowd :
<s> petals on a wet , black bough .

Ezra Pound - 1913

The emission matrix

$B =$

	in	a	...
NN (noun)	0	...	...
VB (verb)	0	...	...
O (other)	2	...	...

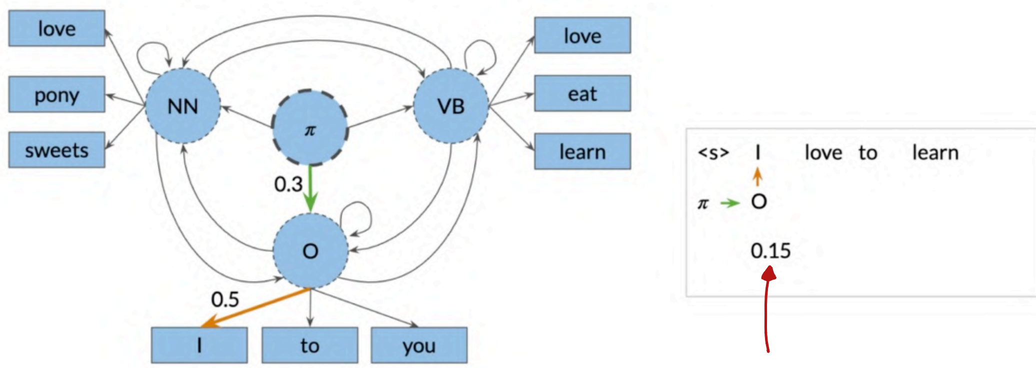
$$P(w_i|t_i) = \frac{C(t_i, w_i) + \epsilon}{\sum_{j=1}^{V} C(t_i, w_j) + N * \epsilon}$$

$$= \frac{C(t_i, w_i) + \epsilon}{C(t_i) + N * \epsilon}$$

Summary

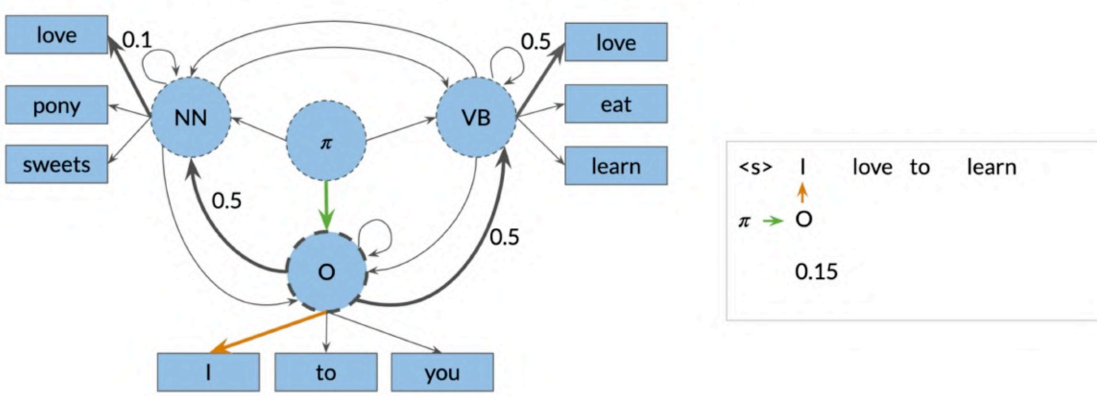
1. Calculate transition and emission matrix
2. How to apply smoothing

