Part-Of-Speech Tagging (POS)

2/2565: FRA501 Introduction to Natural Language Processing with Deep learning

Week 03

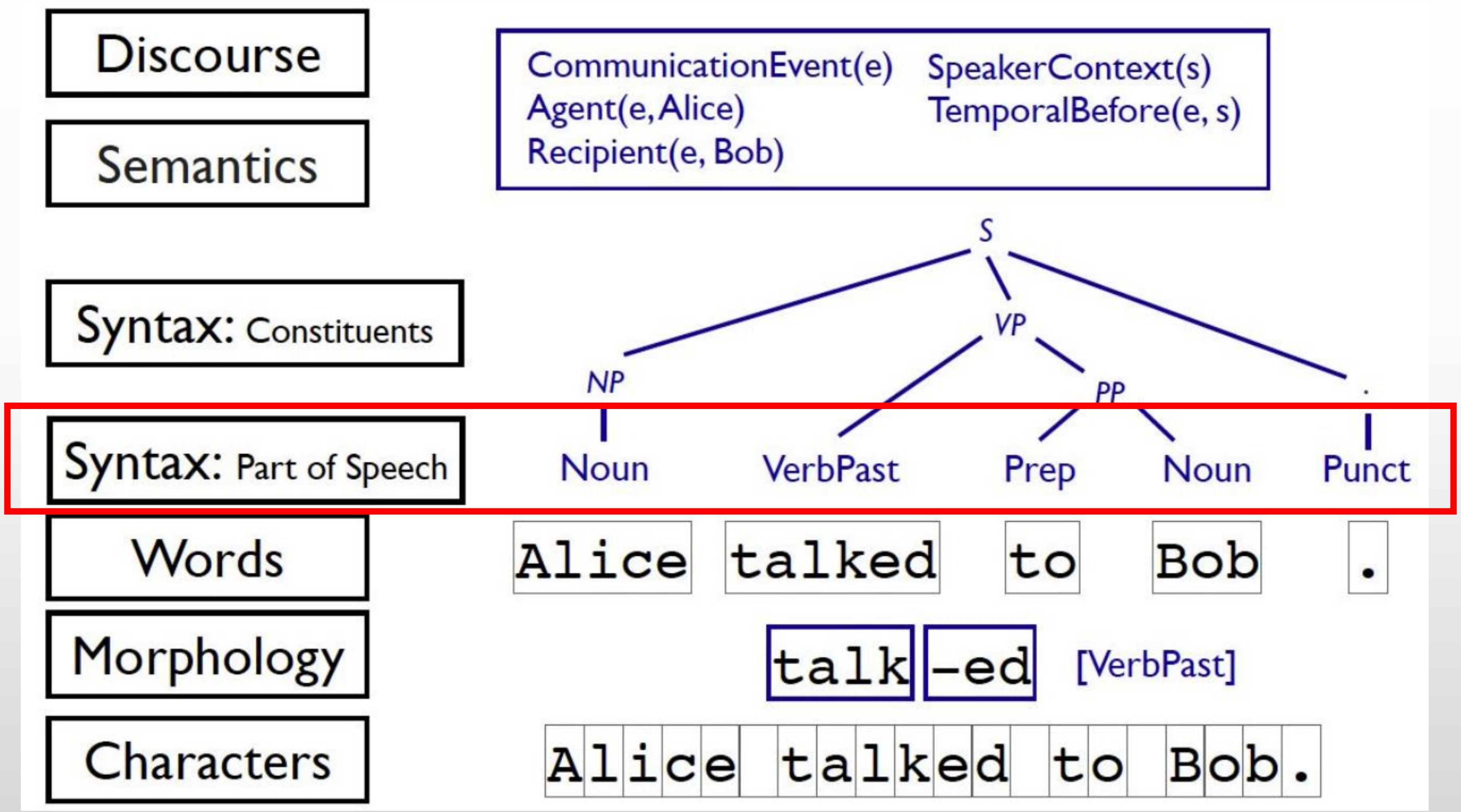
Paisit Khanarsa, Ph.D.

Institute of Field Robotics (FIBO), King Mongkut's University of Technology Thonburi

Outlines

- What is PoS?
- Traditional methods
  - Frequency-based
  - Local methods
  - Sequence methods
    - Hidden Markov Model (HMM)
    - Viterbi and beamsearch
    - Conditional Random Fields (CRF)
- Neural Network Methods
- Demo

What is Part of Speech(PoS)?



https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_natural_language_processing.htm

What is Part of Speech(PoS)? (cont.)

- Categorize words into similar grammatical properties (syntax)
  - Examples: Nouns, Verbs, Acjectives
- Actual applications often use more granular PoS labels

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

What is Part of Speech(PoS)? (cont.)

- Categorize words into similar grammatical properties (syntax)
  - Examples: Nouns, Verbs, Acjectives
- Actual applications often use more granular PoS labels

Input: They refuse to permit us to obtain the refuse permit.
Output: They/*PRP* refuse/*VBP* to/*To* permit/*VB* us/*PRP* to/*TO* obtain/*VB* the/*DT* refuse/*NN* permit/*NN*

https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

What is Part of Speech(PoS)? (cont.)

- Word disambiguation
  - Different word vectors for different PoS of the same words
- Text-to-speech
  - Different pronunciations based on PoS tag
- Helps other NLP tasks
- PoS provides additional information that helps other tasks
  - Tokenization
  - Name-Entity Recognition (NER)
  - Parsing

Traditional methods - Frequency-based

- Tag using most frequent tag for each word in the vocabulary
- If OOV (Out of vocabulary), tag using most frequent tag

Input: They refuse to permit us to obtain the refuse permit.
Output: They/*PRP* refuse/*VBP* to/*To* permit/*VB* us/*PRP* to/*TO* obtain/*VB* the/*DT* refuse /*VBP* permit /*VB*

Traditional methods - Local methods

- Decides the PoS tags using word and context feature
- PoS as a multiclass classification task
  - Logistic regression, decision tree, etc.
- Each PoS assignments are independent of each other
- Use local features
  - Word features
  - Context features

Method	Seen word accuracy	Unseen word accuracy
Most Frequent Tag	90%	50%
Log-linear Model with word features	93.7%	82.6%
Log-linear Model with context features	96.6%	86.8%

word feature	example
Prefixes	unfathomable: un- → adjective
Suffixes	surprisingly: -ly → adverb
Capitalization	Meridian: CAP → proper noun

Sequence methods

- Determining the PoS tag depends on the decision of the words around it
- Example

Input: They refuse to permit us to obtain the refuse permit.
Output: They/*PRP* refuse/*VBP* to/*To* permit/*VB* us/*PRP* to/*TO* obtain/*VB* the/*DT* refuse/*NN* permit/*NN*

Problem setup

- Sequence of words: $W \coloneqq \{w_1, w_2, w_3, \ldots, w_n\}$
- Sequence of tags: $T \coloneqq \{t_1, t_2, t_3, \ldots, t_n\}$
- Given W predict T

Problem setup (cont.)

- Sequence of words: $W \coloneqq \{w_1, w_2, w_3, \ldots, w_n\}$
- Sequence of tags: $T \coloneqq \{t_1, t_2, t_3, \ldots, t_n\}$
- Given W predict T

❖ $argmax_T P(T|W)$; Discriminative Model

❖ $argmax_T P(T|W) = argmax_T \frac{P(T,W)}{P(W)} \Rightarrow argmax_T P(T,W)$; Generative Model

$P(w)$ is constant

$P(a,b)$: joint distribution
$P(a|b)$: conditional distribution
$P(a) = \sum_b P(a,b)$:marginal distribution

Modeling $P(T, W)$

- What is the problem of modeling $P(T, W)$?
  - Sequence of words: $W \coloneqq \{w_1, w_2, w_3, \dots, w_n\}$
  - Sequence of tags: $T \coloneqq \{t_1, t_2, t_3, \dots, t_n\}$

$$P(T, W) = P(w_1, w_2, w_3, \dots, w_n, t_1, t_2, t_3, \dots, t_n)$$

Recap modeling distribution

- Example: Predicting PM2.5 level given wind in the morning.
- Find $argmax\ P(Y|X)$

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$

Day	X	Y
1	W	M
2	C	M
3	W	M
4	W	H
5	C	L
6	W	L
7	C	H
8	W	L

$P(X,Y)$	L	M	H
C			
W			

Joint distribution

$$P(X,Y) = \frac{Count(X,Y)}{Total\ count}$$

$P(Y\|X)$	L	M	H
C			
W			

Conditional distribution

$$P(X|Y) = \frac{Count(X,Y)}{Total\ count\ X}$$

Recap modeling distribution (cont.)

- Example: Predicting PM2.5 level given wind in the morning.
- Find $argmax\ P(Y|X)$

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$

Day	X	Y
1	W	M
2	C	M
3	W	M
4	W	H
5	C	L
6	W	L
7	C	H
8	W	L

$Count(X,Y)$	L	M	H
C	1	1	1
W	2	2	1

$P(X,Y)$	L	M	H
C			
W			

Joint distribution

$$P(X,Y) = \frac{Count(X,Y)}{Total\ count}$$

$P(Y\|X)$	L	M	H
C			
W			

Conditional distribution

$$P(X|Y) = \frac{Count(X,Y)}{Total\ count\ X}$$

Recap modeling distribution (cont.)

- Example: Predicting PM2.5 level given wind in the morning.
- Find $argmax\ P(Y|X)$

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$

Day	X	Y
1	W	M
2	C	M
3	W	M
4	W	H
5	C	L
6	W	L
7	C	H
8	W	L

$Count(X,Y)$	L	M	H
C	1	1	1
W	2	2	1

$P(X,Y)$	L	M	H
C	1/8	1/8	1/8
W	2/8	2/8	1/8

Joint distribution

$$P(X,Y) = \frac{Count(X,Y)}{Total\ count}$$

$P(Y|X)$	L	M	H
C			
W			

Conditional distribution

$$P(X|Y) = \frac{Count(X,Y)}{Total\ count\ X}$$

Recap modeling distribution (cont.)

- Example: Predicting PM2.5 level given wind in the morning.
- Find $argmax\ P(Y|X)$

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$

Day	X	Y
1	W	M
2	C	M
3	W	M
4	W	H
5	C	L
6	W	L
7	C	H
8	W	L

$Count(X,Y)$	L	M	H
C	1	1	1
W	2	2	1

$P(X,Y)$	L	M	H
C	1/8	1/8	1/8
W	2/8	2/8	1/8

Joint distribution

$$P(X,Y) = \frac{Count(X,Y)}{Total\ count}$$

$P(Y\|X)$	L	M	H
C	1/3	1/3	1/3
W	2/5	2/5	1/5

Conditional distribution

$$P(X|Y) = \frac{Count(X,Y)}{Total\ count\ X}$$

Curse of dimensionality

- Example: Predicting PM2.5 level given wind in the morning and evening.
- Find $argmax\ P(Y,Z|X)$

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$
$Z \in \{Low, Med, High\}$

Day	X	Z	Y
1	W	L	M
2	C	M	M
3	W	H	M
4	W	M	H
5	C	H	L
6	W	M	L
7	C	L	H
8	W	H	L

$P(Y,Z\|X=C)$	Z=L	Z=M	Z=H
Y=L			
Y=M			
Y=H			

$P(Y,Z\|X=W)$	Z=L	Z=M	Z=H
Y=L			
Y=M			
Y=H			

Curse of dimensionality (cont.)

- Example: Predicting PM2.5 level given wind in the morning and evening.
- Find $argmax\ P(Y, Z|X)$

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$
$Z \in \{Low, Med, High\}$

Day	X	Z	Y
1	W	L	M
2	C	M	M
3	W	H	M
4	W	M	H
5	C	H	L
6	W	M	L
7	C	L	H
8	W	H	L

$P(Y, Z\|X = C)$	Z=L	Z=M	Z=H
Y=L	0	0	1/3
Y=M	0	1/3	0
Y=H	1/3	0	0

$P(Y, Z\|X = W)$	Z=L	Z=M	Z=H
Y=L	0	1/5	1/5
Y=M	1/5	0	1/5
Y=H	0	1/5	0

Curse of dimensionality (cont.)

- Example: Predicting PM2.5 level given wind in the morning and evening.
- Find $argmax\ \boxed{P(Y,Z|X) = P(Z|X)P(Y|X)\ \ ;\ \ Z \perp Y \mid X}$

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$
$Z \in \{Low, Med, High\}$

Day	X	Z	Y
1	W	L	M
2	C	M	M
3	W	H	M
4	W	M	H
5	C	H	L
6	W	M	L
7	C	L	H
8	W	H	L

$P(Y\|X)$	L	M	H
C			
W			

$P(Z\|X)$	L	M	H
C			
W			

Curse of dimensionality (cont.)

- Example: Predicting PM2.5 level given wind in the morning and evening.
- Find $argmax\ \boxed{P(Y,Z|X) = P(Z|X)P(Y|X)\ \ ;\ \ \ Z \perp Y \mid X}$

$X\ \in \{Calm, Windy\}$
$Y\ \in \{Low, Med, High\}$
$Z\ \in \{Low, Med, High\}$

Day	X	Z	Y
1	W	L	M
2	C	M	M
8	W	H	L

$P(Y\|X)$	L	M	H
C			
W			

$$P(T,W) = P(w_1, w_2, w_3, \dots, w_n, t_1, t_2, t_3, \dots, t_n)$$

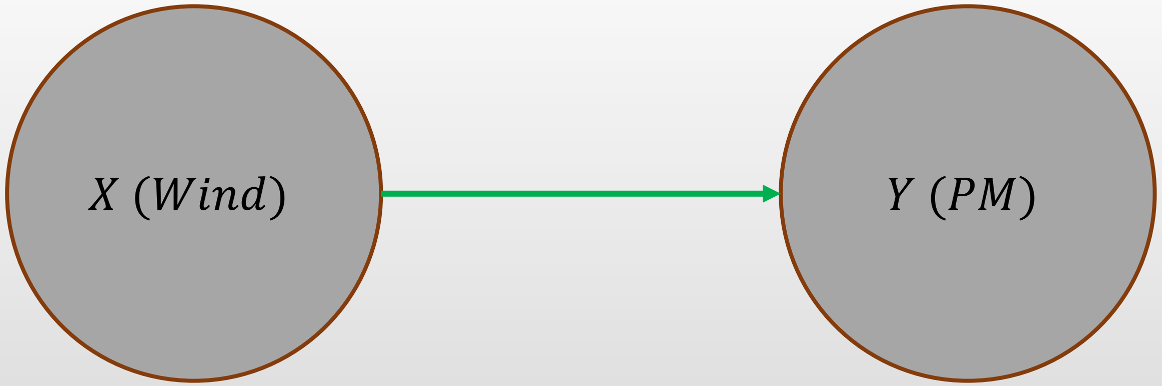
Problems: Curse of dimensionality

We use Markov assumptions

Dependence as graphical models

- In probabilistic graphical models, we can draw the relationship between random variables using graph notations
  - A node is a variable
  - An arrow is the relationship between the variables

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$

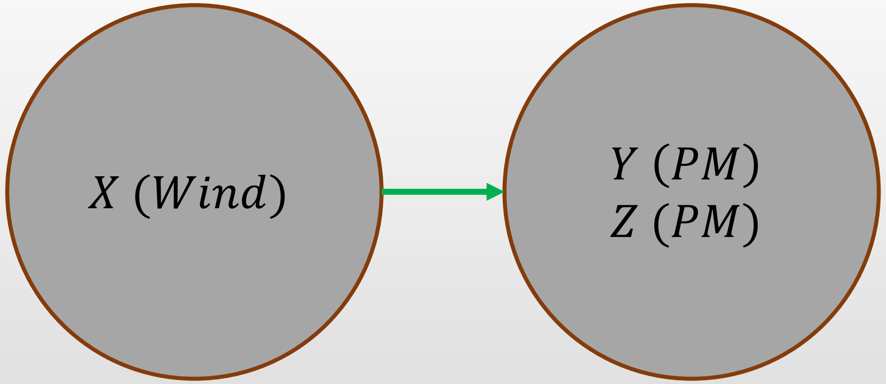


$$P(X,Y) = P(X)P(Y|X)$$

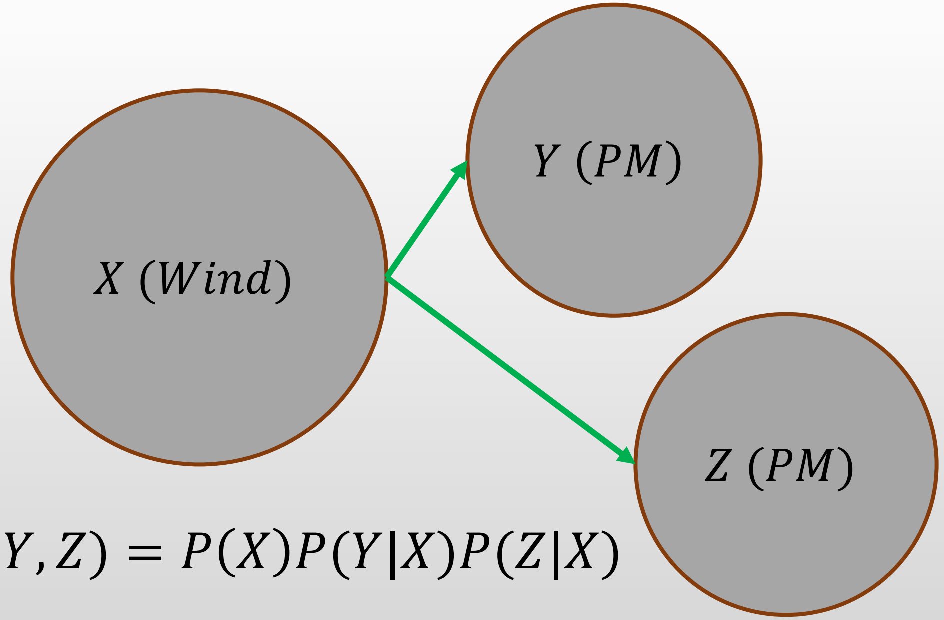
Dependence as graphical models (cont.)

