
Foundations of Natural Language Processing

Lecture 25

Part-of-speech Tagging and HMMs

Ivan Titov



Sequence labeling problems

- ▶ **Definition:**

- ▶ Input: sequences of variable length $\mathbf{x} = (x_1, x_2, \dots, x_{|x|}), x_i \in \mathcal{X}$
- ▶ Output: every position is associated with a label $\mathbf{y} = (y_1, y_2, \dots, y_{|x|}), y_i \in \{1, \dots, N\}$

- ▶ **An example:**

- ▶ Part-of-speech tagging

$\mathbf{x} =$	John	carried	a	tin	can	.
$\mathbf{y} =$	NNP	VBD	DT	NN	NN	.

- ▶ Named-entity recognition, shallow parsing ("chunking"), gesture recognition from video-streams, ...

Part-of-speech tagging: ambiguity

In fact, even knowing that the previous word is a noun is not enough

x =	John	carried	a	tin	can	.
y =	NNP	VBD	DT	NN	NN or MD?	.

► Labels:

- NNP – proper singular noun;
- VBD – verb, past tense
- DT – determiner
- NN – singular noun
- MD - modal
- . - final punctuation

If you just predict the most frequent tag for each word you will make a mistake here

► Consider

Tin	can	cause	poisoning	...
NN	MD	VB	NN	...

One need to model interaction between labels to successfully resolve ambiguities, hence called *structured prediction* problem

Named Entity Recognition

[ORG Chelsea], despite their name, are not based in [LOC Chelsea], but in neighbouring [LOC Fulham] .

[PER Bill Clinton] embarrassed [PER Chelsea] at her wedding at [LOC Astor Courts]

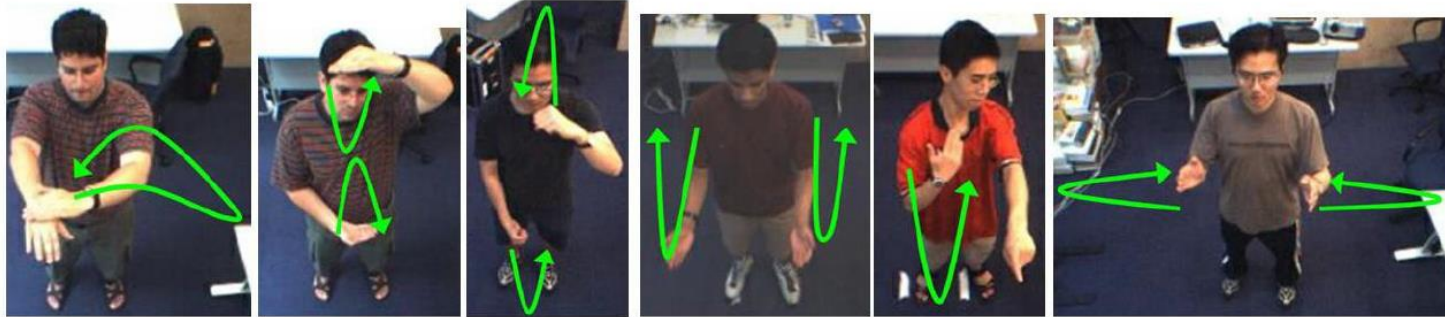
X =	Bill	Clinton	embarrassed	Chelsea	at	her	wedding	at	Astor	Courts
y =	B-PERS	I-PERS	O	B-PERS	O	O	O	O	B-LOC	I-LOC

Vision: Gesture Recognition

- ▶ Given a sequence of frames in a video annotate each frame with a gesture type:



- ▶ Types of gestures:



Flip back

Shrink
vertically

Expand
vertically

Double
back

Point
and back

Expand horizontally

It may be hard to predict gestures for each frame in isolation, again need to exploit interaction between gestures in different frames

Hidden Markov Models

- ▶ We will consider the part-of-speech (POS) tagging example

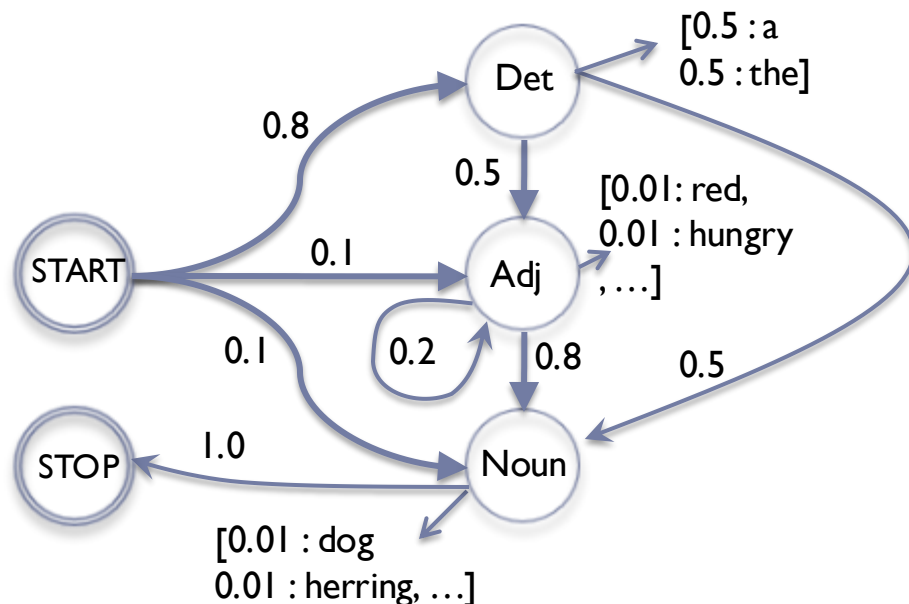
John	carried	a	tin	can	.
NNP	VBD	DT	NN	NN	.