Viterbi algorithm – a graph algorithm

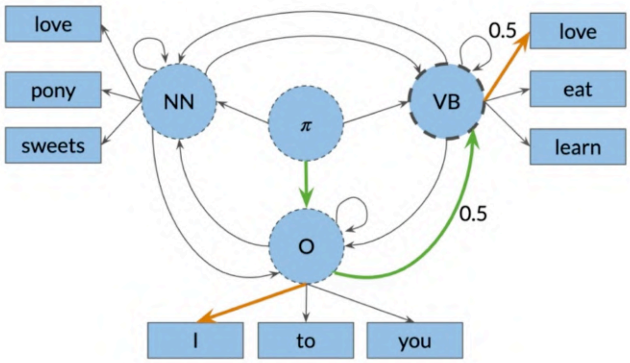


0.3 * 0.5=0.15

Viterbi algorithm – a graph algorithm



Viterbi algorithm – a graph algorithm



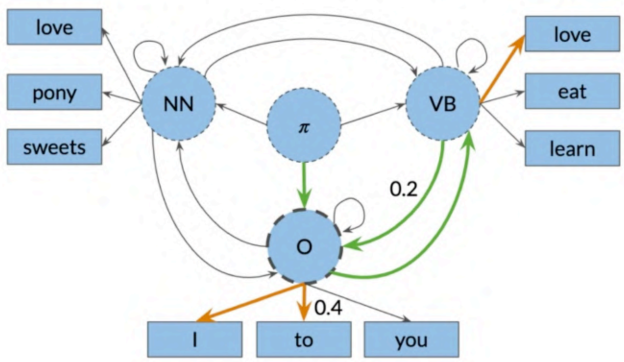
<s> I love to learn

$\pi \rightarrow O \rightarrow VB$

0.15 0.25

$O \xrightarrow{0.5} VB \xrightarrow{0.5} love = 0.25$

Viterbi algorithm – a graph algorithm



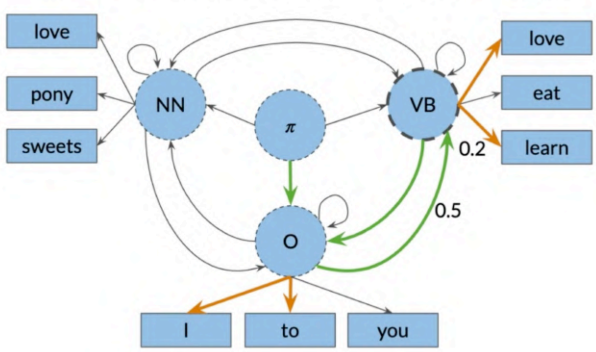
<s> I love to learn

$\pi \rightarrow O \rightarrow VB \rightarrow O$

0.15 0.25 0.08

$VB \xrightarrow{0.2} O \xrightarrow{0.4} to = 0.08$

Viterbi algorithm – a graph algorithm



<s> I love to learn

$\pi \rightarrow O \rightarrow VB \rightarrow O \rightarrow VB$

0.15 * 0.25 * 0.08 * 0.1

Probability for this sequence of hidden states: 0.0003

Viterbi algorithm – Steps

1. Initialization step
2. Forward pass
3. Backward pass

$C =$

	w_1	w_2	...	w_K
t_1				
...				
t_N				

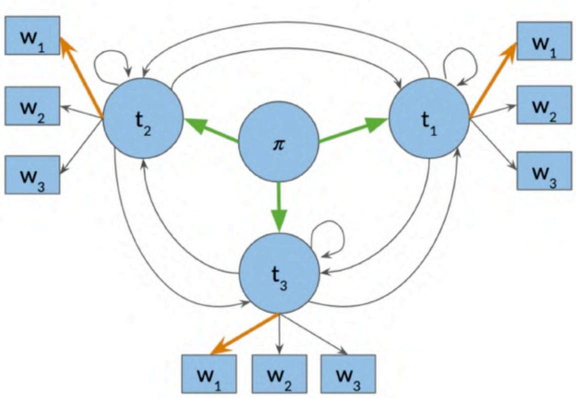
$D =$

	w_1	w_2	...	w_K
t_1				
...				
t_N				

Viterbi algorithm – Steps

1. Initialization step

Initialization step

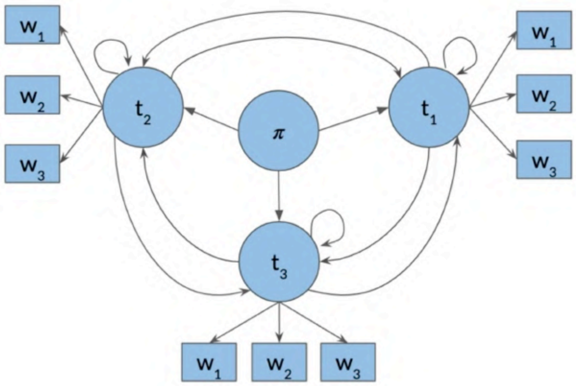


$C =$

	w_1	w_2	...	w_K
t_1	$c_{1,1}$			
...				
t_N	$c_{N,1}$			

$$c_{i,1} = \boxed{\pi_i} * \boxed{b_{i,cindex(w_1)}}$$
$$= a_{1,i} * b_{i,cindex(w_1)}$$