- In probabilistic graphical models, we can draw the relationship between random variables using graph notations
  - A node is a variable
  - An arrow is the relationship between the variables

$X \in \{Calm, Windy\}$
$Y \in \{Low, Med, High\}$
$Z \in \{Low, Med, High\}$

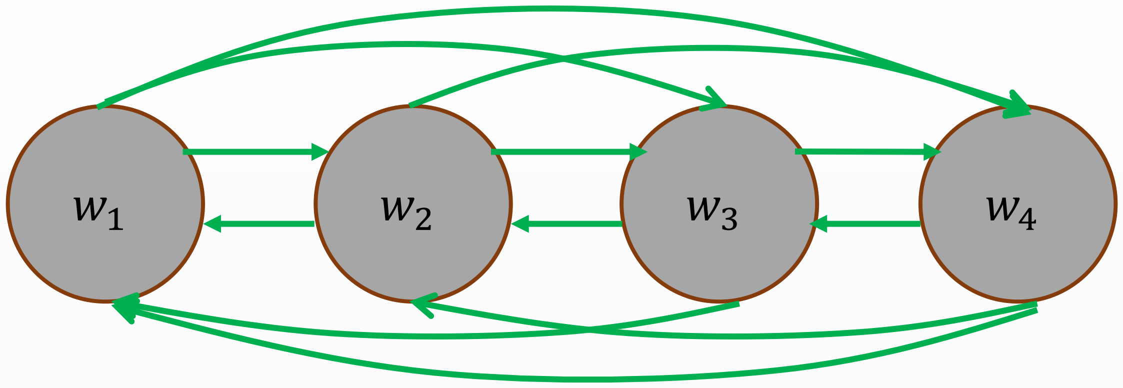


$$P(X,Y,Z) = P(X)P(Y,Z|X)$$

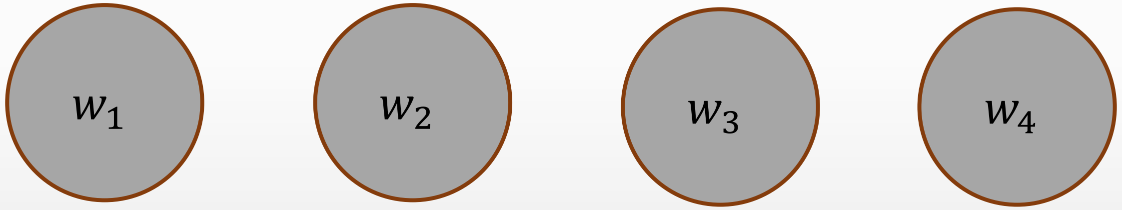


$$P(X,Y,Z) = P(X)P(Y|X)P(Z|X)$$

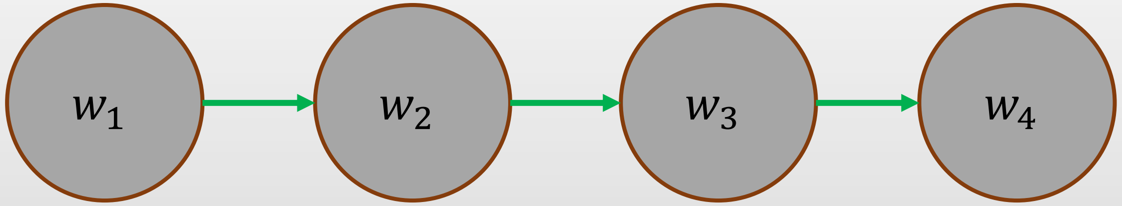
Dependence as graphical models (cont.)



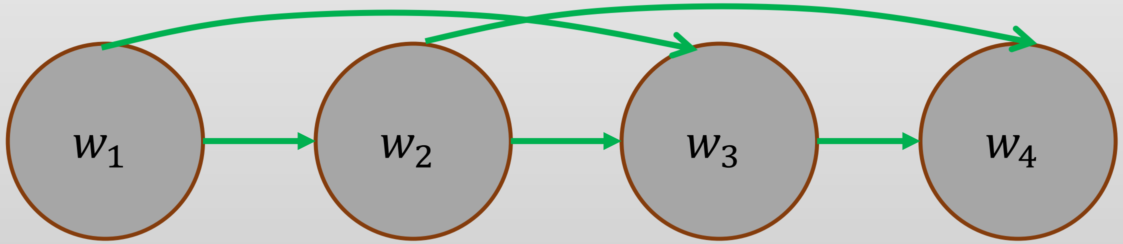
$P(w_1, w_2, w_3, w_4)$; Full dependency



$P(w_1, w_2, w_3, w_4) = P(w_1)P(w_2)P(w_3)P(w_4)$
; Unigrams



$P(w_1, w_2, w_3, w_4) = P(w_1)P(w_2|w_1)P(w_3|w_2)P(w_4|w_3)$
; Bigrams



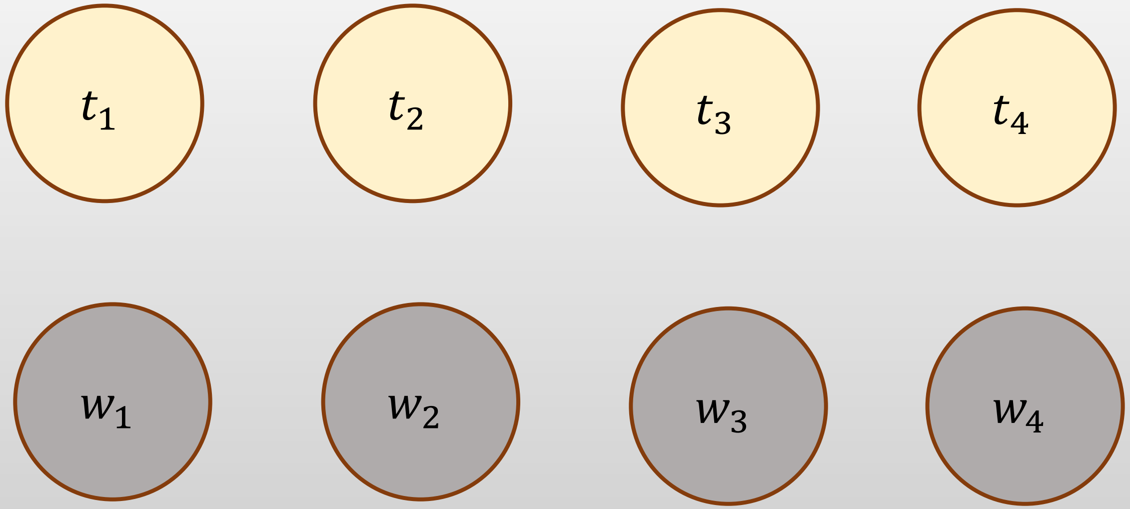
$P(w_1, w_2, w_3, w_4) = P(w_1)P(w_2|w_1)P(w_3|w_2w_1)P(w_4|w_3w_2)$
; Trigrams

Latent variables

- **Dark** colors the nodes for the variables that observe from data (known values)
- **Light** color the nodes for the variables that do not know from data (unknown values, latent values)

$$\boxed{P(T, W) = P(w_1, w_2, w_3, \dots, w_n, t_1, t_2, t_3, \dots, t_n)}$$

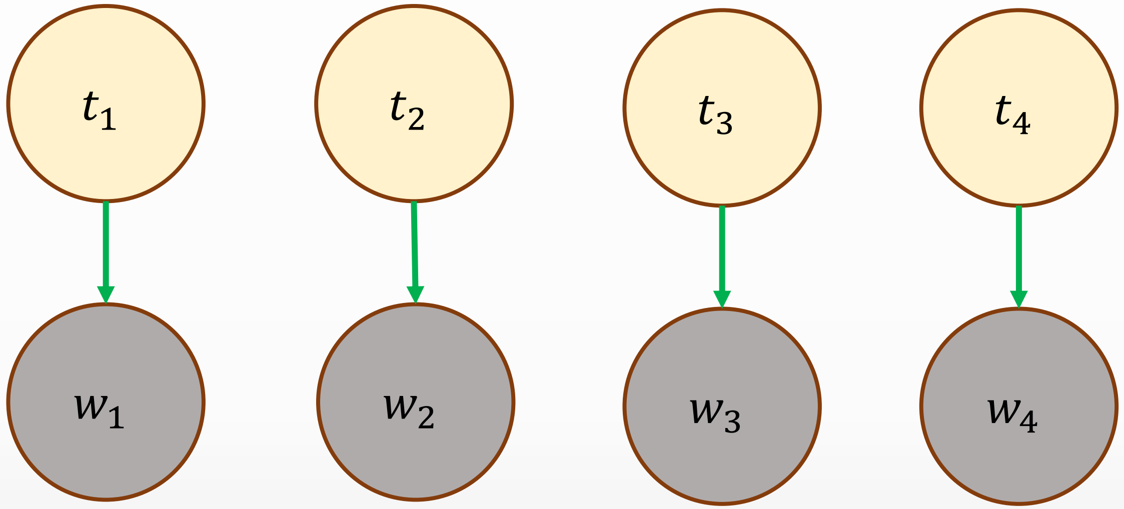
Sequence of words: $W \coloneqq \{w_1, w_2, w_3, \dots, w_n\}$
Sequence of tags: $T \coloneqq \{t_1, t_2, t_3, \dots, t_n\}$



Local tagging

Sequence of words: $W \coloneqq \{w_1, w_2, w_3, \dots, w_n\}$
Sequence of tags: $T \coloneqq \{t_1, t_2, t_3, \dots, t_n\}$
$n = 4$

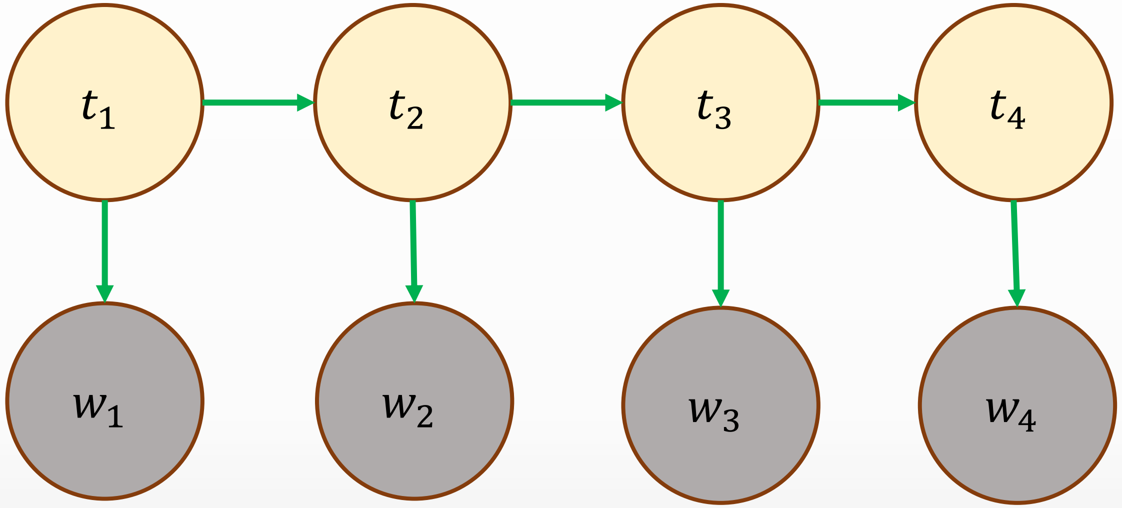
$$
\begin{aligned}
P(T, W) &= P(w_1, w_2, w_3, \dots, w_n, t_1, t_2, t_3, \dots, t_n) \\
&= P(w_1, w_2, w_3, w_4, t_1, t_2, t_3, t_4) \\
&= P(w_1|t_1)P(w_2|t_2)\, P(w_3|t_3)\, P(w_4|t_4)
\end{aligned}
$$



Local tagging (cont.)

