Part of Speech Tagging and HMM

What is part of speech tagging ?

Adverb, verb , noun, adj, pronoun, punctuation, sentence closer

Part of speech (POS) tagging

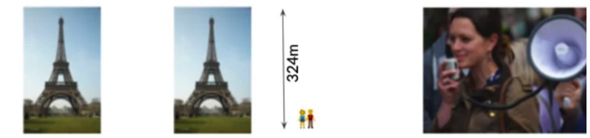
Part of speech tags:

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

Applications of POS tagging

Why not learn something ?

WRB RB VB NN .



Named entities Co-reference resolution Speech recognition

Markov chains

Markov chains are really important because they are used in speech recognition and for parts of speech tagging.

Visual Representation

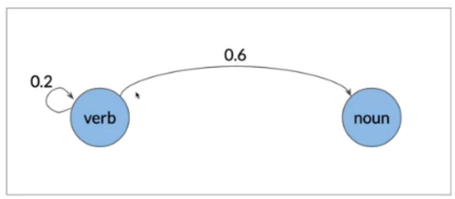
Part of Speech Dependencies

Why not learn ...

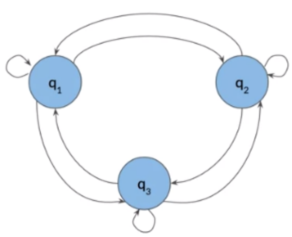
verb verb?

noun?

... ?



They're a type of stochastic model that describes a sequence of possible events. To get the probability for each event, it needs only the states of the previous events. The word stochastic just means random or randomness. So a stochastic model incorporates and models processes does have a random component to them. A Markov chain can be depicted as a directed graph. Graph of states and transitions between them

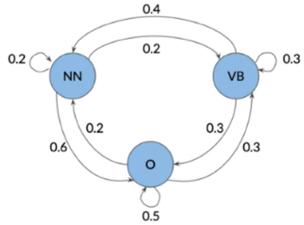


$Q = \{q_1, q_2, q_3\}$

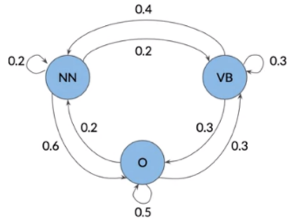
Markov chains and POS tags

Markov property, which states that the probability of the next event only depends on the current events. The Markov property helps keep the model simple by saying all you need to determine the next state is the current states. It doesn't need information from any of the previous states

Transition probabilities



The transition matrix



$A =$

	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

Transition table and matrix

$A =$

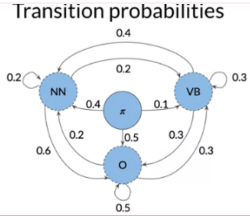
	NN	VB	O
π (initial)	0.4	0.1	0.5
NN (noun)	0.2	0.2	0.6
VB (verb)	0.4	0.3	0.3
O (other)	0.2	0.3	0.5

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

Hidden Markov models

The name hidden Markov model implies that states are hidden or not directly observable.
Going back to the Markov model that has the states for the parts of speech, such as noun, verb, or other, you can now think of these as hidden states because these are not directly observable from the text data

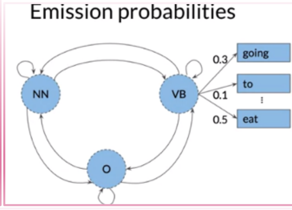
Transition probabilities



$A =$

	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

Emission probabilities



$B =$

	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	

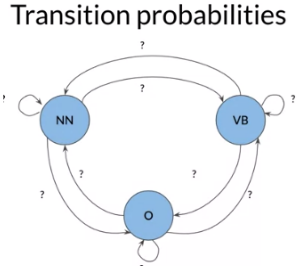
States — Transition matrix — Emission matrix

$$Q = \{q_1, \ldots, q_N\} \quad A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,N} \\ \vdots & \ddots & \vdots \\ a_{N+1,1} & \cdots & a_{N+1,N} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & \cdots & b_{1V} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & b_{NV} \end{pmatrix}$$

Calculating probabilities

Calculate prob for both transition and emission matrices.