Initialization step



$D =$

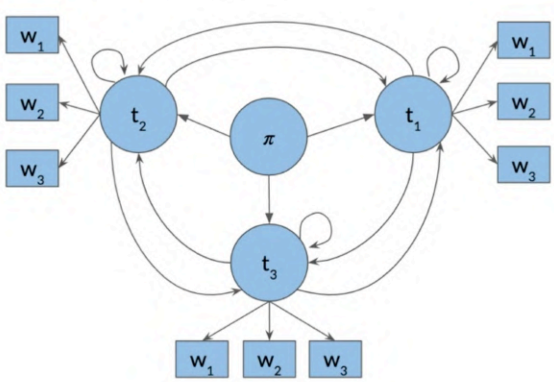
	w_1	w_2	...	w_K
t_1	$d_{1,1}$			
...				
t_N	$d_{N,1}$			

$$d_{i,1} = 0$$

Viterbi algorithm – Steps

2. Forward pass

Forward pass



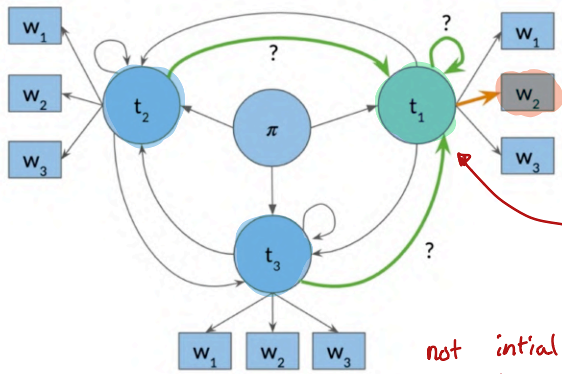
$C =$

	w_1	w_2	...	w_K
t_1	$c_{1,1}$	$c_{1,2}$		$c_{1,K}$
...				
t_N	$c_{N,1}$	$c_{N,2}$		$c_{N,K}$

$$c_{i,j} = \max_k c_{k,j-1} * a_{k,i} * b_{i,cindex(w_j)}$$

Forward pass

Populated column by column



$C =$

	w_1	w_2	...	w_K
t_1	$c_{1,1}$	$\boxed{c_{1,2}}$		$c_{1,K}$
...				
t_N	$c_{N,1}$	$c_{N,2}$		$c_{N,K}$

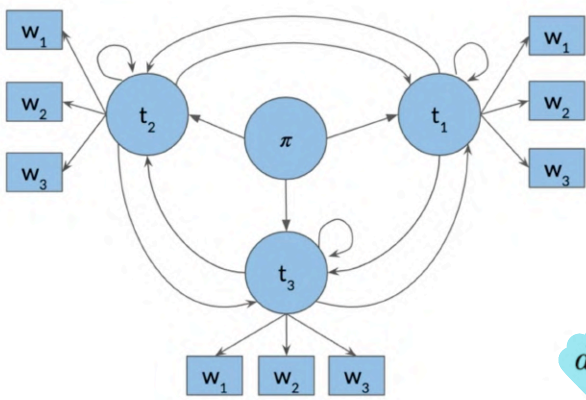
$$c_{1,2} = \max_k \boxed{c_{k,1}} * \boxed{a_{k,1}} * \boxed{b_{1,cindex(w_2)}}$$

not intialized states

states transitions come from

prev transition state

Forward pass



$D =$

	w_1	w_2	...	w_K
t_1	$d_{1,1}$	$d_{1,2}$		$d_{1,K}$
...				
t_N	$d_{N,1}$	$d_{N,2}$		$d_{N,K}$

Save max

$$c_{i,j} = \max_k c_{k,j-1} * a_{k,i} * b_{i,cindex(w_j)}$$

$$d_{i,j} = \operatorname{argmax}_k c_{k,j-1} * a_{k,i} * b_{i,cindex(w_j)}$$

Save K of max

Viterbi algorithm – Steps

3. Backward pass

Backward pass

First, calculate the index of the entry Ci,K with the highest probability in the last column of C. The probability at this index is the probability of the most likely sequence of hidden states, generating the given sequence of words.