Sequence of words: $W \coloneqq \{w_1, w_2, w_3, \dots, w_n\}$
Sequence of tags: $T \coloneqq \{t_1, t_2, t_3, \dots, t_n\}$
$n = 4$

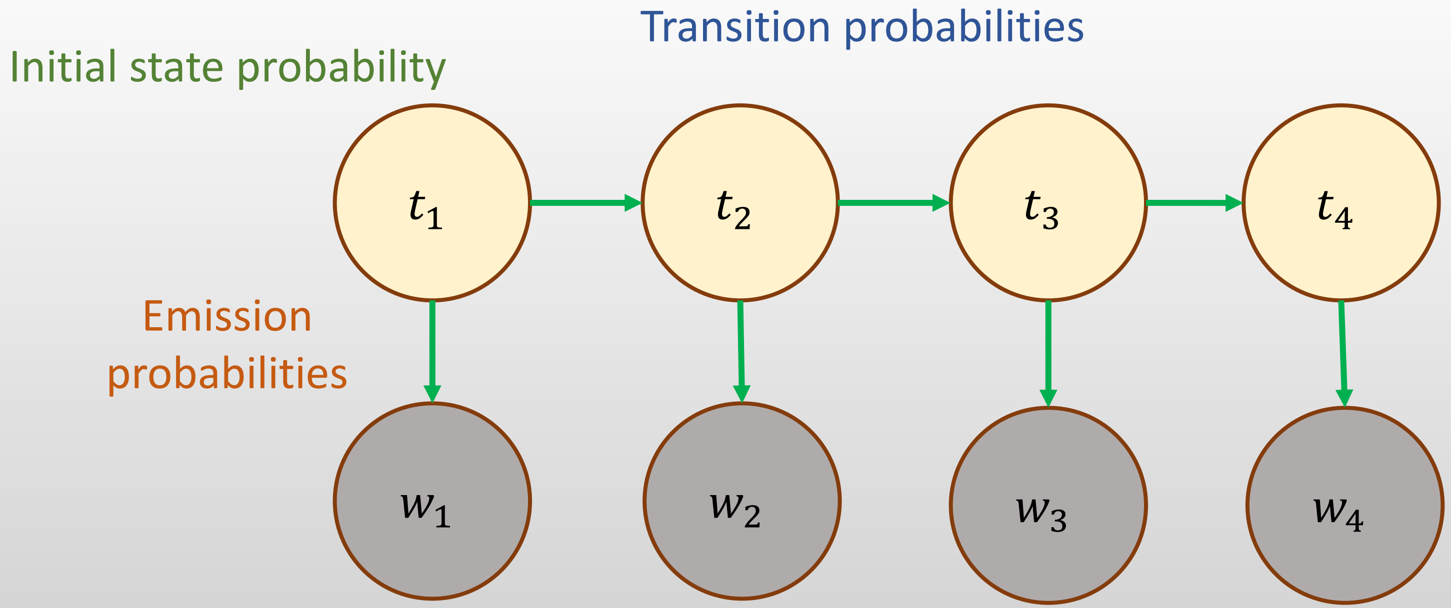
$$
\begin{aligned}
P(T, W) &= P(w_1, w_2, w_3, \dots, w_n, t_1, t_2, t_3, \dots, t_n) \\
&= P(w_1, w_2, w_3, w_4, t_1, t_2, t_3, t_4) \\
&= P(w_1|t_1)P(w_2|t_2)\,P(w_3|t_3)\,P(w_4|t_4)
\end{aligned}
$$



PoS is the sequence dependence
Solution: Hidden Markov Model (HMM)

Hidden Markov Model

- Markov assumption
  - Current value only depends immediate pass.
- $P(T,W) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)P(w_1|t_1)P(w_2|t_2)P(w_3|t_3)P(w_4|t_4)$

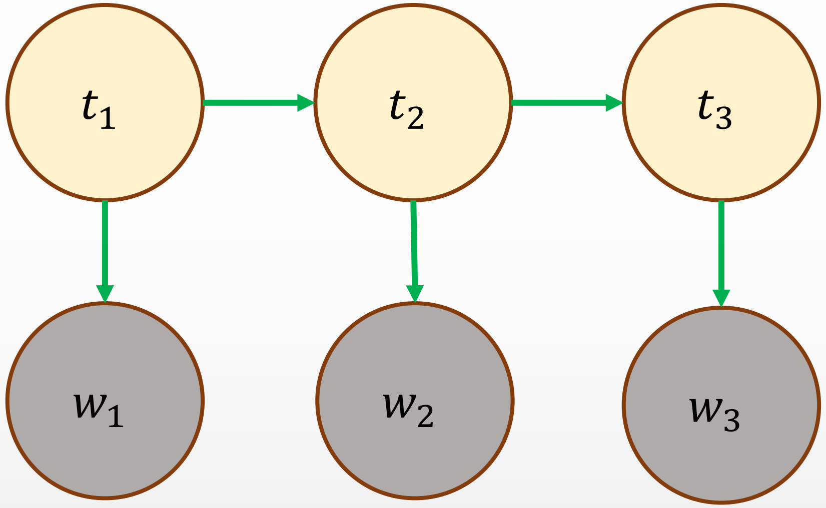


Hidden Markov Model (cont.)

- Example

N = Noun
NN = Noun
i, j = state (tag) index
k = state (word) index

$t \in$ {N,NN}, w $\in$ {I,eat,chinese}
A_{ij} = Transition probability
B_{ij} = Emission probability
Starting state probability p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

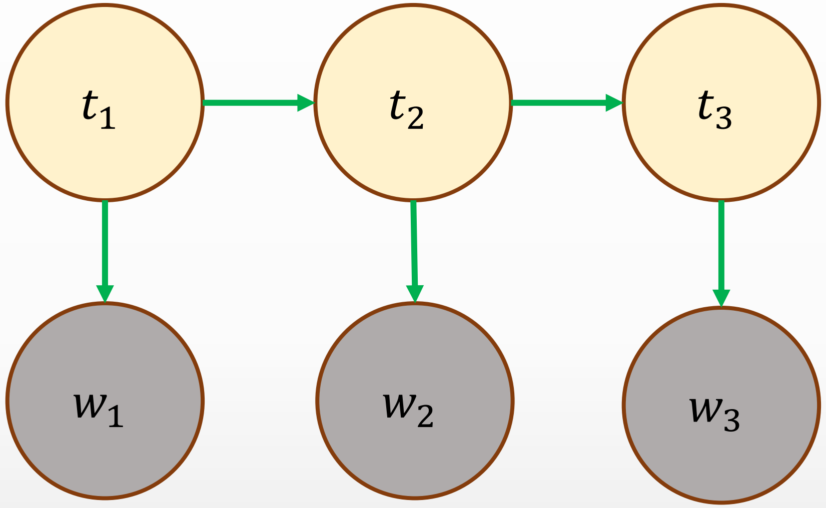
Question: What's the probability of $P([N\ NN\ N],[I\ eat\ chinese])$?
Ans: $P([N\ NN\ N],[I\ eat\ chinese]) \ =$

Hidden Markov Model (cont.)

- Example

N = Noun
NN = Noun
i, j = state (tag) index
k = state (word) index

$t \in \{N, NN\}$, $w \in \{I, eat, chinese\}$
A_{ij} = Transition probability
B_{ij} = Emission probability
Starting state probability p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

Question: What's the probability of $P([N\ NN\ N], [I\ eat\ chinese])$?

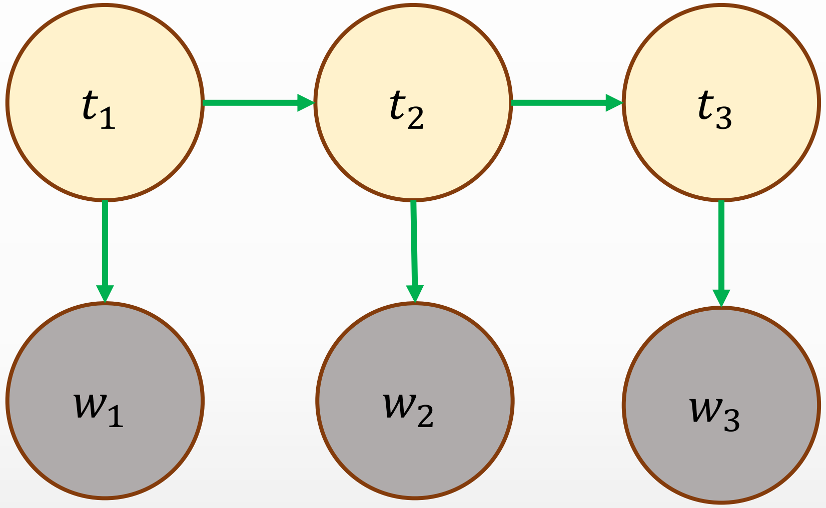
Ans: $P([N\ NN\ N], [I\ eat\ chinese]) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(w_1|t_1)P(w_2|t_2)P(w_3|t_3)$

Hidden Markov Model (cont.)

- Example

N = Noun
NN = Noun
i, j = state (tag) index
k = state (word) index

t ∈ {N,NN}, w ∈ {I,eat,chinese}
A_{ij} = Transition probability
B_{ij} = Emission probability
Starting state probability p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

Question: What's the probability of $P([N\ NN\ N], [I\ eat\ chinese])$?

Ans: $P([N\ NN\ N], [I\ eat\ chinese]) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(w_1|t_1)P(w_2|t_2)P(w_3|t_3)$

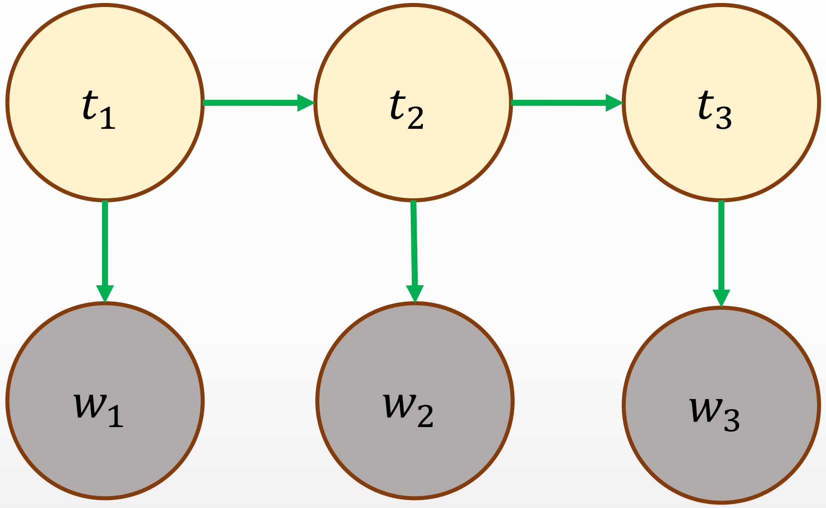
$$= p_0(N) \cdot A_{12} \cdot A_{21} \cdot B_{11} \cdot B_{22} \cdot B_{13}$$

Hidden Markov Model (cont.)

- Example

N = Noun
NN = Noun
i, j = state (tag) index
k = state (word) index

t ∈ {N,NN}, w ∈ {I,eat,chinese}
A_{ij} = Transition probability
B_{ij} = Emission probability
Starting state probability p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

Question: What's the probability of $P([N\ NN\ N],[I\ eat\ chinese])$?

Ans: $P([N\ NN\ N],[I\ eat\ chinese]) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(w_1|t_1)P(w_2|t_2)P(w_3|t_3)$
$= p_0(N) \cdot A_{12} \cdot A_{21} \cdot B_{11} \cdot B_{22} \cdot B_{13} = 0.7 \cdot 0.4 \cdot 0.5 \cdot 0.8 \cdot 0.45 \cdot 0.19$

Hidden Markov Model (cont.)

- How to estimate A_{ij} and B_{ij}
- $P(A_{11}) = \frac{Count(from\ \mathrm{N}\ to\ \mathrm{NN})}{Count(\mathrm{N})}$
- $P(B_{11}) = \frac{Count(\text{"I"},\mathrm{N})}{Count(\mathrm{N})}$

Transition probabilities

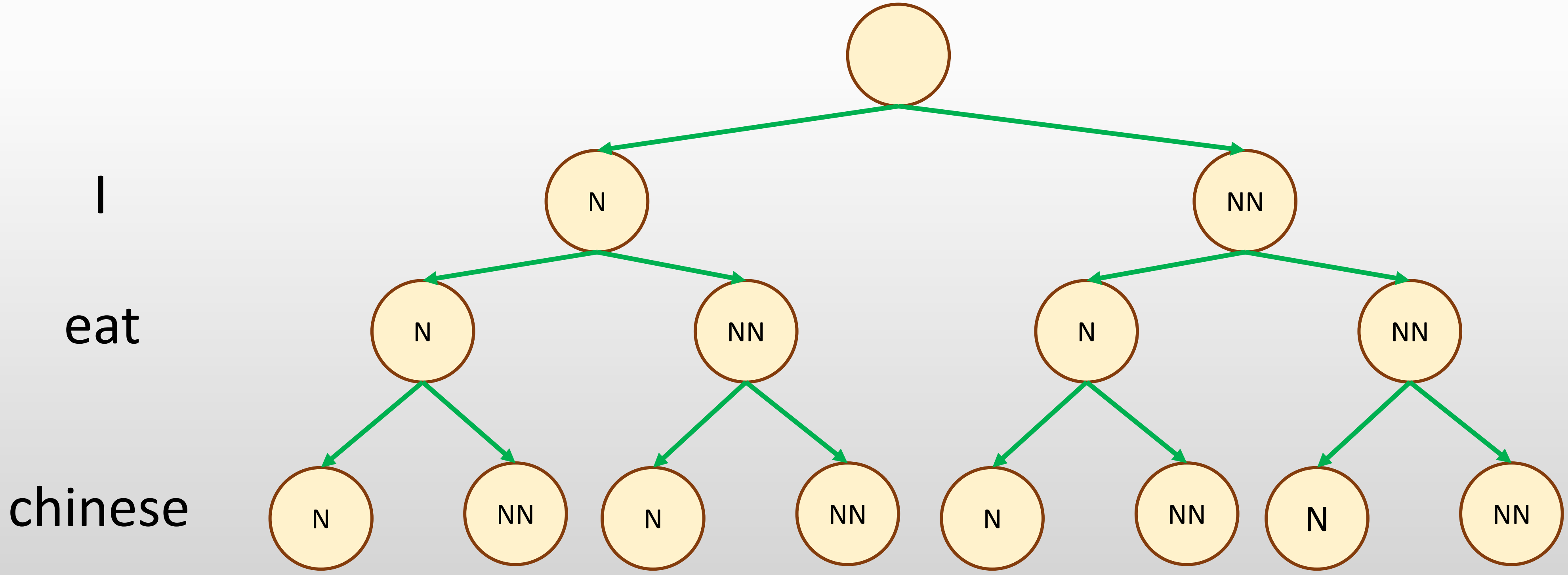
A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

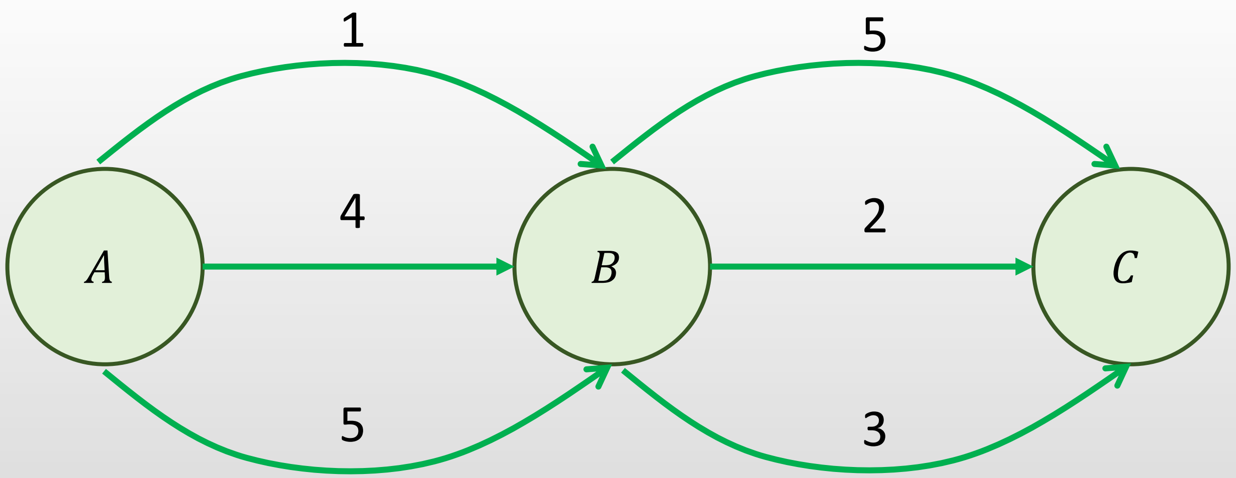
Search in Hidden Markov Model

- Recall we want to find the sequence of tags that maximizes the joint probability: $argmax_T P(T, W)$
- **Brute force:** Find all possible sequence of T, calculate $P(T, W)$ and compare



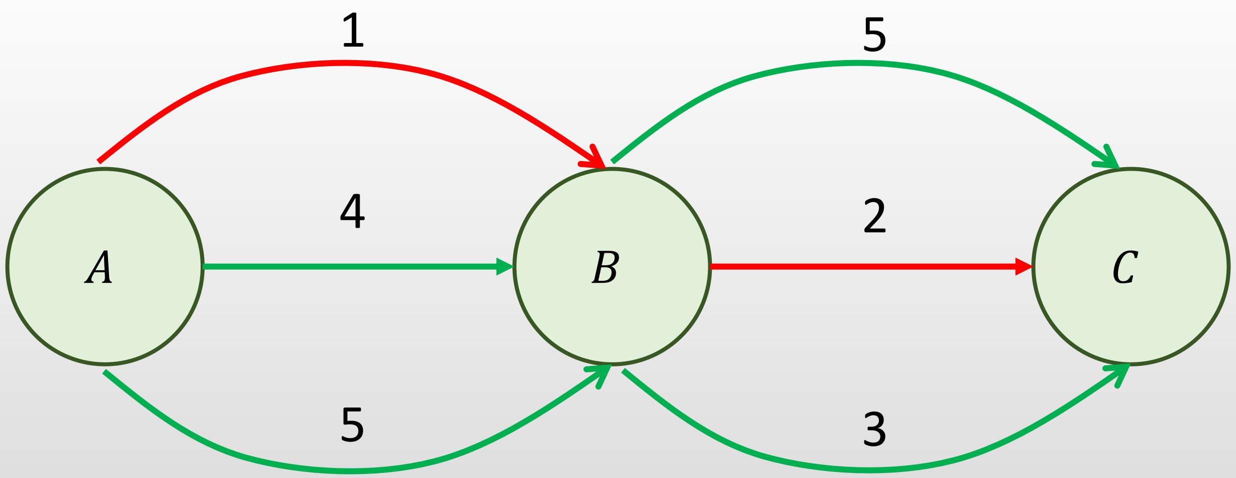
The Viterbi Algorithm

- Some computation are redundant, we can save computation from previous steps.
- Example: Find the shortest route from A to C