- ▶ A “**generative**” model, i.e.:
 - ▶ **Model:** Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$
 - ▶ **Learning:** use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$
 - ▶ **Decoding:** $\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

Hidden Markov Models

A simplistic state diagram for noun phrases:

N – tags, M – vocabulary size



Example:

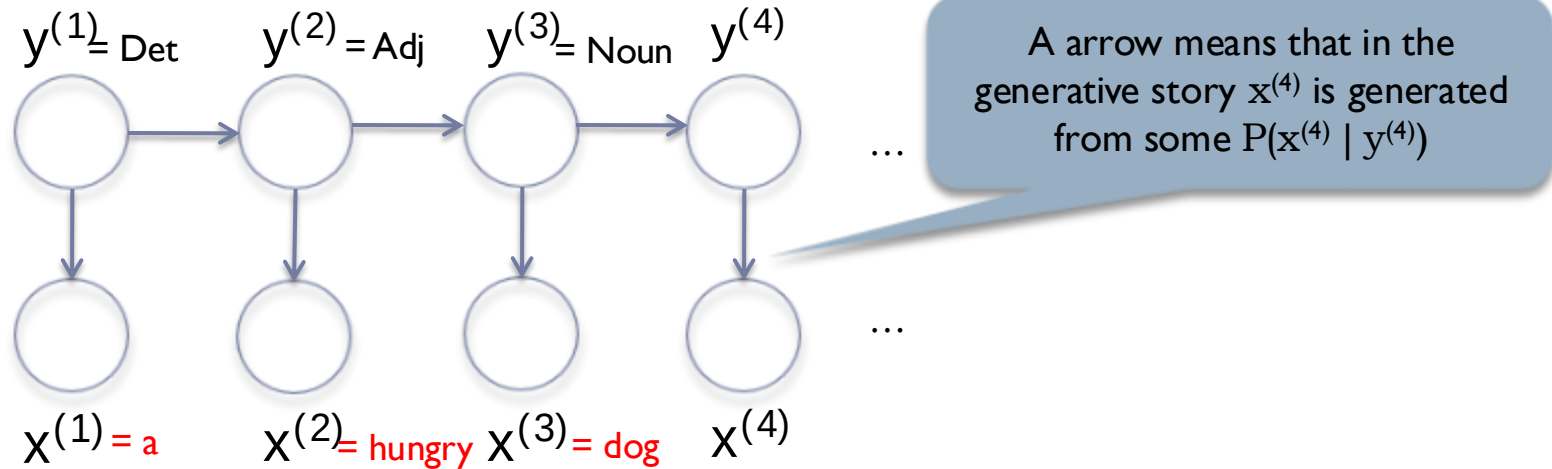
a hungry dog

- ▶ States correspond to POS tags,
- ▶ Words are emitted independently from each POS tag
- ▶ Parameters (to be estimated from the training set):
 - ▶ Transition probabilities $P(y^t | y^{t-1})$: $[N \times N]$ matrix
 - ▶ Emission probabilities $P(x^t | y^t)$: $[N \times M]$ matrix

Stationarity assumption: this probability does not depend on the position in the sequence t

1st order Hidden Markov Models

Representation as an instantiation of *a graphical model*: N – tags, M – vocabulary size



- ▶ States correspond to POS tags,
- ▶ Words are emitted independently from each POS tag
- ▶ Parameters (to be estimated from the training set):
 - ▶ Transition probabilities $P(y^t | y^{t-1})$: $[N \times N]$ matrix
 - ▶ Emission probabilities $P(x^t | y^t)$: $[N \times M]$ matrix

Stationarity assumption: this probability does not depend on the position in the sequence t

Hidden Markov Models

- ▶ We will consider the part-of-speech (POS) tagging example

John	carried	a	tin	can	.
NNP	VBD	DT	NN	NN	.

- ▶ A “**generative**” model, i.e.:

- ✗ ▶ **Model:** Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$
- ▶ **Learning:** use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$
- ▶ **Decoding:** $\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

We do not want to specify the transition system (associated probabilities) but learn it from the data

Hidden Markov Models: Estimation

- ▶ N – the number tags, M – vocabulary size
- ▶ **Parameters** (to be estimated from the training set):
 - ▶ Transition probabilities $a_{ij} = P(y^t = j \mid y^{t-1} = i)$, A - $[N \times N]$ matrix
 - ▶ Emission probabilities $b_{ik} = P(x^t = k \mid y^t = i)$, B - $[N \times M]$ matrix
- ▶ **Training corpus:**
 - ▶ (In, an, Oct., 19, review, of,), (IN, DT, NNP, CD, NN, IN,)
 - ▶ (Ms., Haag, plays, Elianti,.), (NNP, NNP, VBZ, NNP, .)
 - ▶ ...
 - ▶ (The, company, said,...), (DT, NN, VBD, NNP, .)

Hidden Markov Models: Estimation

- ▶ N – the number tags, M – vocabulary size
- ▶ **Parameters** (to be estimated from the training set):
 - ▶ Transition probabilities $a_{ij} = P(y^t = j \mid y^{t-1} = i)$, A - $[N \times N]$ matrix
 - ▶ Emission probabilities $b_{ik} = P(x^t = k \mid y^t = i)$, B - $[N \times M]$ matrix
- ▶ **Training corpus:**
 - ▶ (In, an, Oct., I 9, review, of,), (IN, DT, NNP, CD, NN, IN,)
 - ▶ (Ms., Haag, plays, Elianti,.), (NNP, NNP, VBZ, NNP, .)
 - ▶ ...
 - ▶ (The, company, said,...), (DT, NN, VBD, NNP, .)

$$P(x^t = k \mid y^t = i) = b_{ik} = \frac{C_E(i, k)}{\sum_{k'} C_E(i, k')}$$

$C_E(i, k)$ is #times
word k is emitted
by tag i

Hidden Markov Models: Estimation

- ▶ N – the number tags, M – vocabulary size
- ▶ **Parameters** (to be estimated from the training set):
 - ▶ Transition probabilities $a_{ij} = P(y^t = j \mid y^{t-1} = i)$, A - $[N \times N]$ matrix
 - ▶ Emission probabilities $b_{ik} = P(x^t = k \mid y^t = i)$, B - $[N \times M]$ matrix
- ▶ **Training corpus:**
 - ▶ (In, an, Oct., 19, review, of,), (IN, DT, NNP, CD, NN, IN,)
 - ▶ (Ms., Haag, plays, Elianti,.), (NNP, NNP, VBZ, NNP, .)
 - ▶ ...
 - ▶ (The, company, said,...), (DT, NN, VBD, NNP, .)

$$P(y^t = j \mid y^{t-1} = i) = a_{ij} = \frac{C_T(i, j)}{\sum_{j'} C_T(i, j')}$$