$C =$

	w_1	w_2	...	w_K
t_1	$c_{1,1}$	$c_{1,2}$		$c_{1,K}$
...				
t_N	$c_{N,1}$	$c_{N,2}$		$c_{N,K}$

$D =$

	w_1	w_2	...	w_K
t_1	$d_{1,1}$	$d_{1,2}$		$d_{1,K}$
...				
t_N	$d_{N,1}$	$d_{N,2}$		$d_{N,K}$

$$\boxed{s = \operatorname*{argmax}_i c_{i,K}}$$

You use this index s to traverse backwards through the matrix D, to reconstruct the sequence of parts of speech tags. First, calculate the index of the entry Ci,K with the highest probability in the last column of C. The probability at this index is the probability of the most likely sequence of hidden states, generating the given sequence of words. You use this index s to traverse backwards through the matrix D, to reconstruct the sequence of parts of speech tags.

Backward pass

$D =$

	w_1	w_2	w_3	w_4	w_5
t_1	0	1	3	2	3
t_2	0	2	4	1	3
t_3	0	2	4	1	4
t_4	0	4	4	3	1

<s> w1 w2 w3 w4 w5

The matrix D, stores all the labels of the hidden states you've traversed in the forward path. If you're going back through the states, starting with the path that has the highest probability, you effectively got the most likely sequence of hidden states, or parts of speech sites. You start by looking up the entry with the highest probability in the last row of the matrix C, and extract the index s of that entry.

Backward pass

$C =$

	w_1	w_2	w_3	w_4	w_5
t_1	0.25	0.125	0.025	0.0125	0.01
t_2	0.1	0.025	0.05	0.01	0.003
t_3	0.3	0.05	0.025	0.02	0.0000
t_4	0.2	0.1	0.000	0.0025	0.0003

Highest Probability

$$\boxed{s = \operatorname*{argmax}_i c_{i,K} = 1}$$

Backward pass

$D =$

	w_1	w_2	w_3	w_4	w_5
t_1	0	1	3	2	3
t_2	0	2	4	1	3
t_3	0	2	4	1	4
t_4	0	4	4	3	1

$$\boxed{s = \operatorname*{argmax}_i c_{i,K} = 1}$$

<s> w1 w2 w3 w4 w5

Backward pass

$D =$

	w_1	w_2	w_3	w_4	w_5
t_1	0	1	3	2	3
t_2	0	2	4	1	3
t_3	0	2	4	1	4
t_4	0	4	4	3	1

<s> w1 w2 w3 w4 w5

$t_3 \leftarrow t_1$

Backward pass

$D =$

	w_1	w_2	w_3	w_4	w_5
t_1	0	1	3	2	3
t_2	0	2	4	1	3
t_3	0	2	4	1	4
t_4	0	4	4	3	1

<s> w1 w2 w3 w4 w5

$t_1 \leftarrow t_3 \leftarrow t_1$

Backward pass

$D =$

	w_1	w_2	w_3	w_4	w_5
t_1	0	1	3	2	3
t_2	0	2	4	1	3
t_3	0	2	4	1	4
t_4	0	4	4	3	1

<s> w1 w2 w3 w4 w5

$t_3 \leftarrow t_1 \leftarrow t_3 \leftarrow t_1$

Backward pass

$D =$

	w_1	w_2	w_3	w_4	w_5
t_1	0	1	3	2	3
t_2	0	2	4	1	3
t_3	0	2	4	1	4
t_4	0	4	4	3	1

"End" at π

<s> w1 w2 w3 w4 w5

$\pi \leftarrow t_2 \leftarrow t_3 \leftarrow t_1 \leftarrow t_3 \leftarrow t_1$

Implementation notes

1. In Python index starts with 0!
2. Use log probabilities

$$c_{i,j} = \max_k c_{k,j-1} * a_{k,i} * b_{i,cindex(w_j)}$$

$$log(c_{i,j}) = \max_k log(c_{k,j-1}) + log(a_{k,i}) + log(b_{i,cindex(w_j)})$$

when you multiply many very small numbers like probabilities, this will lead to numerical issues, so you should use log probabilities instead, where numbers are summed instead of multiplied.