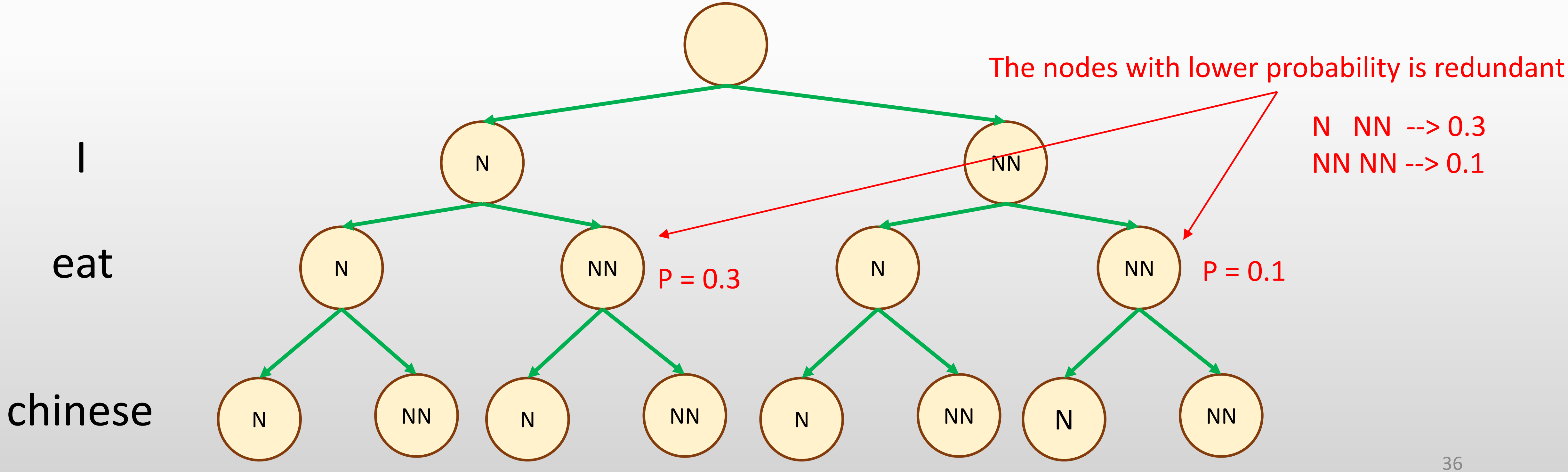
The Viterbi Algorithm (cont.)

- Some computation are redundant, we can save computation from previous steps.
- Example: Find the shortest route from A to C



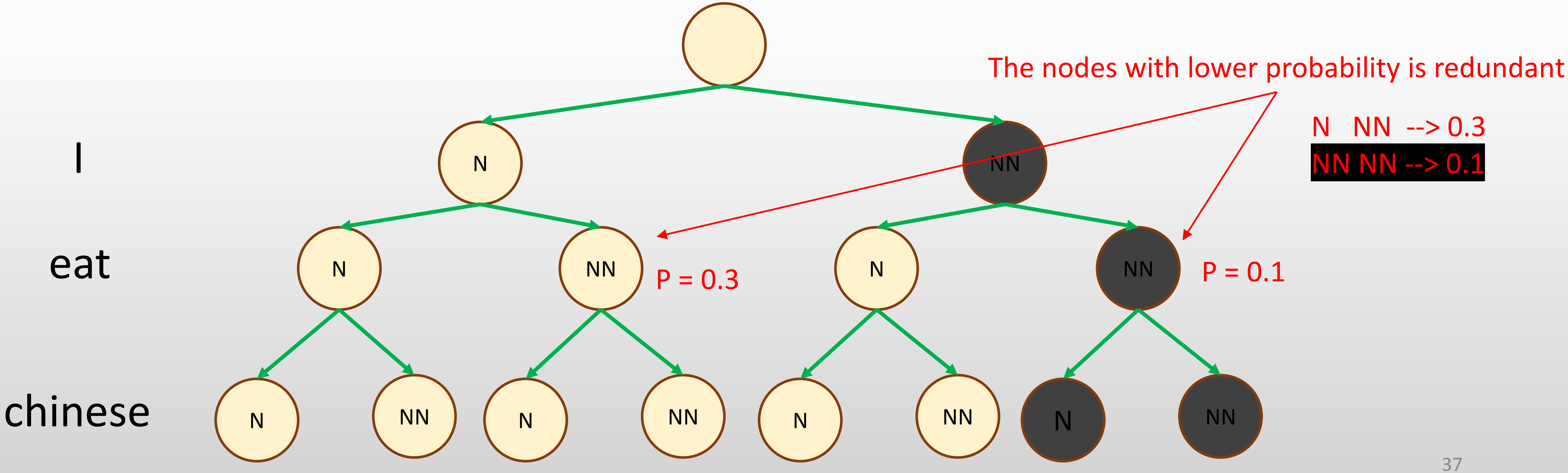
The Viterbi Algorithm (cont.)

- Some computation are redundant, we can save computation from previous steps.



The Viterbi Algorithm (cont.)

- Some computation are redundant, we can save computation from previous steps.



The Viterbi Algorithm (cont.)

- Creates two matrices
  - $\pi[i, t]$ saves the best probability at word position i for hidden state t.
  - $\mathrm{B}[i, t]$ saves the previous hidden state that maximize this current state probability

Base Step:

$$\pi[0, <S>] = \log 1 = 0$$
$$\pi[0, t] = \log 0 = -\infty,\ if\ t \neq <S>$$

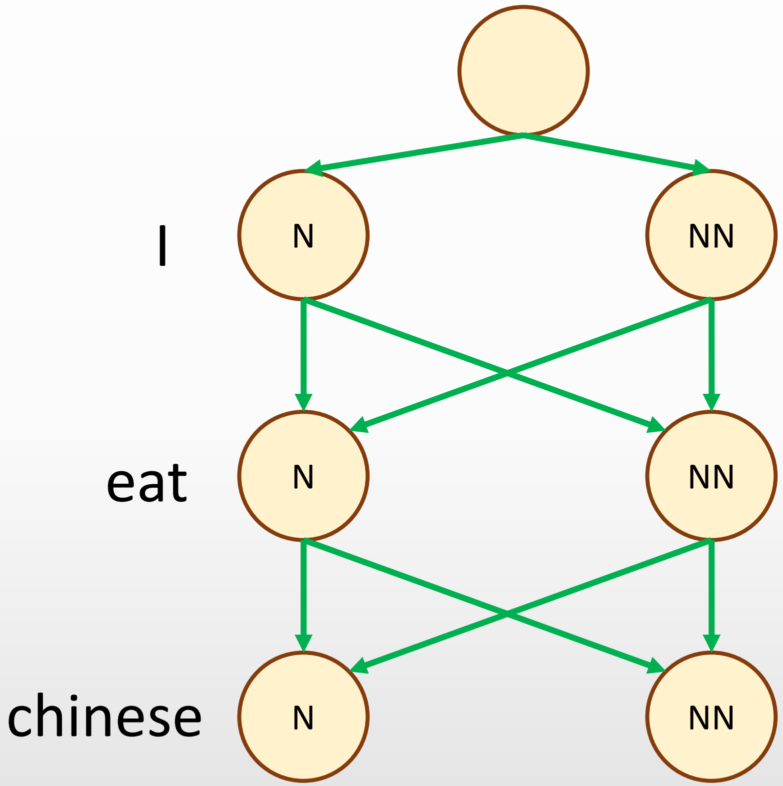
where $< S >$ is the start symbol.

Recursive Step:

$$\pi[i, t] = \max_{t'}\{\pi[i-1, t'] + logP(t|t') + logP(w_i|t)\}$$

The Viterbi Algorithm (cont.)

p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

$\pi[i,t]$	I	eat	chinese
State N			
State NN			

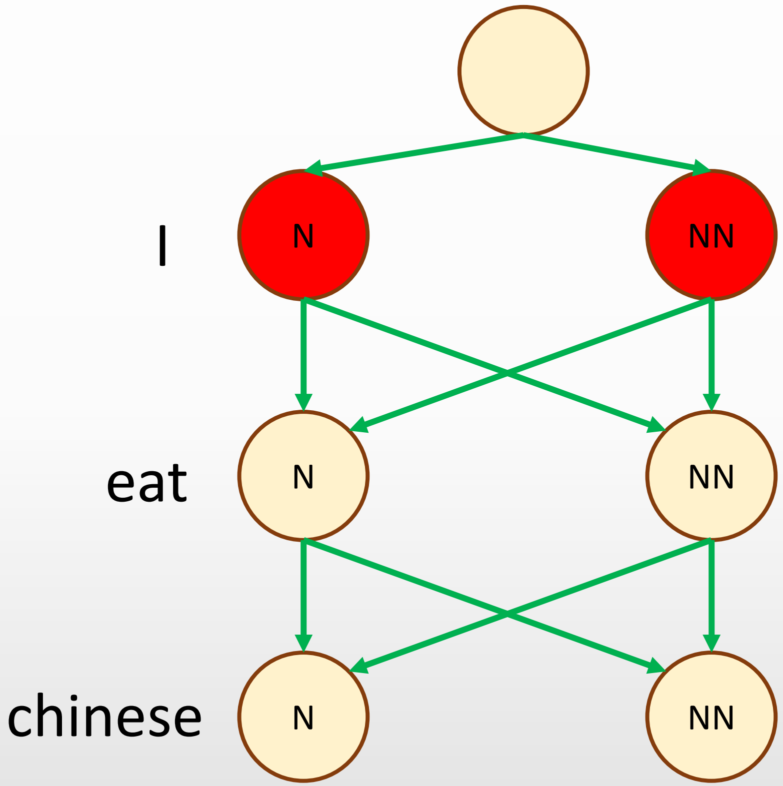
B$[i,t]$	I	eat	chinese
State N	-		
State NN	-		

Recursive Step:

$$\pi[i,t] = \max_{t'}\{\pi[i-1,t'] + logP(t|t') + logP(w_i|t)\}$$

The Viterbi Algorithm (cont.)

p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

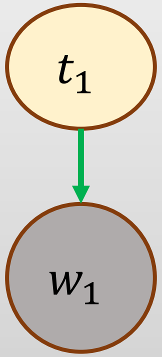
B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

$\pi[i,t]$	I	eat	chinese
State N	0.56		
State NN	0.03		

$\mathrm{B}[i,t]$	I	eat	chinese
State N	-		
State NN	-		

Recursive Step:

$$\pi[i,t] = \max_{t'}\{\pi[i-1,t'] + logP(t|t') + logP(w_i|t)\}$$

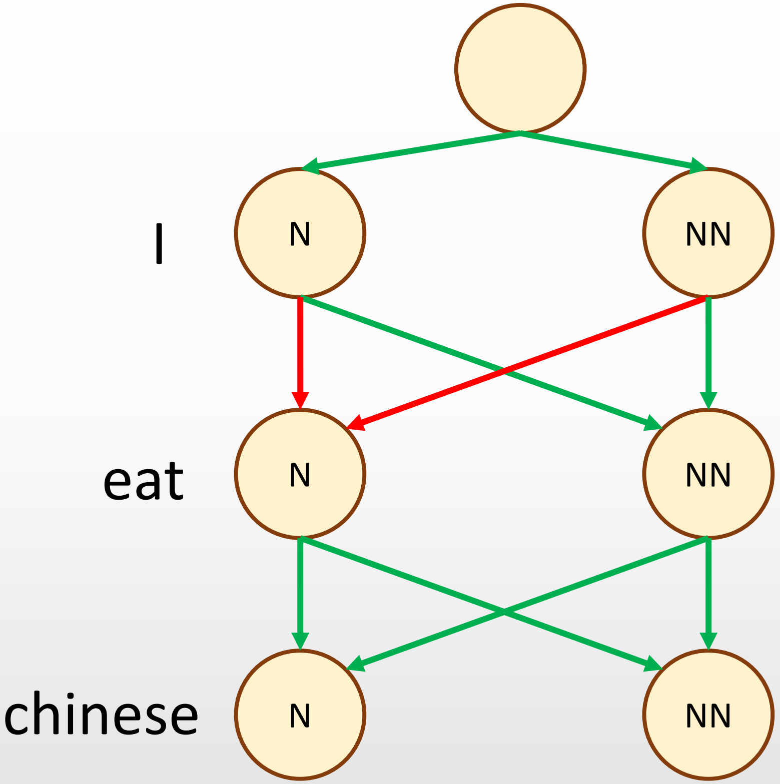


$P(t_1, w_1) \; = P(t_1)P(w_1|t_1)$

- $P(t_1 = \mathrm{N})P(w_1 = \mathrm{I}|t_1 = \mathrm{N}) = 0.7 * 0.8 = 0.56$
- $P(t_1 = NN)P(w_1 = \mathrm{I}|t_1 = \mathrm{NN}) = 0.3 * 0.1 = 0.03$

The Viterbi Algorithm (cont.)

p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

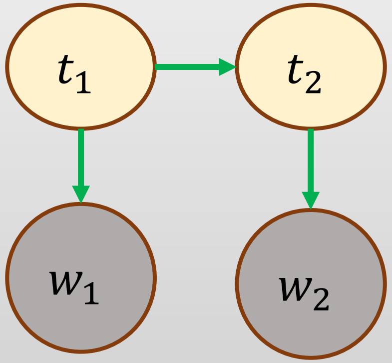
A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

$\pi[i,t]$	I	eat	chinese
State N	0.56		
State NN	0.03		

$B[i,t]$	I	eat	chinese
State N	-		
State NN	-		



$$P(t_1, t_2, w_1, w_2) = P(t_1)P(t_2|t_1)P(w_1|t_1)P(w_2|t_2)$$

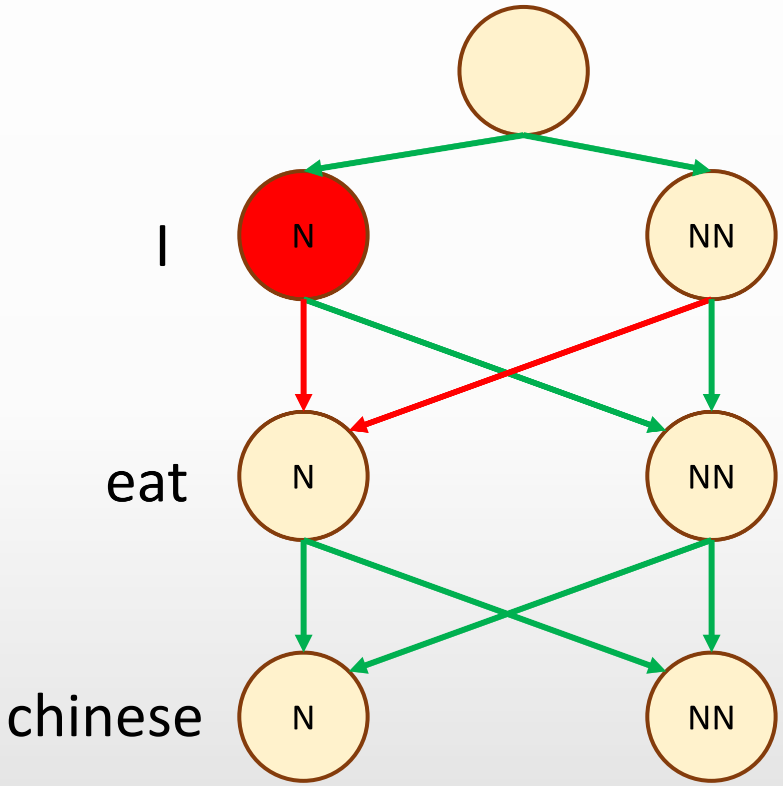
- $P(t_1)P(w_1|t_1)P(t_2 = \mathrm{N}|t_1 = \mathrm{N})(w_2 = eat|t_2 = \mathrm{N})$
  $= 0.56 * 0.6 * 0.01 = 0.00336$
- $P(t_1)P(w_1|t_1)P(t_2 = \mathrm{N}|t_1 = \mathrm{NN})(w_2 = eat|t_2 = N)$
  $= 0.03 * 0.5 * 0.01 = 0.00015$