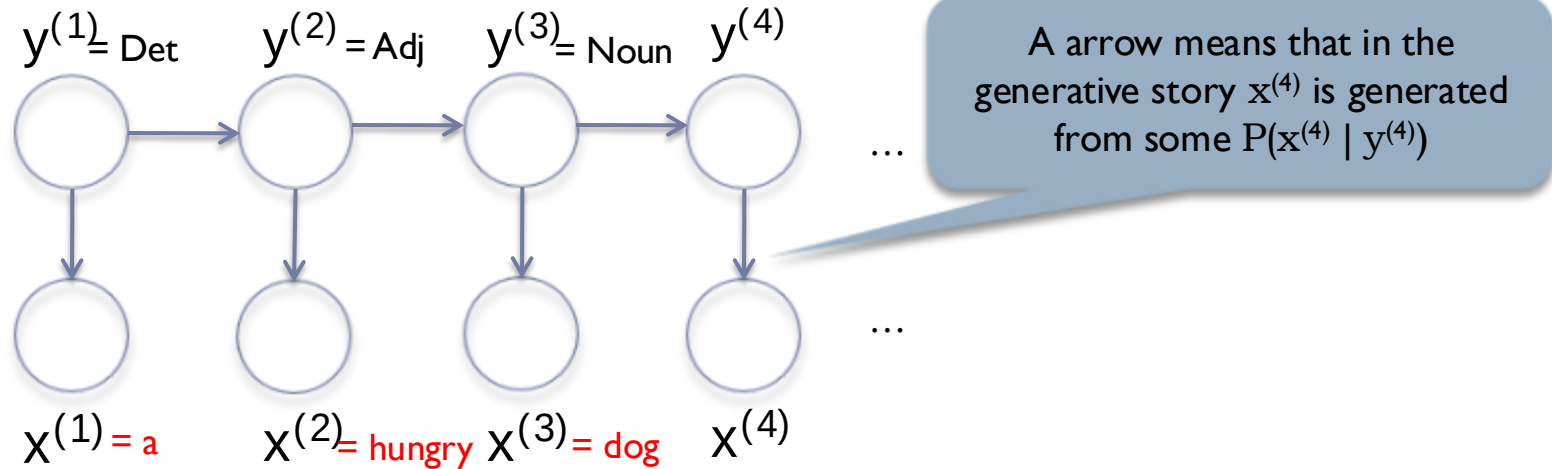
It is convenient to think that the output sequences are padded with START and STOP:

$$y^0 = START, y^{|x|+1} = STOP$$

$C_T(i, j)$ is #times
tag i is followed by
tag j

1st Order

Representation as an instantiation of a *graphical model*: N – tags, M – vocabulary size

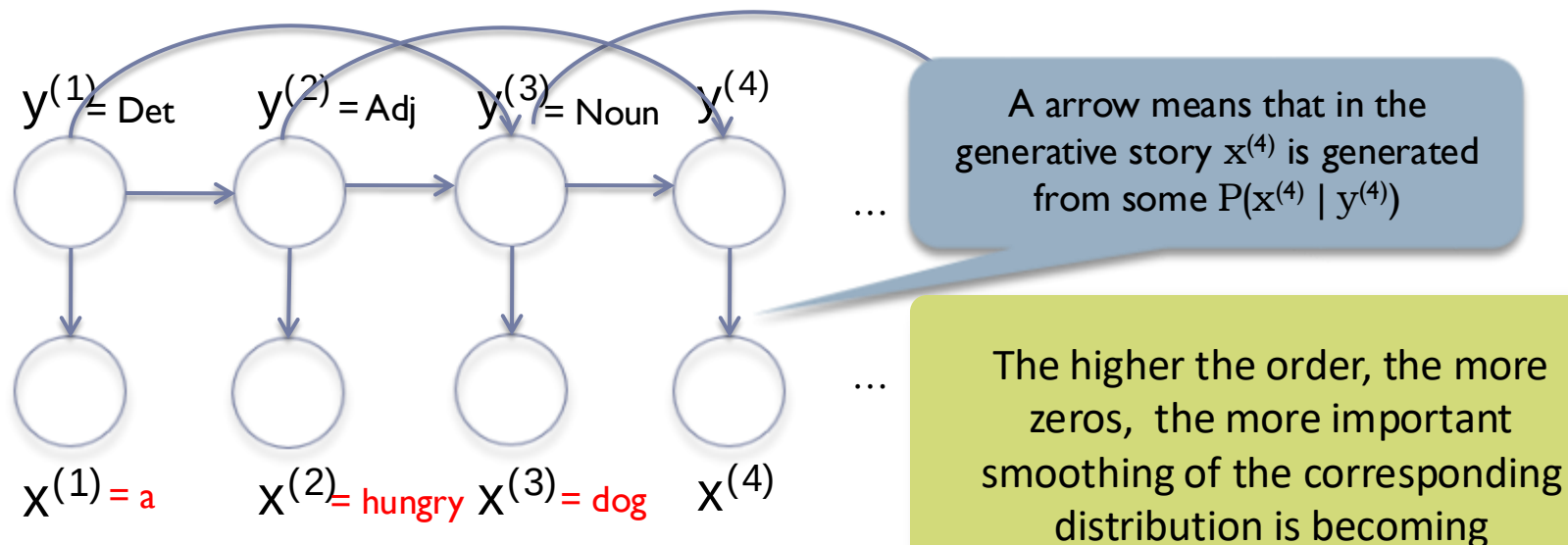


- ▶ HMM States correspond to POS tags,
- ▶ Words are emitted independently from each POS tag
- ▶ Parameters (to be estimated from the training set):
 - ▶ Transition probabilities $P(y^t | y^{t-1})$: $[N \times N]$ matrix
 - ▶ Emission probabilities $P(x^t | y^t)$: $[N \times M]$ matrix

Stationarity assumption: this probability does not depend on the position in the sequence t

2nd Order

Representation as an instantiation of a *graphical model*: N – tags, M – vocabulary size



- ▶ HMM States correspond to POS tags,
- ▶ Words are emitted independently from each POS tag
- ▶ Parameters (to be estimated from the training set):

- ▶ Transition probabilities $P(y^t | y^{t-1}, y^{t-2})$: $[N \times N \times N]$ matrix
- ▶ Emission probabilities $P(x^t | y^t)$: $[N \times M]$ matrix

Stationarity assumption: this probability does not depend on the position in the sequence t

Hidden Markov Models

- ▶ We will consider the part-of-speech (POS) tagging example

John	carried	a	tin	can	.
NNP	VBD	DT	NN	NN	.

- ▶ A “**generative**” model, i.e.:

✗ ▶ **Model:** Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$

✗ ▶ **Learning:** use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$

▶ **Decoding:** $\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

Hidden Markov Models: decoding

- ▶ Predict PoS-tags for a sentence

John	carried	a	tin	can	.
?	?	?	?	?	?