Transition probabilities



1. Count occurrences of tag pairs

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

2. Calculate probabilities using the counts

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

Populating the transition matrix

- Add a start tag <s>
- Lowercase the text

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

- 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

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

-

Populating Emission Matrix

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 .

The emission matrix

$B =$

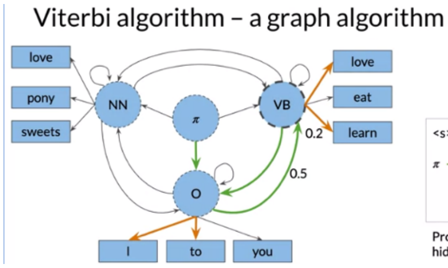
	in	a	...
NN (noun)	0	...	...
VB (verb)	0	...	...
O (other)	2	...	...

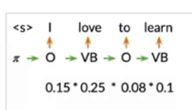
$$P(w_i|t_i) = \frac{C(t_i, w_i) + \epsilon}{\sum_{j=1}^{V} C(t_i, w_j) + N * \epsilon} = \frac{C(t_i, w_i) + \epsilon}{C(t_i) + N * \epsilon}$$

Viterbi algorithm (graph algo)

The Viterbi algorithm actually computes several such paths at the same time in order to find the most likely sequence of hidden states. It uses the matrix representation of the Hidden Markov model

Viterbi algorithm – a graph algorithm





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				

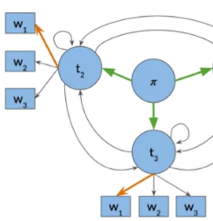
$D =$

	w_1	w_2	...	w_K
t_1				
...				
t_N				

Given your transition and emission probabilities, you first populate and then use the auxiliary matrices C and D. The matrix C holds the intermediate optimal probabilities and matrix D, the indices of the visited states. As you're traversing the model graph to find the most likely sequence of parts of speech tags for the given sequence of words, W_1 all the way to W_K. These two matrices have n rows, where n is the number of parts of speech tags or hidden states in our model, and k columns, where k is the number of words in the given sequence.

Viterbi Initialization

Initialization step

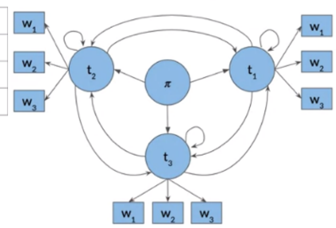


$C =$

	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



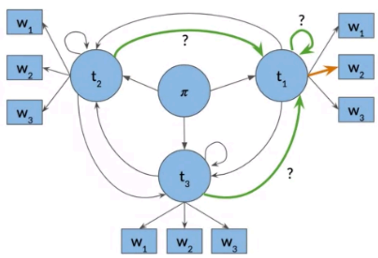
$D =$

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

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

Viterbi Forward Pass

Forward pass

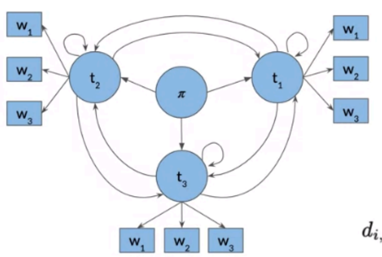


$C =$

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

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

Forward pass



$D =$

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

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

Viterbi Backward Pass

- Find max probability of last column in matrix C.
-

$C =$

	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

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

$D =$

	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

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

- For w5 it is first cell. Now look at matrix D, first cell , last col is 3. Which is t3, then t1, t3, t2 and finally 0.

Backward pass

- $$D = \begin{array}{c|ccccc} & w_1 & w_2 & w_3 & w_4 & w_5 \\ \hline t_1 & 0 & 1 & \boxed{3} & 2 & \boxed{3} \\ t_2 & \boxed{0} & 2 & 4 & 1 & 3 \\ t_3 & 0 & \boxed{2} & 4 & \boxed{1} & 4 \\ t_4 & 0 & 4 & 4 & 3 & 1 \end{array}$$

<s>	w1	w2	w3	w4	w5
π	$\leftarrow t_2$	$\leftarrow t_3$	$\leftarrow t_1$	$\leftarrow t_3$	$\leftarrow t_1$