Recursive Step:

$$\pi[i,t] = \max_{t'}\{\pi[i-1,t'] + logP(t|t') + logP(w_i|t)\}$$

The Viterbi Algorithm (cont.)

p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

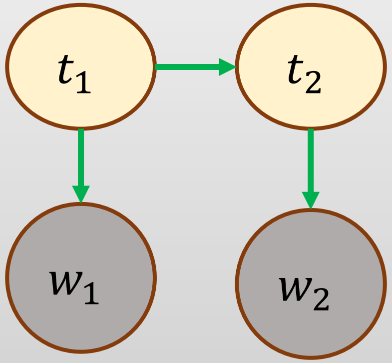
B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

$\pi[i,t]$	I	eat	chinese
State N	0.56	0.00336	
State NN	0.03		

B$[i,t]$	I	eat	chinese
State N	-	N	
State NN	-		

Recursive Step:

$$\pi[i,t] = \max_{t'}\{\pi[i-1,t'] + logP(t|t') + logP(w_i|t)\}$$

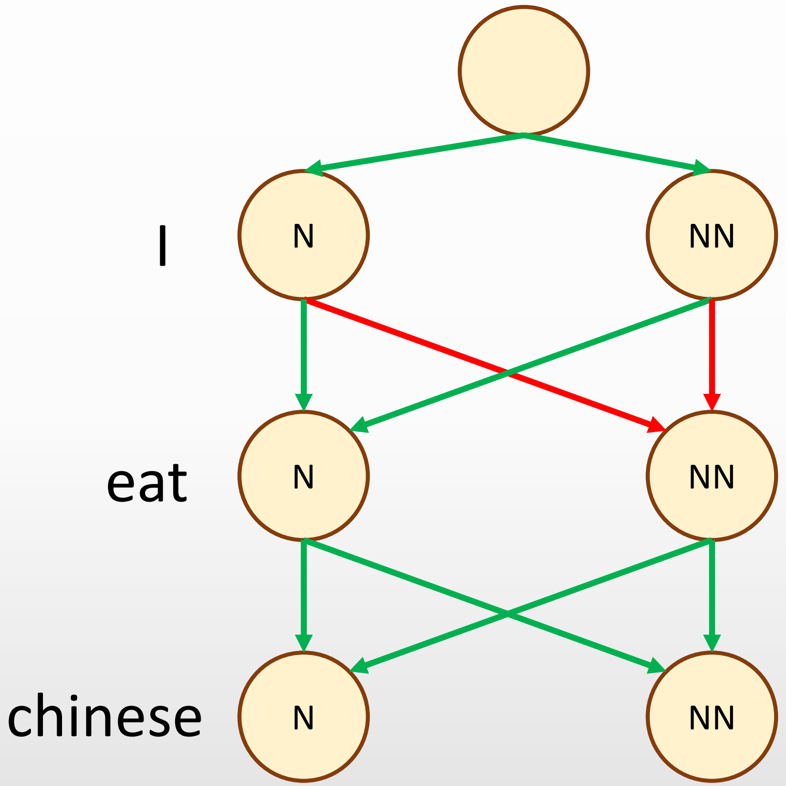


$$P(t_1,t_2,w_1,w_2) = P(t_1)P(t_2|t_1)P(w_1|t_1)P(w_2|t_2)$$

- $P(t_1)P(w_1|t_1)P(t_2 = \mathrm{N}|t_1 = \mathrm{N})(w_2 = eat|t_2 = \mathrm{N})$
  $= 0.56 * 0.6 * 0.01 = 0.00336$
- $P(t_1)P(w_1|t_1)P(t_2 = \mathrm{N}|t_1 = \mathrm{NN})(w_2 = eat|t_2 = N)$
  $= 0.03 * 0.5 * 0.01 = 0.00015$

The Viterbi Algorithm (cont.)

p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

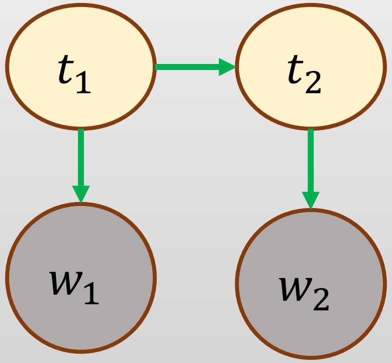
B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

$\pi[i,t]$	I	eat	chinese
State N	0.56	0.00336	
State NN	0.03		

$B[i,t]$	I	eat	chinese
State N	-	N	
State NN	-	N	

Recursive Step:

$$\pi[i,t] = \max_{t'}\{\pi[i-1,t'] + logP(t|t') + logP(w_i|t)\}$$



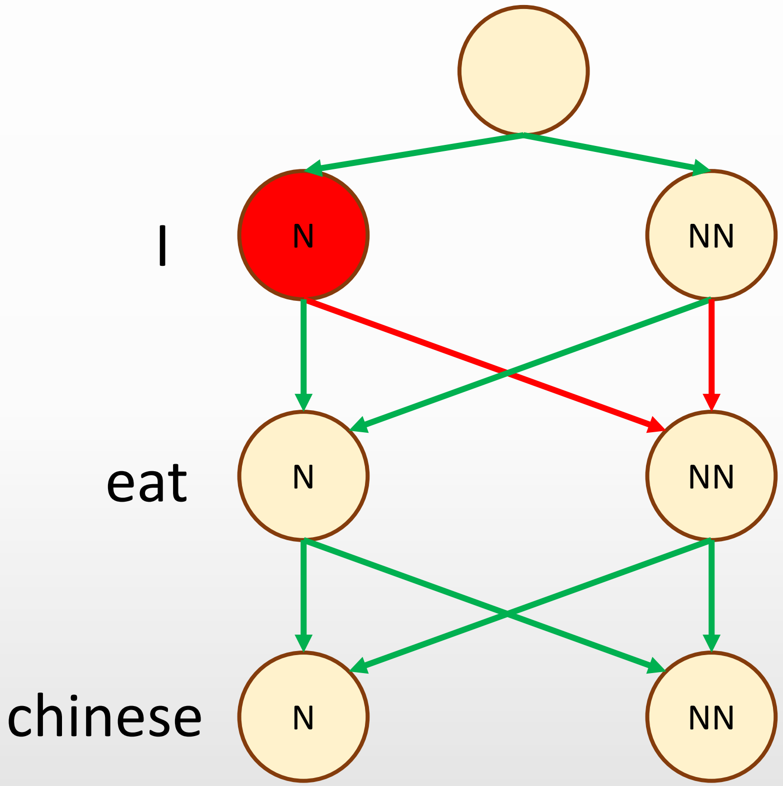
$P(t_1, t_2, w_1, w_2) = P(t_1)P(t_2|t_1)P(w_1|t_1)P(w_2|t_2)$

- $P(t_1)P(w_1|t_1)P(t_2 = \mathrm{NN}|t_1 = \mathrm{N})(w_2 = eat|t_2 = \mathrm{NN})$
  $=$

$P(t_1)P(w_1|t_1)P(t_2 = \mathrm{NN}|t_1 = \mathrm{NN})(w_2 = eat|t_2 = \mathrm{NN})$
$=$

The Viterbi Algorithm (cont.)

p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

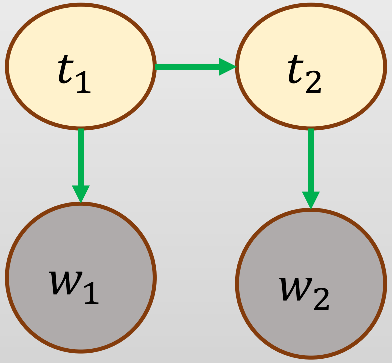
B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

$\pi[i,t]$	I	eat	chinese
State N	0.56	0.00336	
State NN	0.03	0.1008	

$B[i,t]$	I	eat	chinese
State N	-	N	
State NN	-	N	

Recursive Step:

$$\pi[i,t] = \max_{t'}\{\pi[i-1,t'] + logP(t|t') + logP(w_i|t)\}$$



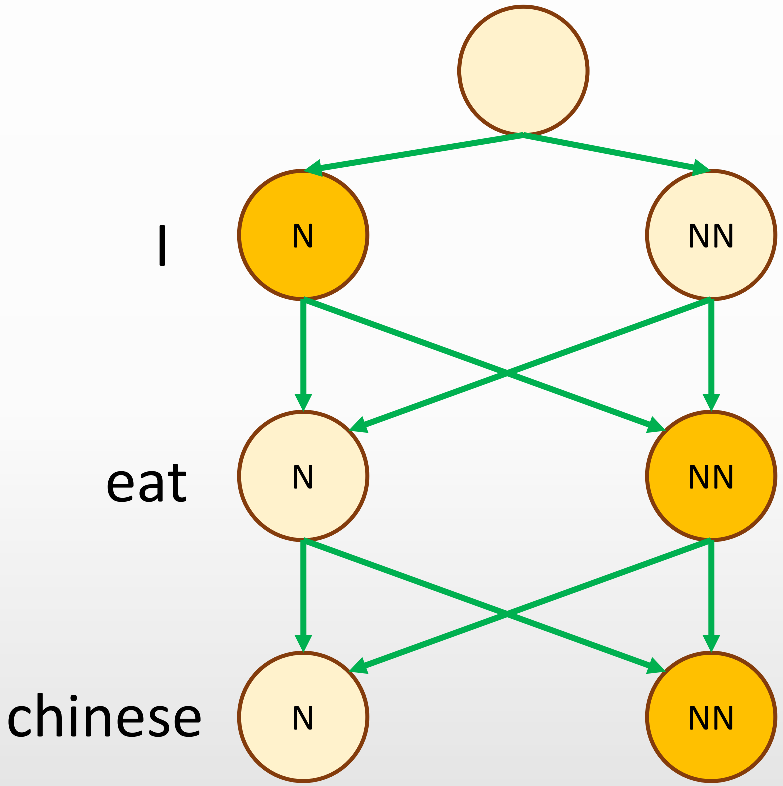
$P(t_1, t_2, w_1, w_2) = P(t_1)P(t_2|t_1)P(w_1|t_1)P(w_2|t_2)$

- $P(t_1)P(w_1|t_1)P(t_2 = \text{NN}|t_1 = \text{N})(w_2 = eat|t_2 = \text{NN})$
  $= 0.56 * 0.4 * 0.45 = 0.1008$

$P(t_1)P(w_1|t_1)P(t_2 = \text{NN}|t_1 = \text{NN})(w_2 = eat|t_2 = \text{NN})$
$= 0.03 * 0.5 * 0.45 = 0.00675$

The Viterbi Algorithm (cont.)

p_0 = [N,NN] = [0.7,0.3]



Transition probabilities

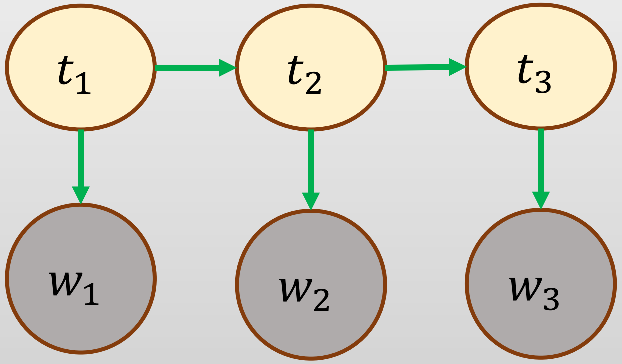
A_{ij}	to N	to NN
From N	0.6	0.4
From NN	0.5	0.5

Emission probabilities

B_{ik}	I	eat	chinese
State N	0.8	0.01	0.19
State NN	0.1	0.45	0.45

$\pi[i,t]$	I	eat	chinese
State N	0.56	0.00336	0.009576
State NN	0.03	0.1008	0.02268

B$[i,t]$	I	eat	chinese
State N	-	N	NN
State NN	-	N	NN



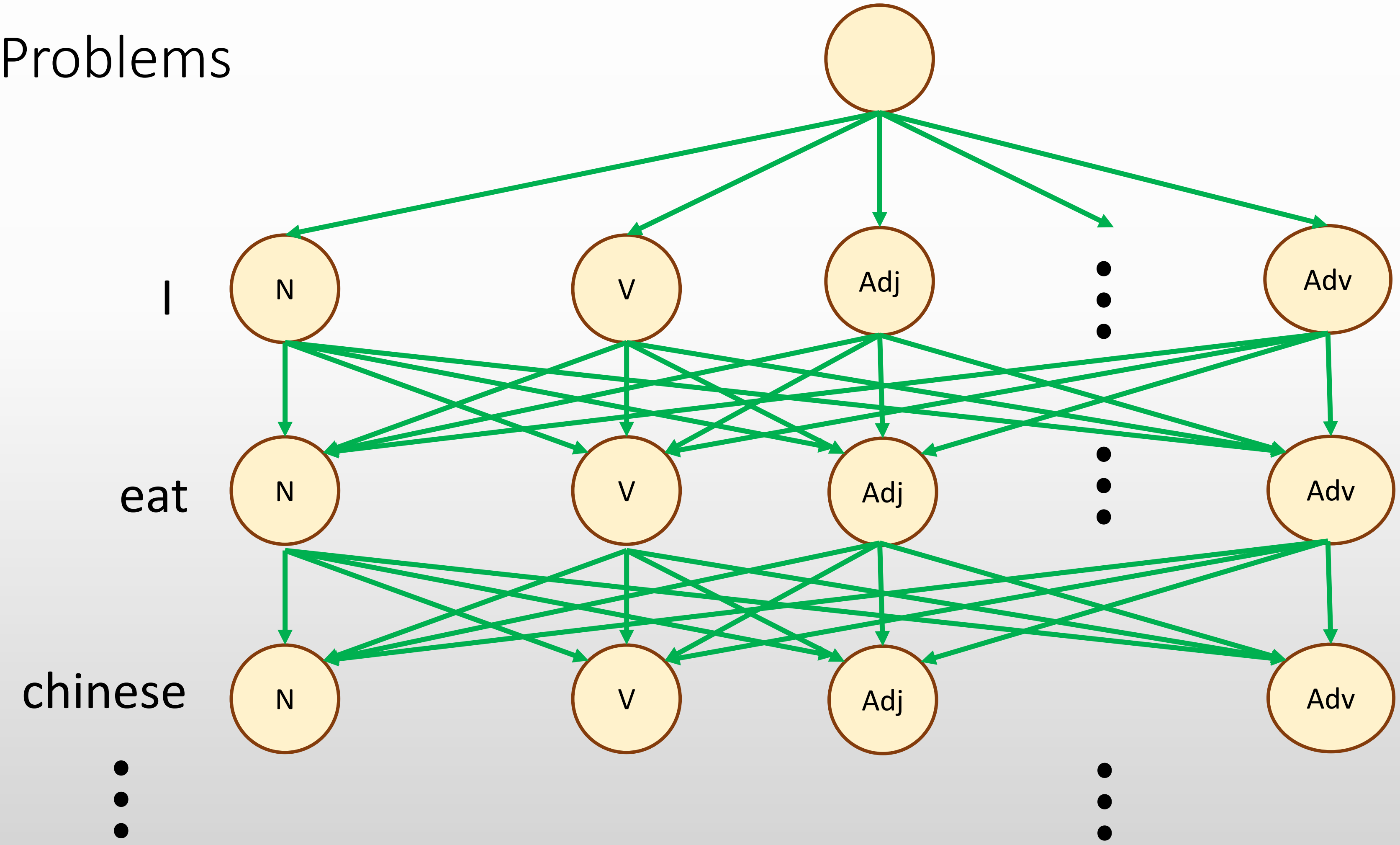
Backtrack
N, NN, NN

Recursive Step:

$$\pi[i,t] = \max_{t'}\{\pi[i-1,t'] + logP(t|t') + logP(w_i|t)\}$$

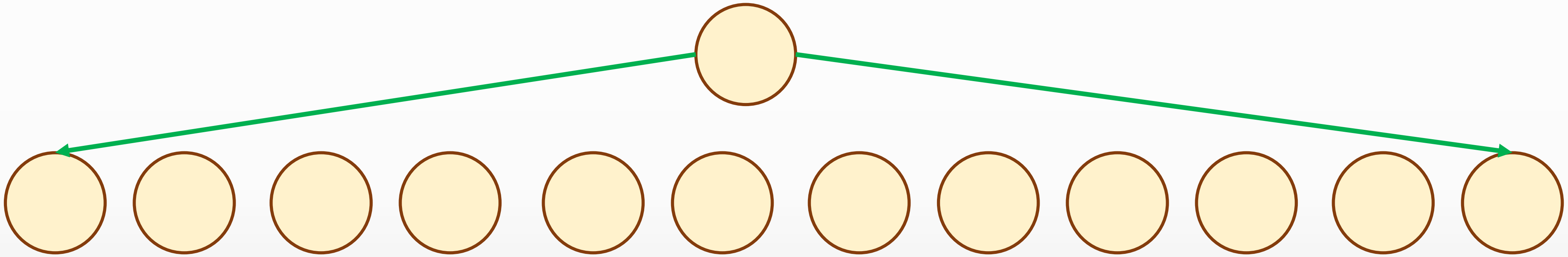
The Viterbi Algorithm (cont.)