- ▶ Corresponds to maximization:

$$\hat{\mathbf{y}} = \arg \max_y P(\mathbf{y}|\mathbf{x}, A, B) = \arg \max_y P(\mathbf{y}, \mathbf{x}|\theta)$$

- ▶ Brute force

- ▶ Consider all the sequences $\mathbf{y}$ and choose the highest scored one: $O(N^{|\mathbf{x}|})$

- ▶ A better alternative -- a dynamic programming algorithm, Viterbi $O(|\mathbf{x}|N^2)$

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

The probability of the most probable sequence up to t ending with a tag i

$$\max_{y_1, \dots, y^{t-1}} P(x_1, \dots, x_t, y_1, \dots, y_t = i | \theta)$$

	$time_1$	$flies_2$	$fast_3$	-
NN				
VB				
JJ				
RB				
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.5x0.1=0.05			
VB				
JJ				
RB				
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.5x0.1=0.05			
VB	0.25x0.01=0.0025			
JJ				
RB				
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.5x0.1=0.05			
VB	0.25x0.01=0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05			
VB	0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05 $\xrightarrow{\times 0.25}$			
VB	0.0025 $\xrightarrow{\times 0.25}$			
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05 $\xrightarrow{\times 0.25}$ 0.05 x 0.25 x 0.01			
VB	0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05	1.25E-4		
VB	0.0025			
JJ	0			
RB	0			
STOP	-	-	-	

store the backpointer

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05	1.25E-4		
VB	0.0025	0.5		
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05	1.25E-4		
VB	0.0025	0.05 x 0.5 x 0.1		
JJ	0			
RB	0			
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05	1.25E-4		
VB	0.0025	0.0025		
JJ	0			
RB	0			
STOP	-	-	-	

again, store the backpointer

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05	1.25E-4	6.25E-6	
VB	0.0025	0.0025	6.25E-7	
JJ	0	0	6.25E-5	
RB	0	0	6.25E-5	
STOP	-	-	-	

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

Final: $v_{STOP}^{|x|+1} = \max_i v_i^{|x|} a_{i,STOP}$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	-

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$$

Final: $v_{STOP}^{|x|+1} = \max_i v_i^{|x|} a_{i,STOP}$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	6.25E-5 x 0.5

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$$

Final: $v_{STOP}^{|x|+1} = \max_i v_i^{|x|} a_{i,STOP}$

	$time_1$	$flies_2$	$fast_3$	-
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	3.125E-5

Hidden Markov Models: decoding

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	...	...	...
NN	0.1	0.01	0.01	...	...	...
VB	0.01	0.1	0.01	...	...	...
JJ	0	0	0.1	...	...	...
RB	0	0	0.1	...	...	...

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$$

Final: $v_{STOP}^{|x|+1} = \max_i v_i^{|x|} a_{i,STOP}$

Retrace the
backpointers
to recover

The probability of the
most probable tagged
word sequence

	$time_1$	$flies_2$	$fast_3$	
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	3.125E-5

Hidden Markov Models: decoding

The most probable sequence can be recovered by back-tracing the pointers starting with "final":

NN VB RB

Final:

$$v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$$

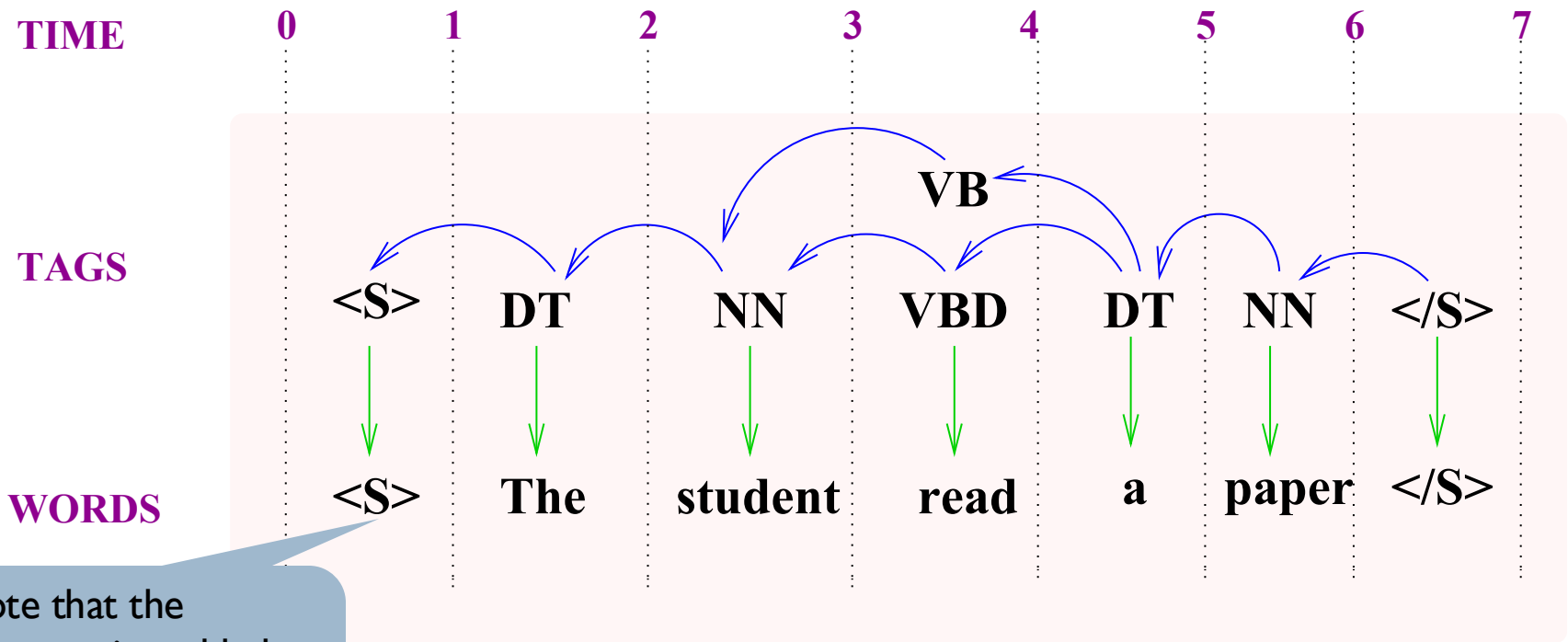
Retrace the
backpointers
to recover

The probability of the
most probable tagged
word sequence

	<i>time</i> ₁	<i>flies</i> ₂	<i>fast</i> ₃	
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5	-
STOP	-	-	-	3.125E-5

Viterbi Algorithm

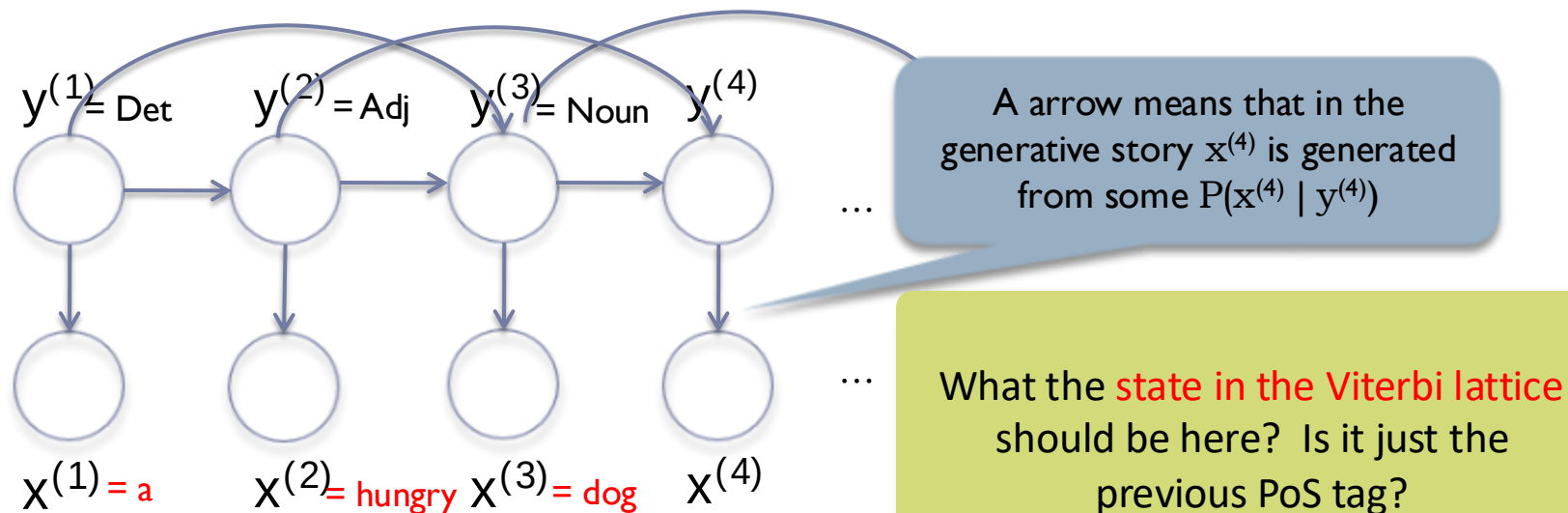
- ▶ A state **lattice** for a given sentence: we need to traverse only relevant states



Note that the sentence is padded with <s> and </s> symbols

2nd Order

Representation as an instantiation of a *graphical model*: N – tags, M – vocabulary size



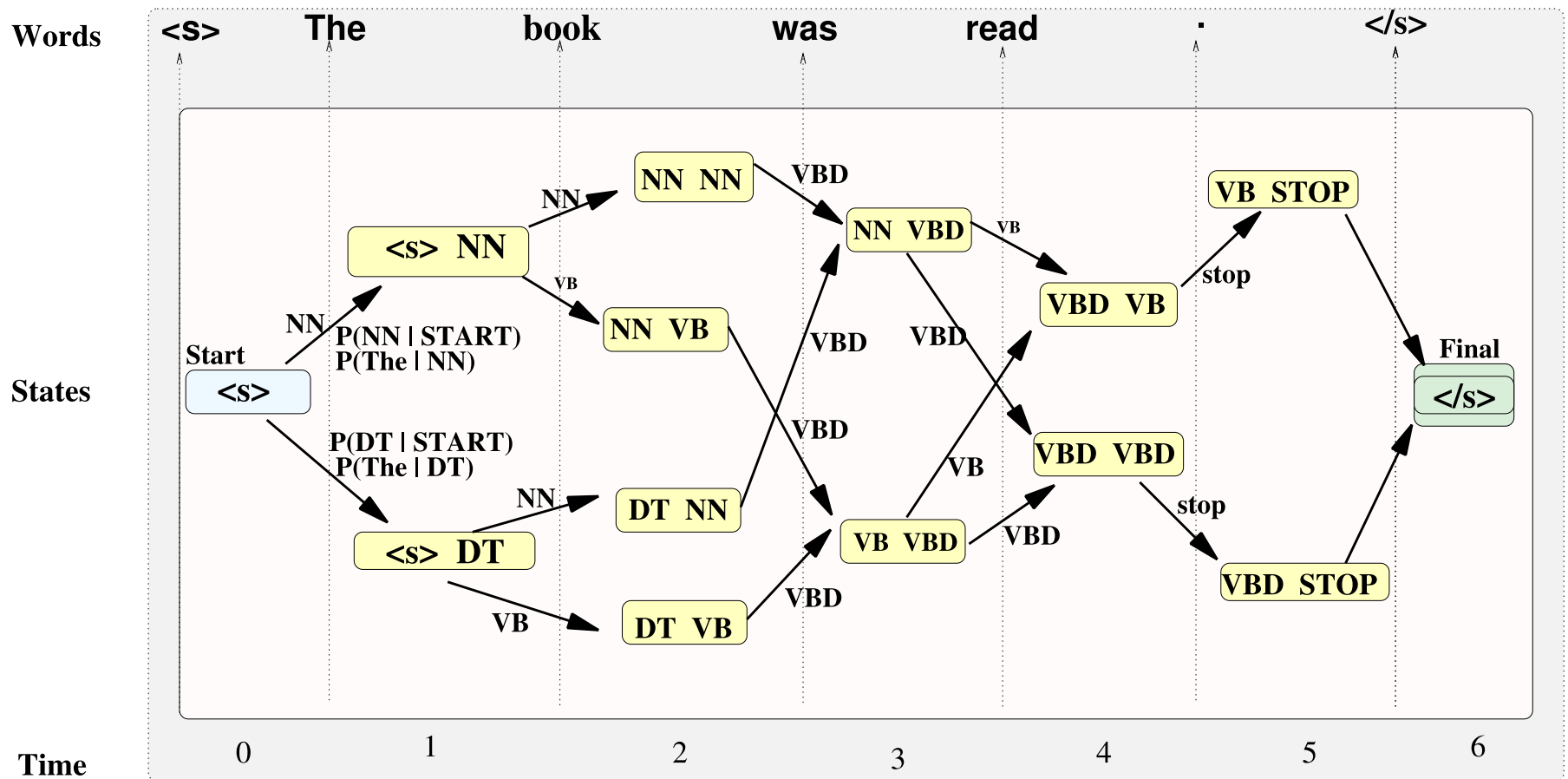
- ▶ HMM States correspond to POS tags,
- ▶ Words are emitted independently from each POS tag
- ▶ Parameters (to be estimated from the training set):