- Problems



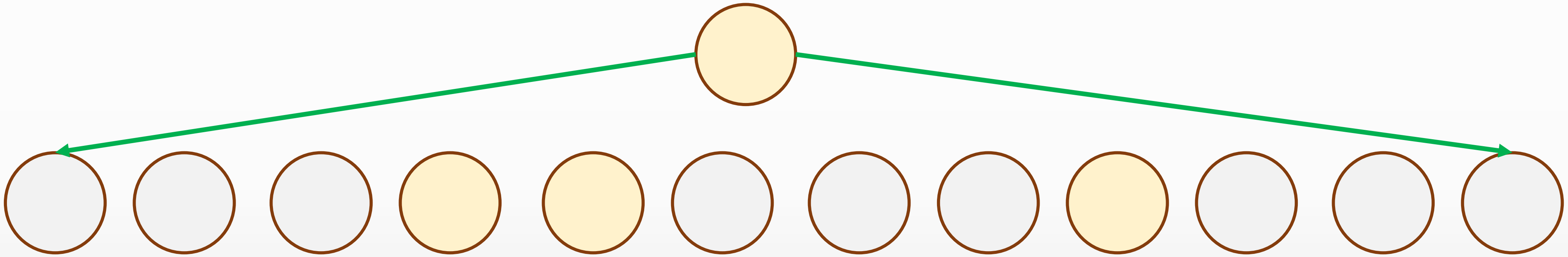
Pruning with Beamsearch

- Keep K active states at each step ; K = 3



Pruning with Beamsearch (cont.)

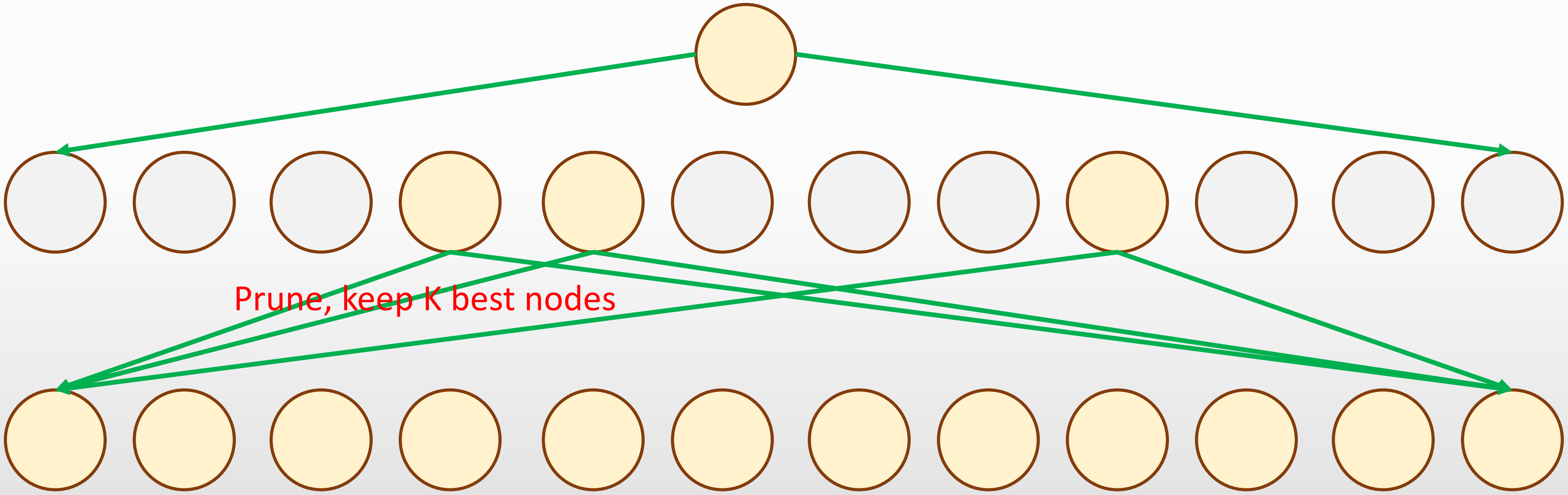
- Keep K active states at each step ; K = 3



Prune, keep K best nodes

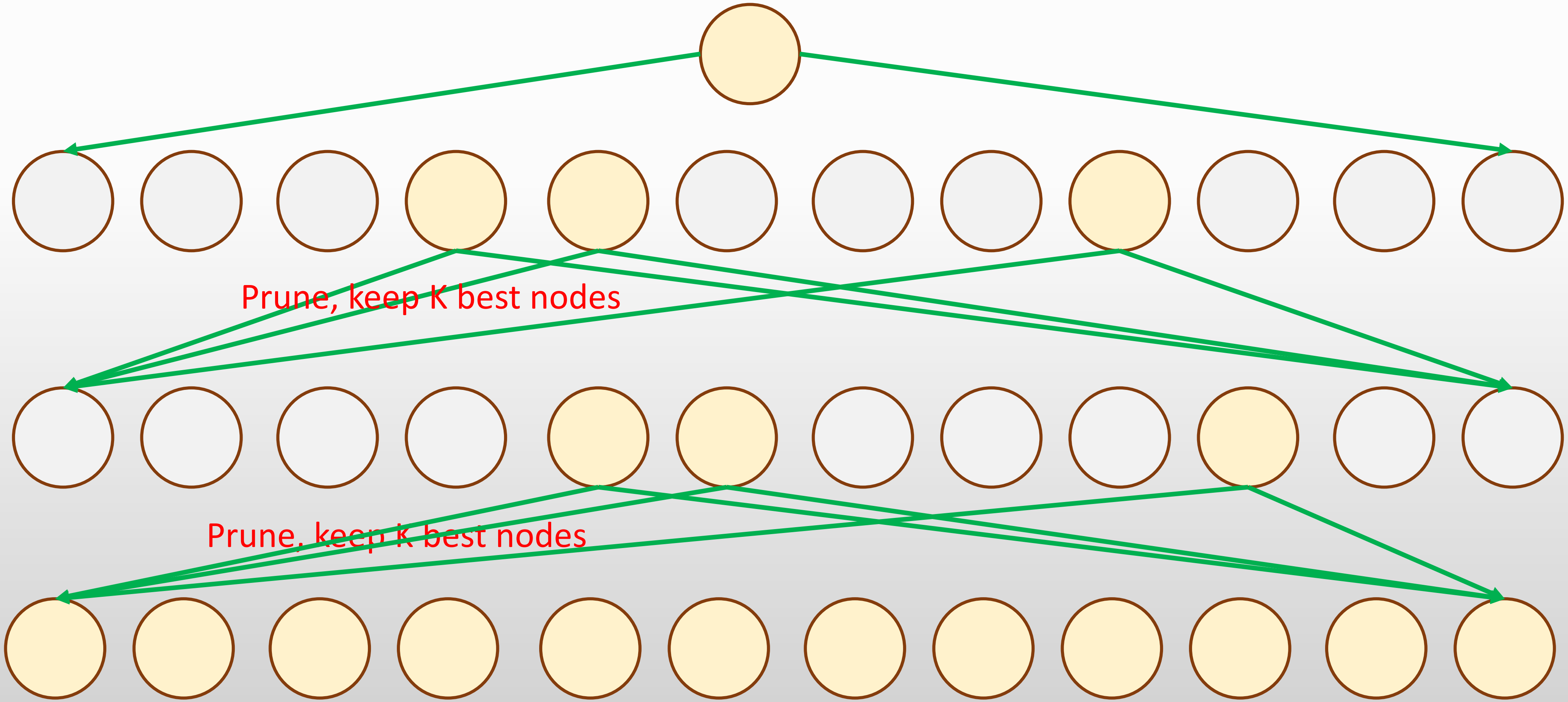
Pruning with Beamsearch (cont.)

- Keep K active states at each step ; K = 3



Pruning with Beamsearch (cont.)

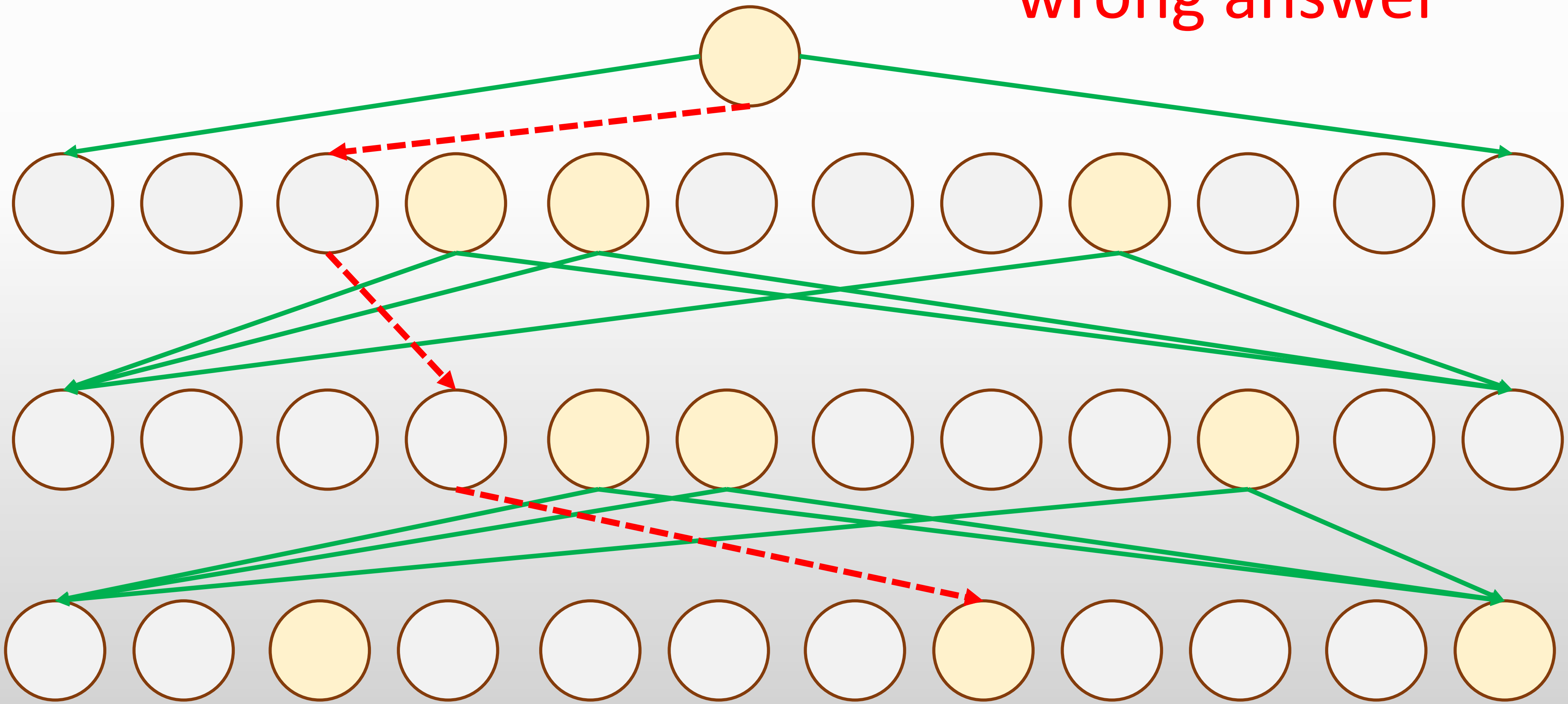
- Keep K active states at each step ; K = 3



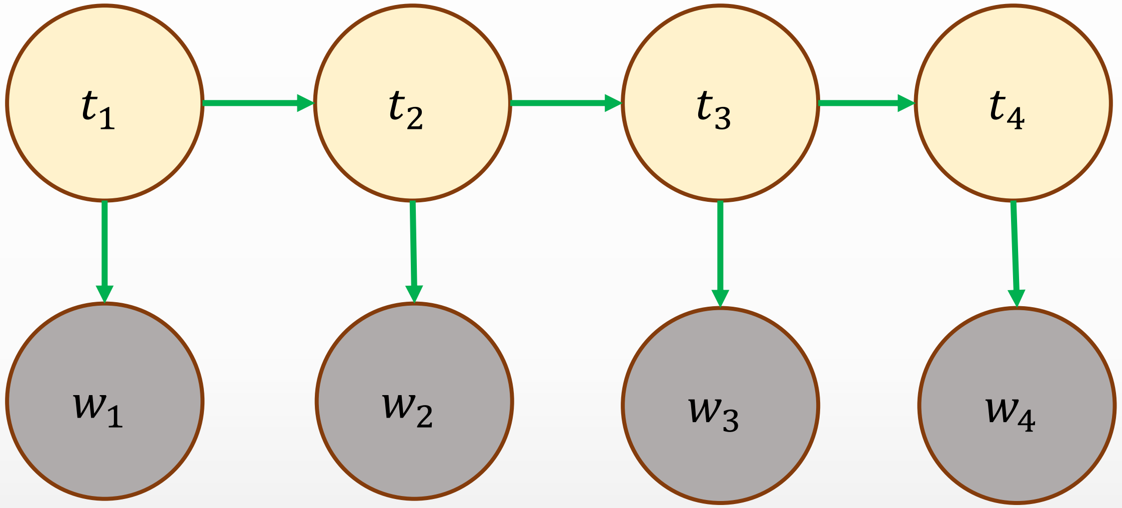
Beamsearch and Inadmissibility

- Keep K active states at each step ; K = 3

Pruning can give the wrong answer

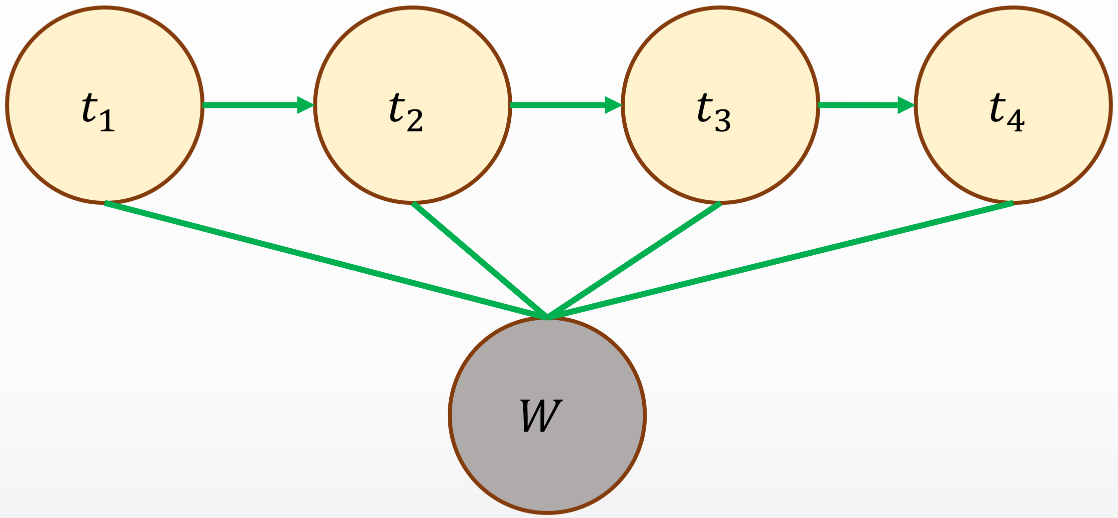


HMM and Disadvantages



- No dependency across words.
- HMM is a generative model $P(T, W)$, But we care about $P(T|W)$.