- ▶ Transition probabilities $P(y^t | y^{t-1}, y^{t-2})$: $[N \times N \times N]$ matrix
- ▶ Emission probabilities $P(x^t | y^t)$: $[N \times M]$ matrix

Stationarity assumption: this probability does not depend on the position in the sequence t

Viterbi Algorithm for 2nd order model

- ▶ A state in the lattice now include **both current and the previous state**



Hidden Markov Models: decoding

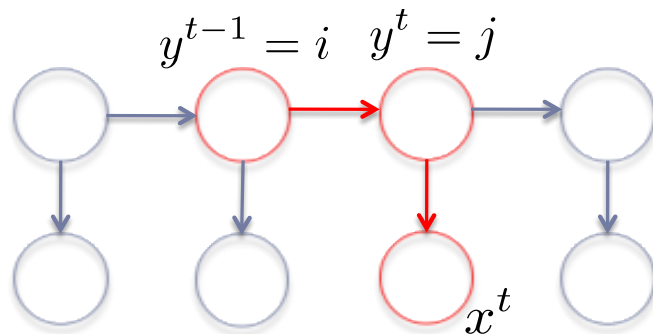
Initialization: $v_j^1 = a_{START,j} b_{j,x^1}, \quad j = 1, \dots, N;$

Recomputation: $v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

Final: $v_{STOP}^{|x|+1} = \max_i v_i^{|x|} a_{i,STOP}$

Equivalently, in the log-space (and in a more generalized form):

Define: $g^t(\mathbf{x}, i, j) = \begin{cases} \log a_{ij} + \log b_{j,x^t} & t = 1, \dots, |\mathbf{x}| \\ \log a_{ij} & t = |\mathbf{x}| + 1 \end{cases}$



The score is associated with a fragment responsible for one transition and one word generation

Initialization: $v_j^1 = g^1(\mathbf{x}, START, j) \quad j = 1, \dots, N;$

Recomputation: $v_j^t = \max_i (v_i^{t-1} + g^t(\mathbf{x}, i, j)), \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

Final: $v_{STOP}^{|x|+1} = \max_i (v_i^{|x|} + g^{|x|+1}(\mathbf{x}, i, STOP))$

Hidden Markov Models

- ▶ We will consider the part-of-speech (POS) tagging example

John	carried	a	tin	can	.
NNP	VBD	DT	NN	NN	.

- ▶ A “generative” model, i.e.:

✗ ▶ **Model:** Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$

✗ ▶ **Learning:** use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$

✗ ▶ **Decoding:** $\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

Is generative modeling necessarily the best approach to tagging? In the classification case, the Naïve Bayes was not necessarily the best choice

Hidden Markov Models: decoding

The score $g^t(\mathbf{x}, i, j)$ effectively tells us how likely a pair of states $(s_{t-1} = i, s_t = j)$ to appear for a sentence $\mathbf{x}$

Initialization: $v_j^1 = g^1(\mathbf{x}, START, j) \quad j = 1, \dots, N;$

Recomputation: $v_j^t = \max_i (v_i^{t-1} + g^t(\mathbf{x}, i, j)), \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \max_i (v_i^{|\mathbf{x}|} + g^{|\mathbf{x}|+1}(\mathbf{x}, i, STOP))$

What if we train a neural network to train to make this prediction?

Recap: pretraining / fine-tuning paradigm

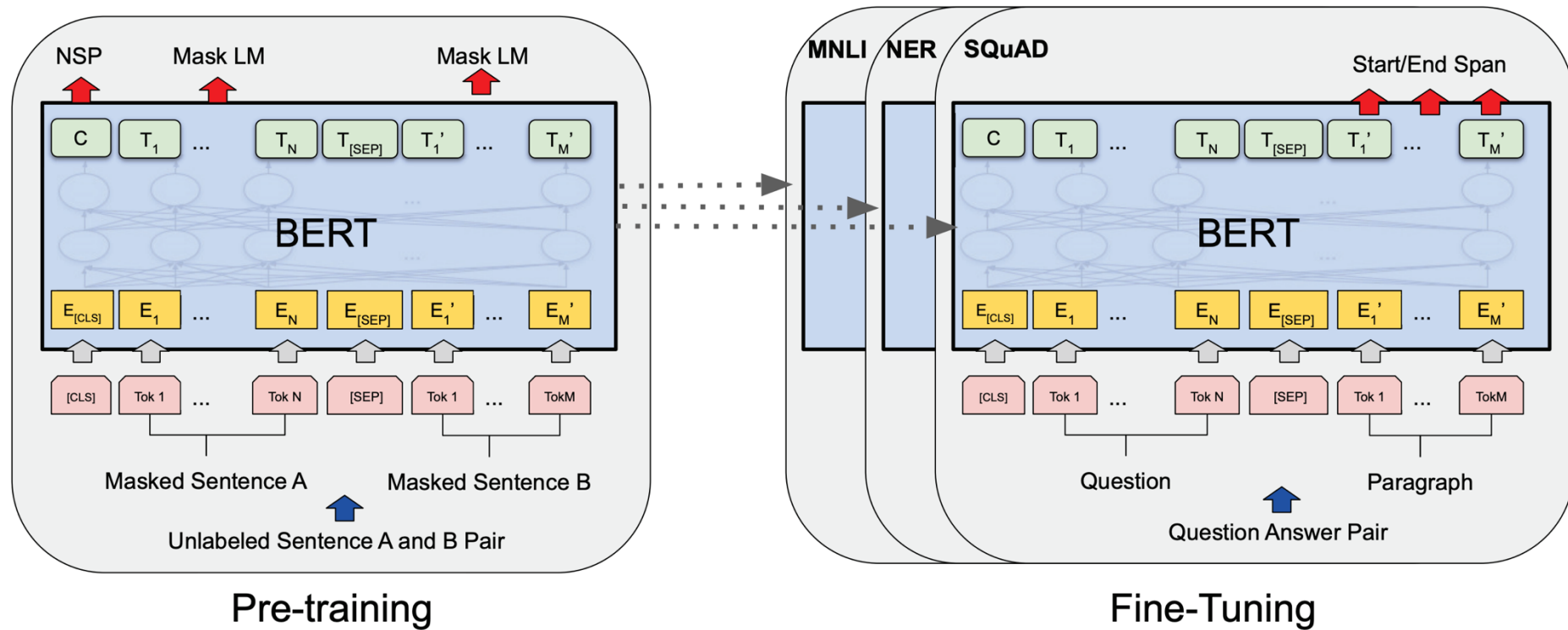
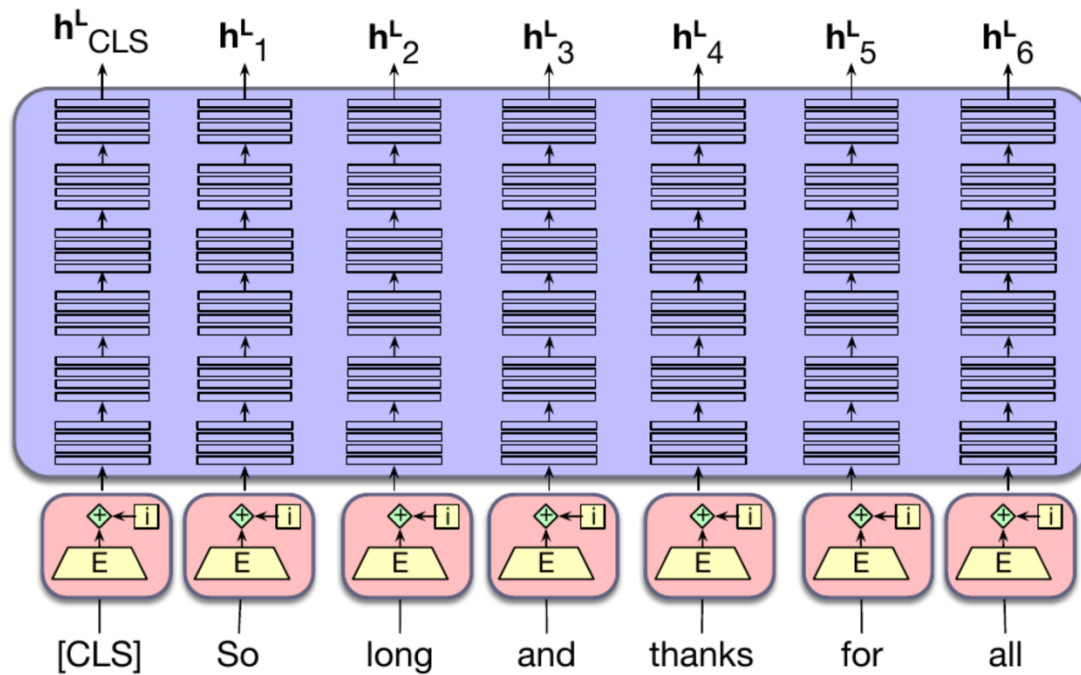


Figure from Devlin *et al.* (2019).