Conditional Random Fields (CRF)

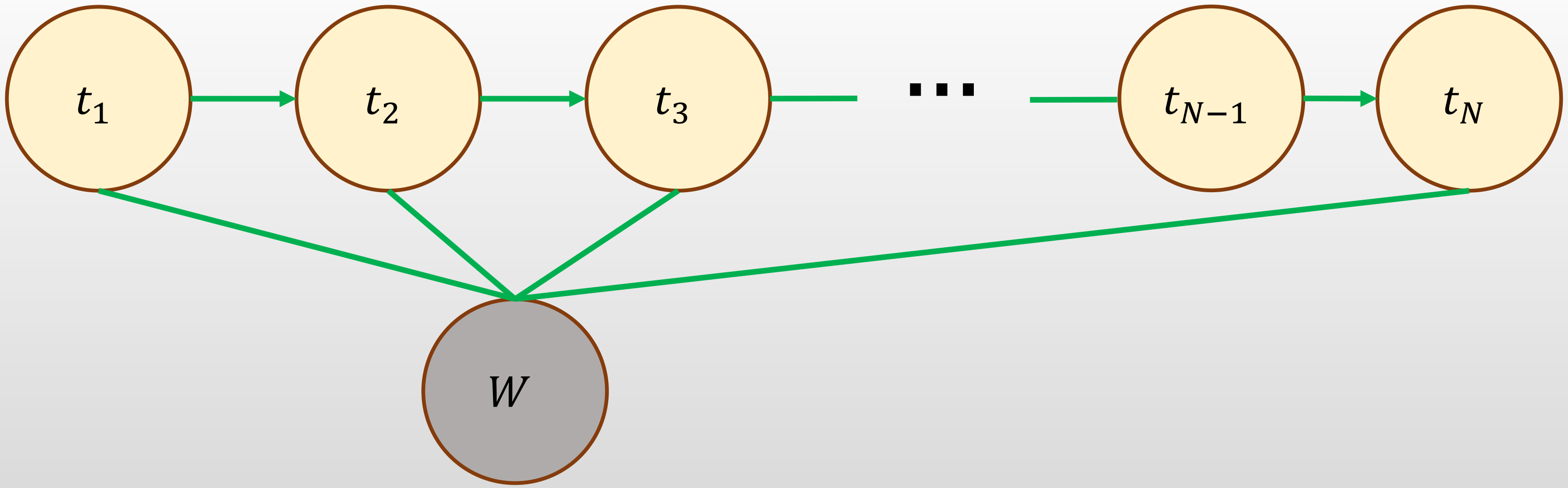


$$P(T|W) = P(t_n|t_{n-1}, W) = P(t_1|W)P(t_2|t_1, W)P(t_3|t_2, W)P(t_4|t_3, W)$$

- Every point in the chain now depends on the entire sentence
- CRF is a discriminative model

Linear chain CRF

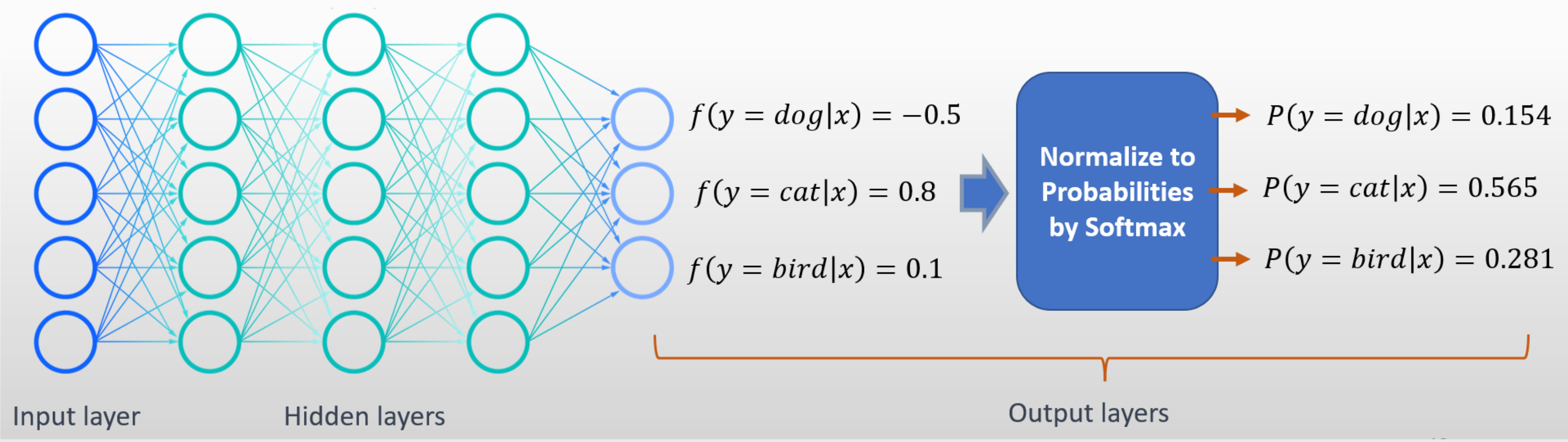
- Assumptions:
  - Each label t_n only depends on previous label t_{n-1}
  - Each label t_n globally depends on W



$P(T|W) = P(t_n|t_{n-1}, W)$ is hard to find.

Recap Deep learning models and CRF ideas

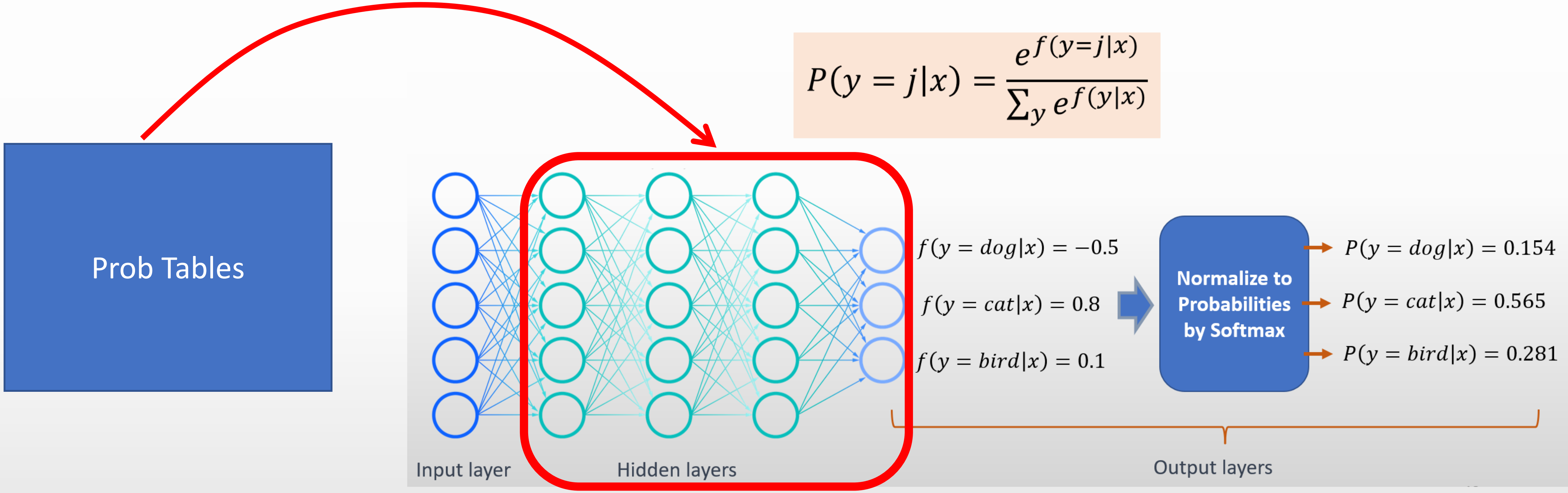
$$P(y = j|x) = \frac{e^{f(y=j|x)}}{\sum_y e^{f(y|x)}}$$



- Probability distribution is a function (with special constraints).
- Find a function that will represent "probabilities".
- We can turn functions into probabilities easily.

Recap Deep learning models and CRF ideas

$$P(y=j|x)=\frac{e^{f(y=j|x)}}{\sum_y e^{f(y|x)}}$$



- Probability distribution is a function (with special constraints).
- Find a function that will represent "probabilities".
- We can turn functions into probabilities easily.

Goals of Linear-CRF

- Building functions that represent the whole sequence is hard
  - We'll build by combining pieces
  - But each piece should have the form $f(t_n, t_{n-1}, W, n) \rightarrow \mathbb{R}$
  - This is from our independence assumption
  - We call these functions, feature functions

Goals of Linear-CRF (cont.)

- Building functions that represent the whole sequence is hard
  - We'll build by combining pieces
  - But each piece should have the form $f(t_n, t_{n-1}, W, n) \rightarrow \mathbb{R}$
  - This is from our independence assumption
  - We call these functions, feature functions

- Example features
  - Transition function

$$f_1(t_n, t_{n-1}, W, n) = \begin{cases} 1, & \text{if } t_n = \text{NOUN and } t_{n-1} = \text{ADJ} \\ 0, & \text{otherwise} \end{cases}$$

Goals of Linear-CRF (cont.)

- Building functions that represent the whole sequence is hard
  - We'll build by combining pieces
  - But each piece should have the form $f(t_n, t_{n-1}, W, n) \rightarrow \mathbb{R}$
  - This is from our independence assumption
  - We call these functions, feature functions

- Example features

- state function

$$f_2(t_n, t_{n-1}, W, n) = \begin{cases} 1, \text{if } t_n = \text{NOUN and } w_n = \text{fox} \\ 0, \quad \text{otherwise} \end{cases}$$

$$f_3(t_n, t_{n-1}, W, n) = \begin{cases} 1, \text{if } t_n = \text{NOUN and } w_{n-1} = \text{an} \\ 0, \quad \text{otherwise} \end{cases}$$

Goals of Linear-CRF (cont.)

- Example features (more)

- state function

$$f_4(t_n, t_{n-1}, W, n) = \begin{cases} 1, \text{if } t_n = \text{PROPER NOUN and } w_n \text{is capitalized} \\ 0, \qquad \text{otherwise} \end{cases}$$

$$f_5(t_n, t_{n-1}, W, n) = \begin{cases} 1, \text{if } t_n = \text{NOUN and } w_{n-1} \text{ ends with "est"} \\ 0, \qquad \text{otherwise} \end{cases}$$

Potential

$$\Psi_n(t_n, t_{n-1}, W) = \exp\left[\sum_k^K \theta_k f_k(t_n, t_{n-1}, W)\right]$$

- At each time step, a potential is a function that takes all feature functions into account, by summing their products with the associated weight.
- K: number of all feature function
- θ_k : weight parameters

Potential: Examples

- W = The fastest fox jumps
- $\theta_1 = 2.54, \theta_2 = 0.13, \theta_3 = 1.12, \theta_4 = 2.01, \theta_5 = 0.97$
- n = 3

$$f_1(t_n, t_{n-1}, W, n) = \begin{cases} 1, & \text{if } t_n = \text{NOUN and } t_{n-1} = \text{ADJ} \\ 0, & \text{otherwise} \end{cases}$$

$$f_2(t_n, t_{n-1}, W, n) = \begin{cases} 1, \text{if } t_n = \text{NOUN and } w_n = \text{fox} \\ 0, \quad \text{otherwise} \end{cases}$$

$$f_3(t_n, t_{n-1}, W, n) = \begin{cases} 1, \text{if } t_n = \text{NOUN and } w_{n-1} = \text{an} \\ 0, \quad \text{otherwise} \end{cases}$$

$$f_4(t_n, t_{n-1}, W, n) = \begin{cases} 1, \text{if } t_n = \text{PROPER NOUN and } w_n \text{is capitalized} \\ 0, \quad \text{otherwise} \end{cases}$$

$$f_5(t_n, t_{n-1}, W, n) = \begin{cases} 1, \text{if } t_n = \text{NOUN and } w_{n-1} \text{ ends with "est"} \\ 0, \quad \text{otherwise} \end{cases}$$

$$\Psi_3(t_3, t_{3-1}, W) = \exp\left[\sum_{k=1}^{5} \theta_k\, f_k(t_3, t_{3-1}, W)\right]$$

$$= \exp[(0 \times 2.54) + (1 \times 0.13) + (0 \times 1.12) + (0 \times 2.01) + (1 \times 0.97)] = 3$$

Probability of the whole sequence

- Joint probability distribution of input and output sequence can be represented as:

$$P(T|W) = \frac{P(T,W)}{P(W)} = \frac{P(T,W)}{\sum_T P(T,W)} = \frac{1}{Z(W)} \prod_{n=1}^{N} \Psi_n(t_n, t_{n-1}, W)$$

where $Z(W) = \sum_T \prod_{n=1}^{N} \Psi_n(t_n, t_{n-1}, W)$

Probability of the whole sequence (cont.)