Recap: pretraining / fine-tuning paradigm

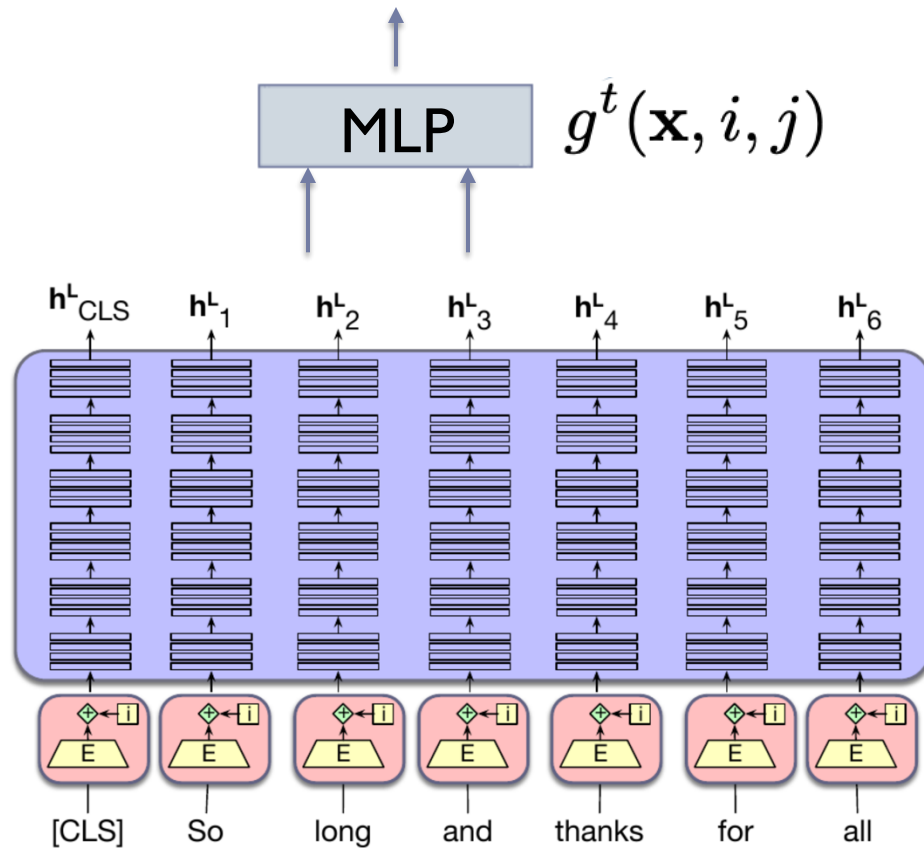


Neuralized sequence modeling

(ADJ, CC): -0.02

(VBZ, CC): -0.01

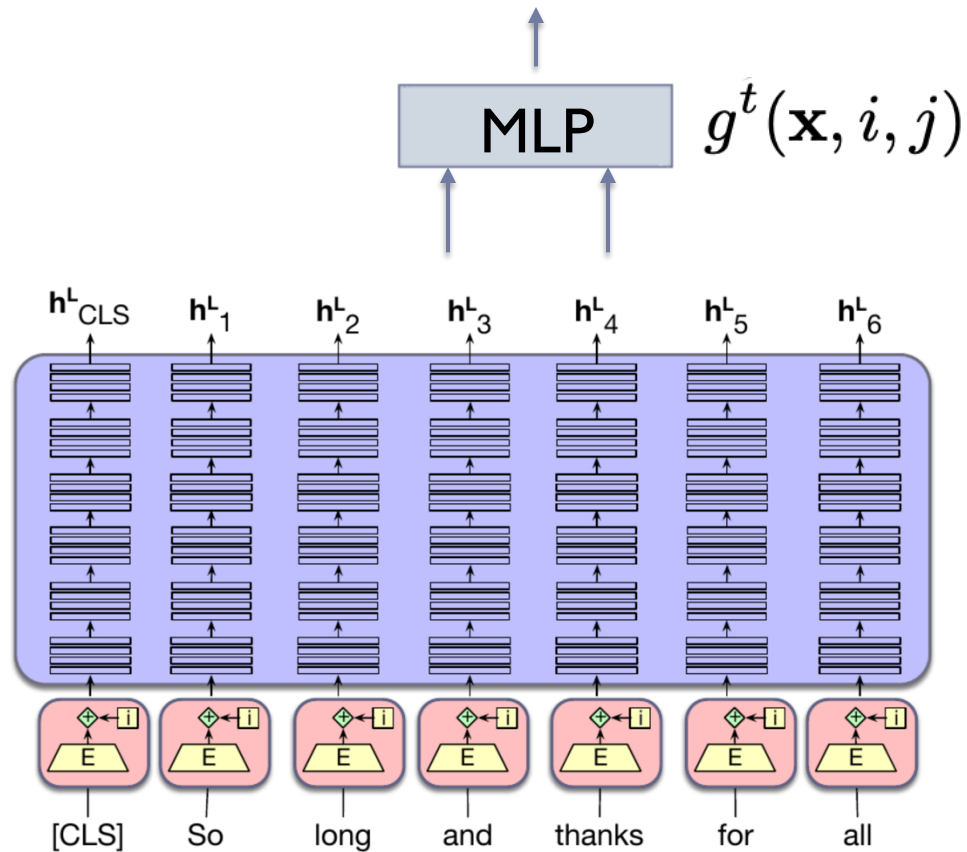
....



Neuralized sequence modeling

(CC, NN): -0.12

....



Recap: pretraining / fine-tuning paradigm

How do we train it?

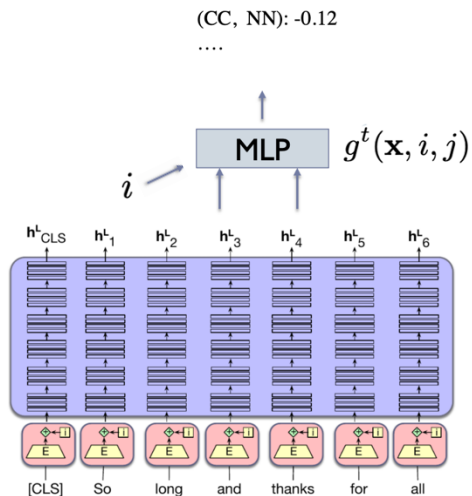
[Option I] **Local conditional estimation:**

$$g^t(\mathbf{x}, i, j) = \log P(y_t = j \mid y_{t-1} = i, \mathbf{x})$$

Simple cross-entropy, easy to train

Note:

- we **rely on any words** in the sentence
- **not generative** modeling anymore, not an HMM (but still can use Viterbi to do exact decoding)



Recap: pretraining / fine-tuning paradigm

[Option 2] Global model

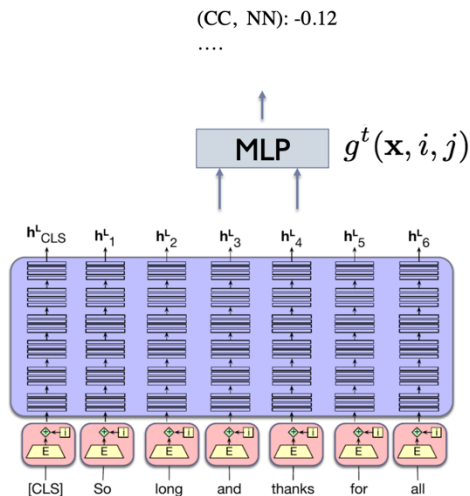
Markov Random Fields (more specifically Conditional Random Field, CRF)

Probability of the entire sequence of tags for a given sentence

Unnormalized score for the sequence $\mathbf{y}$

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{\exp \left(\sum_{t=1}^{|\mathbf{x}|} g^t(\mathbf{x}, y_{t-1}, y_t) \right)}{\sum_{\mathbf{y}'} \exp \left(\sum_{t=1}^{|\mathbf{x}|} g^t(\mathbf{x}, y'_{t-1}, y'_t) \right)}$$

Normalizer over all (exponential number) of tag sequences



Training: requires dynamic programming in training

Decoding: exactly the same (Viterbi)

Why use? has theoretical advantages (look up label bias / exposure bias if curious), often works better

Back to generative modeling – HMMs, what else?

- ▶ Using Viterbi, we can find the best tags for a sentence (decoding), and get $P(y, x|\theta)$
- ▶ We might also want to
 - ▶ compute the likelihood, i.e., the probability of a sentence regardless of its tags (a language model!) $P(x|\theta)$
 - ▶ learn the best set of parameters $\hat{\theta}$ given only an unannotated corpus of sentences.

Computing the likelihood

- ▶ From the probability theory we know

$$P(x|\theta) = \sum_y P(x, y|\theta)$$

- ▶ But there are an exponential number of sequences y
- ▶ Again, by computing and storing partial results, we can solve efficiently.

Viterbi

Initialization: $v_j^1 = a_{START,j} b_{j,x^1}, \quad j = 1, \dots, N;$

Recomputation: $v_j^t = \left(\max_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$

Forward algorithm

Initialization: $v_j^1 = a_{START,j} b_{j,x^1}, \quad j = 1, \dots, N;$

Recomputation: $v_j^t = \left(\sum_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \quad j = 1, \dots, N, \quad t = 2, \dots, |x|$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \sum_i v_i^{|\mathbf{x}|} a_{i,STOP}$

Summary

- ▶ HMM : a generative model of sentences using hidden state sequences
- ▶ Dynamic programming algorithms to compute
 - ▶ Best tag sequence given words (Viterbi algorithm)
 - ▶ Likelihood (forward algorithm)
- ▶ The model can be neutralized to integrate context
 - ▶ Model and estimation changes a bit
 - ▶ Decoding (Viterbi) remains