- Joint probability distribution of input and output sequence can be represented as:

$$P(T|W) = \frac{P(T,W)}{P(W)} = \frac{P(T,W)}{\sum_T P(T,W)} = \frac{1}{Z(W)} \prod_{n=1}^{N} \Psi_n(t_n, t_{n-1}, W)$$

$$P(T|W) = \frac{P(T,W)}{P(W)} = \frac{P(T,W)}{\sum_T P(T,W)} = \frac{1}{Z(W)} \prod_{n=1}^{N} \exp\left[\sum_k^K \theta_k f_k(t_n, t_{n-1}, W)\right]$$

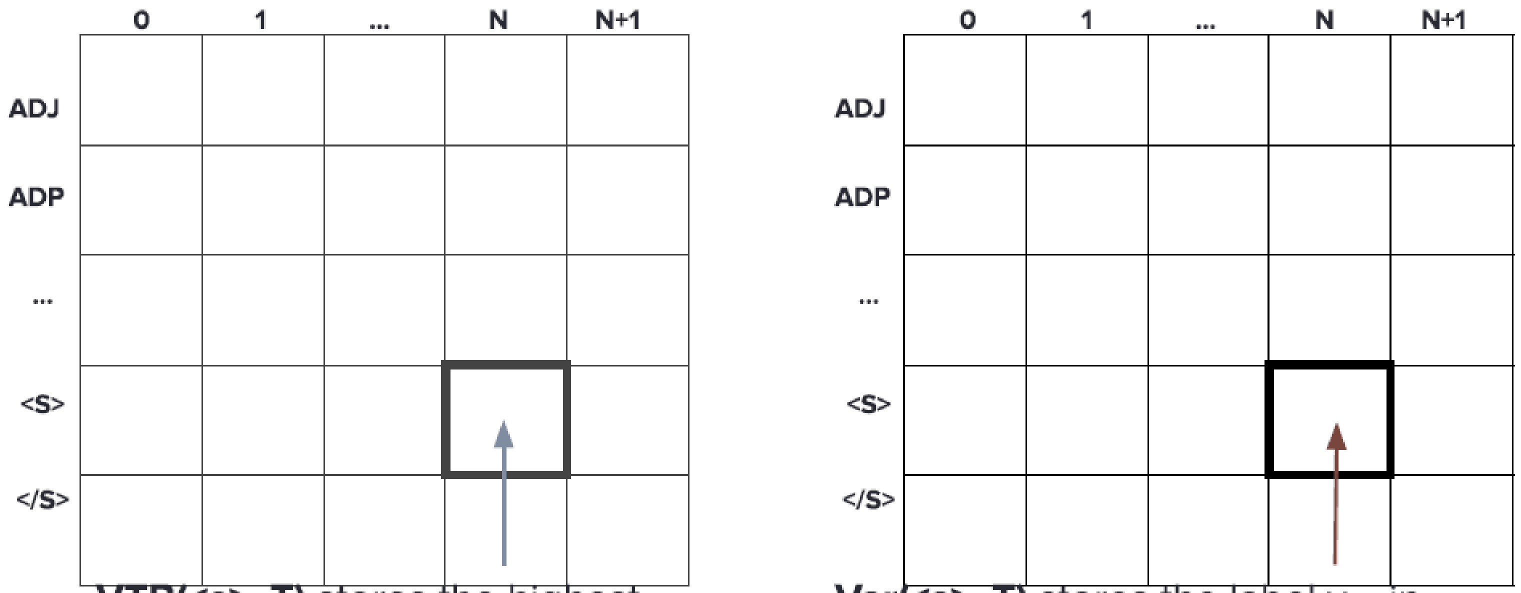
where $Z(W) = \sum_T \prod_{n=1}^{N} \Psi_n(t_n, t_{n-1}, W)$

How to use Linear-CRF

$$P(T|W) = \frac{1}{Z(W)} \boxed{\prod_{n=1}^{N} \Psi_n(t_n, t_{n-1}, W)}$$

<s> indicates the beginning of the sequence
</s> indicates the end of the sequence

- Viterbi algorithm
  - Create two 2D arrays: VTB and Var
  - Backtrack



VTB(<s>, T) stores the highest value a label sequence that y_T is <s> can take

Var(<s>, T) stores the label y_{T-1} in the sequence that yields the value in VTB(<s>, T)

How to use Linear-CRF (cont.)

<s> indicates the beginning of the sequence
</s> indicates the end of the sequence

$$P(T|W) = \frac{1}{Z(W)} \boxed{\prod_{n=1}^{N} \Psi_n(t_n, t_{n-1}, W)} = \frac{1}{Z(W)} \boxed{\prod_{n=1}^{N} \exp\left[\sum_{k}^{K} \theta_k f_k(t_n, t_{n-1}, W)\right]}$$

- Viterbi algorithm
  - Create two 2D arrays: VTB and Var
  - Backtrack

VTB

	0	1	...	N	N+1
ADJ	0	0.12			
ADP	0	0.03			
...	...	...			
</S>	0	10e-9			
<S>	1	10e-3			

For all other labels, apply the same way as ADJ

$$\mathrm{VTB}(\mathrm{ADJ},1) = \underbrace{\Psi_1(\mathrm{ADJ}, < s >, \mathbf{x})}_{\text{Factors at time t=1, for current label=ADJ, previous label=< s >}} \underbrace{\mathrm{VTB}(< s >,0)}_{1}$$

Var

	0	1	...	N	N+1
ADJ	-	<s>			
ADP	-	<s>			
...	...	...			
</S>	-	<s>			
<S>	-	<s>			

The first label of the output sequence must be <s>

How to use Linear-CRF (cont.)

<s> indicates the beginning of the sequence
</s> indicates the end of the sequence

$$P(T|W) = \frac{1}{Z(W)} \boxed{\prod_{n=1}^{N} \Psi_n(t_n, t_{n-1}, W)} = \frac{1}{Z(W)} \boxed{\prod_{n=1}^{N} \exp\left[\sum_{k}^{K} \theta_k f_k(t_n, t_{n-1}, W)\right]}$$

- Viterbi algorithm
  - Create two 2D arrays: VTB and Var
  - Backtrack

Iterate from n=2 to n=N+1

VTB

	0	1	...	N	N+1
ADJ	0	0.12	...	2.32	0.22
ADP	0	0.03	...	0.02	0.10
...	...	...	...	...	...
</S>	0	10e-9	...	0.03	3.09
<S>	1	10e-3	...	1.12	0.02

Var

	0	1	...	N	N+1
ADJ	-	<s>	...	NOUN	PREPO
ADP	-	<s>	...	NOUN	ADJ
...	...	...	...	...	...
</S>	-	<s>	...	VERB	ADJ
<S>	-	<s>	...	X	X

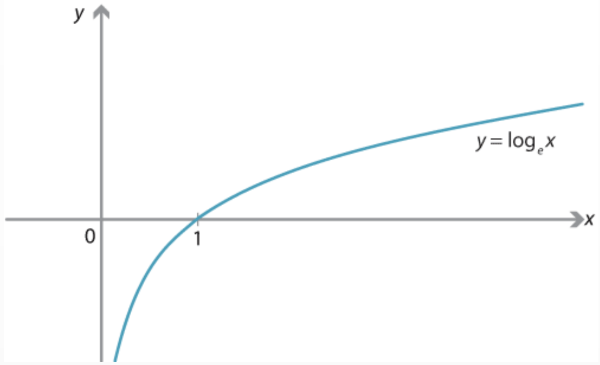
$$VTB(ADJ,n) = \max_{i \in Tags} \underbrace{\Psi_n(ADJ,i,W)}_{\text{Factors at time n}} \underbrace{VTB(i,n-1)}_{\text{Maximum value that label i can take at time n-1}}$$

Find the max among values from all previous label i

Var(t, n) Stores the value i that maximize the value of VTB(t,n)

Linear-CRF training

$$l(\theta) = \sum_{i=1}^{N} \log P(T^{(i)}|W^{(i)})$$

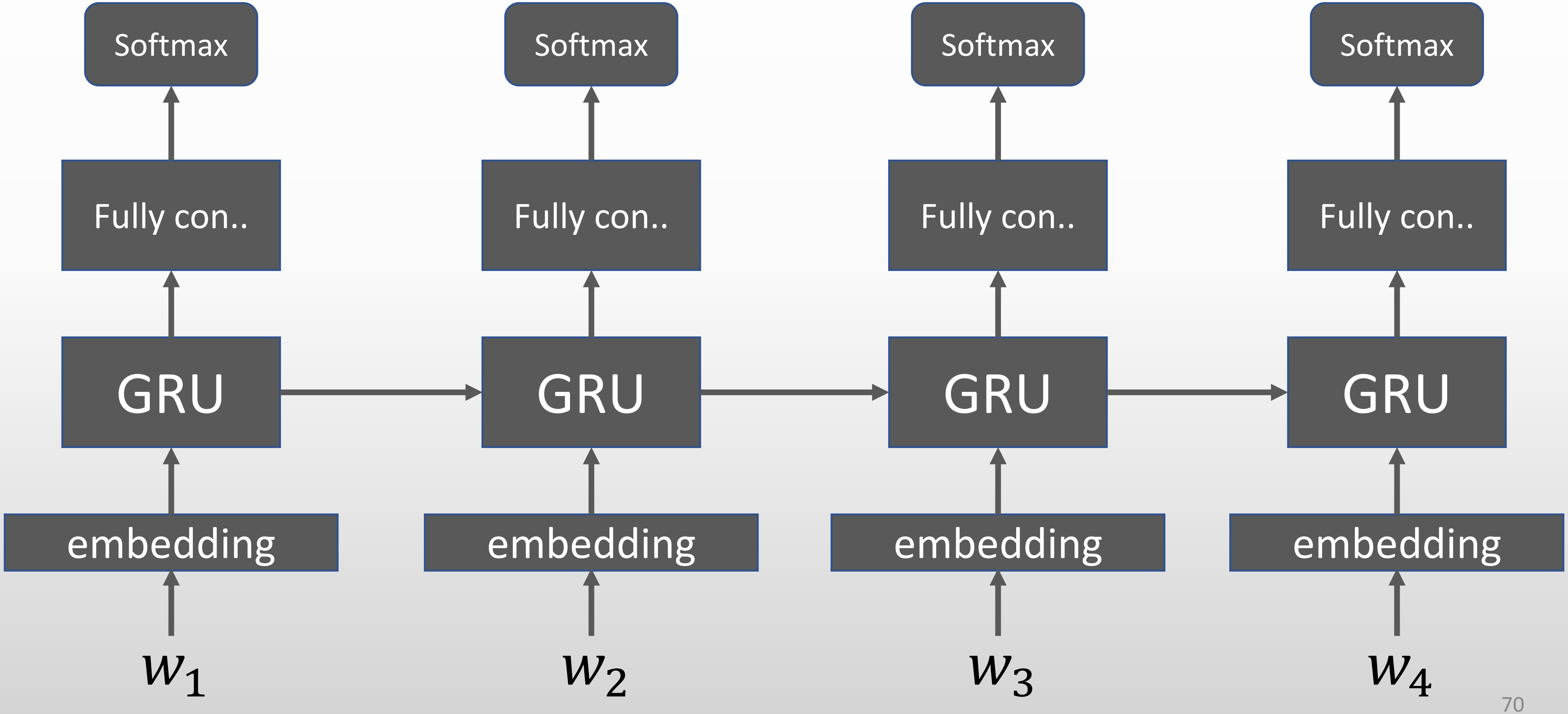


- For linear-chain CRF, parameters are trained by maximum likelihood.
- To clarify, parameters θ are trained to maximize the log probability of all pairs of label $T^{(i)}$ and input $W^{(i)}$ in the training set. (i) represents the i^{th} training sentence.
- Learning algorithm: Limited-memory BFGS, Stochastic Gradient Descent, etc.

CRF with neural networks

- Caveats: most CRF takes discrete features
- Neural networks features (embeddings) are continuous.
- Solution1: discretize the embeddings
  - Example: Cat = [1.1,2.3] --> [1,1]
- Solution2: use continuous version of CRF
- Solution3: change the softmax layer and loss function

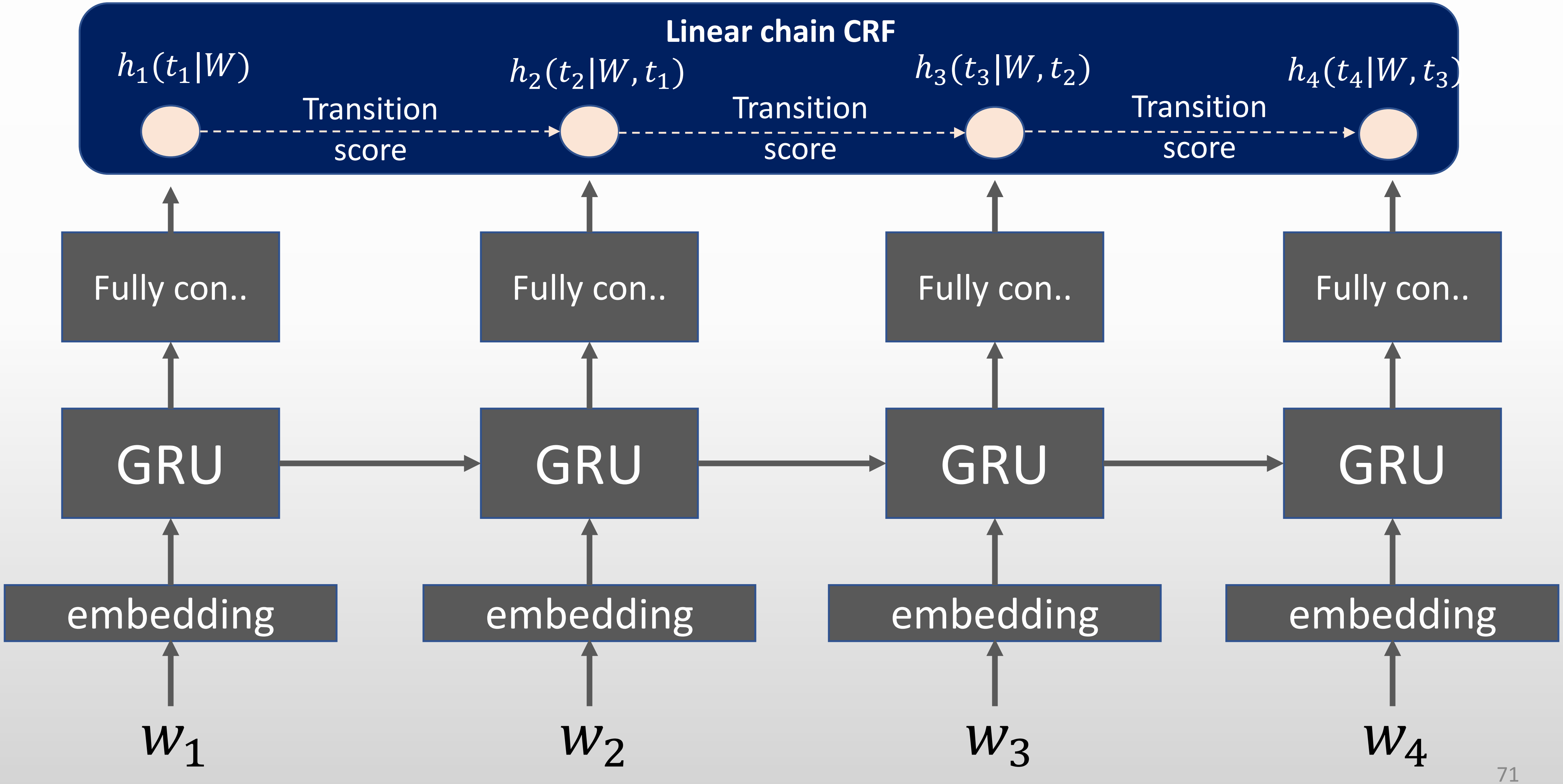
Neural networks for PoS



$$P(T|W) = \frac{1}{Z(W)} \prod_{n=1}^{N} \exp\left[\sum_{k}^{K} \theta_k f_k(t_n, t_{n-1}, W)\right]$$

Neural networks with CRF for PoS

$$= \frac{1}{Z(W)} \prod_{n=1}^{N} \exp[h_n(t_n|W)]$$



Performance

Neural networks

Model	Acc.
Giménez and Màrquez (2004)	97.16
Toutanova et al. (2003)	97.27
Manning (2011)	97.28
Collobert et al. (2011)‡	97.29
Santos and Zadrozny (2014)‡	97.32
Shen et al. (2007)	97.33
Sun (2014)	97.36
Søgaard (2011)	97.50
This paper	**97.55**

Table 4: POS tagging accuracy of our model on test data from WSJ proportion of PTB, together with top-performance systems. The neural network based models are marked with ‡.

Model	F1
Chieu and Ng (2002)	88.31
Florian et al. (2003)	88.76
Ando and Zhang (2005)	89.31
Collobert et al. (2011)‡	89.59
Huang et al. (2015)‡	90.10
Chiu and Nichols (2015)‡	90.77
Ratinov and Roth (2009)	90.80
Lin and Wu (2009)	90.90
Passos et al. (2014)	90.90
Lample et al. (2016)‡	90.94
Luo et al. (2015)	91.20
This paper	**91.21**

Neural networks

Table 5: NER F1 score of our model on test data set from CoNLL-2003. For the purpose of comparison, we also list F1 scores of previous top-performance systems. ‡ marks the neural models.

Ma, Xuezhe, and Eduard Hovy. "End-to-end sequence labeling via bi-directional lstm-cnns-crf." *arXiv preprint arXiv:1603.01354* (2016).

https://arxiv.org/pdf/1603.01354.pdf

Demo: PoS

https://drive.google.com/file/d/1b6PvGunI3ijT5EgTB7JniTiJr4OmWsQq/view?usp=share_link

HW01

https://drive.google.com/drive/folders/1g3vrxK-wXm5g7DBvD0jylcy5874-uE3b?usp=share_link