COMP5046
Natural Language Processing

Lecture 6: Part-of-Speech (POS) Tagging

Semester 1, 2019

School of Computer Science
The University of Sydney, Australia

THE UNIVERSITY OF SYDNEY

Caren Han
Caren.Han@sydney.edu.au

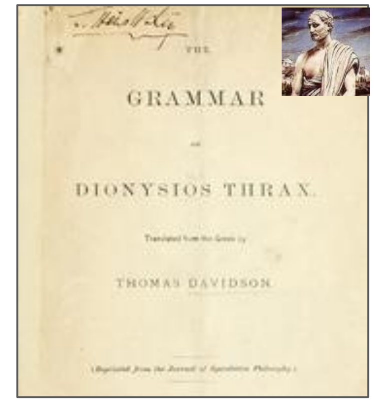
0 LECTURE PLAN

Lecture 6: Part of Speech Tagging

1. **Part-of-Speech Tagging**
2. Baseline Approaches
   1. Lexicon-based Methods
   2. Rule-based Methods
3. Probabilistic Approaches
   1. Hidden Markov Model
   2. Conditional Random Field
4. Deep Learning Approaches

Parts of Speech?

Linguists have been classifying words for a long time



Dionysius Thrax of Alexandria
wrote a grammatical sketch of Greek
involving 8 parts-of-speech:

Nouns	***Verbs***	***Pronouns***	***Prepotisions***
Adverbs	***Conjunctions***	***Participles***	***Articles***

Thrax's list and minor variations on it dominated European language grammars and dictionaries for 2000 years.

English Tagsets

In modern (English) NLP, larger (and more fine-grained) tagsets are preferred.

Example

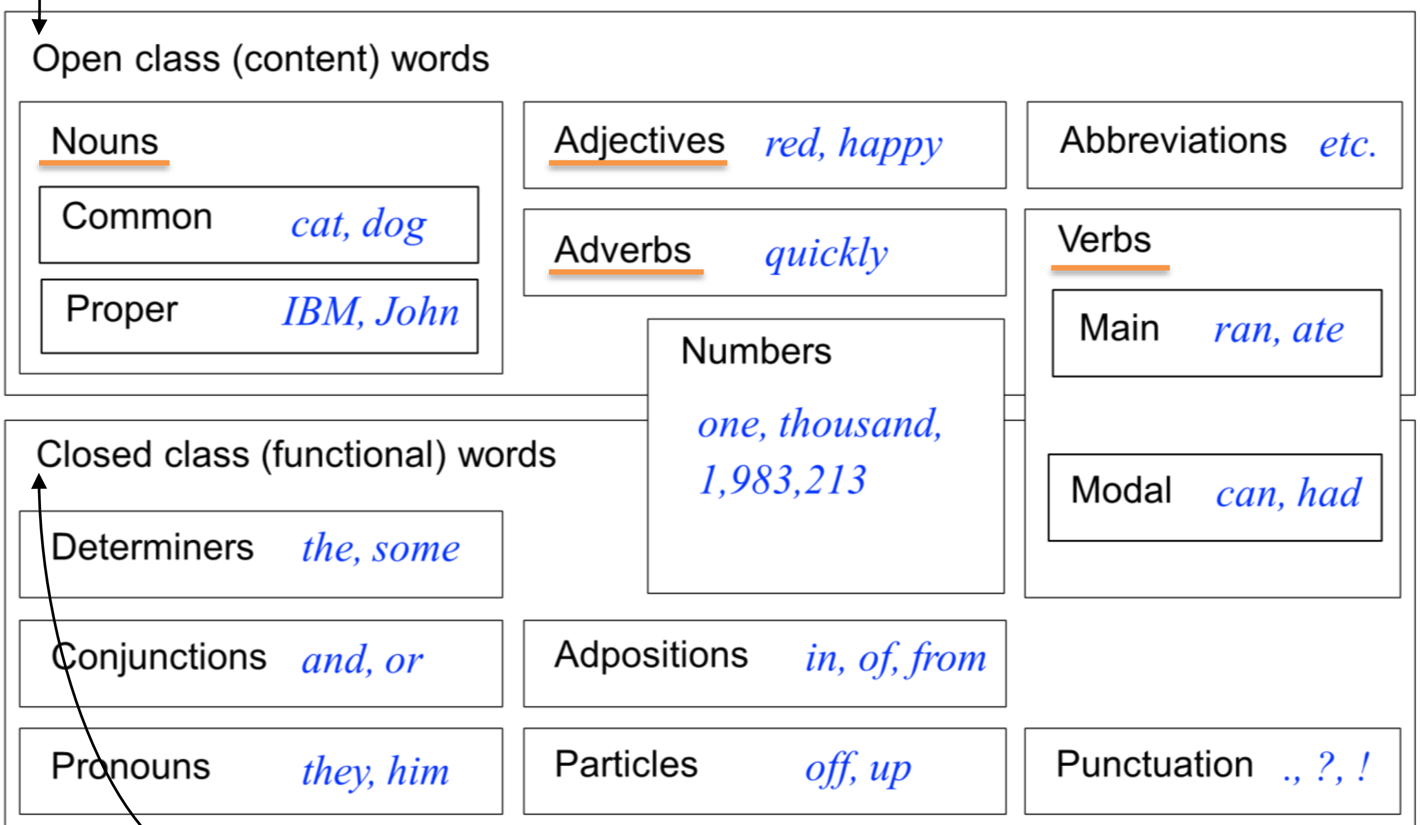
Penn Treebank	*45 tags*	http://bit.ly/1gwbird
Brown Corpus	*87 tags*	https://bit.ly/2FGtdLd
C7 Tagset	*146 tags*	http://bit.ly/1Mh36KX

Trade-off between complexity and precision and whatever tagset we use, there'll be some words that are hard to classify.

Parts-of-Speech (English)

One basic kind of linguistic structure: syntactic word classes

typically large, have fluid membership, and are often stable under translation



small, have fixed membership, and the repertoire of classes varies widely

Criteria for part-of-speech tagging

Three different criteria might be considered.

- ***Distributional*** criteria: Where can the words occur?

- ***Morphological*** criteria: What form does the word have? (E.g. -tion, -ize). What affixes can it take? (E.g. -s, -ing, -est).

- ***Notional(or semantic)*** criteria: What sort of concept does the word refer to? (E.g. nouns often refer to ‘people, places or things’). More problematic: less useful for us

Part-of-Speech Tagging

Criteria for part-of-speech tagging: Nouns

Three different criteria might be considered.

- ***Distributional*** criteria: Where can the nouns appear?

  For example, nouns can appear with possession: "his car", "her idea".

- ***Morphological*** criteria: What form does the word have? (E.g. -tion, -ize). What affixes can it take? (E.g. -s, -ing, -est).

  ness, -tion, -ity, and -ance tend to indicate nouns. (happiness, exertion, levity, significance).

- ***Notional(or semantic)*** criteria: What sort of concept does the word refer to?

  Nouns generally refer to living things (mouse), places (Sydney), non-living things (computer), or concepts (marriage).

Common POS Categories: Nouns

- **NN** - common noun, singular or mass
  - Examples: cabbage, thermostat, investment
- **NNS** - common noun, plural
  - Examples: undergraduates, thieves
- **NNP** - proper singular noun
  - Examples: Mary, Jasper
- **NNPS** - proper plural noun
  - Examples: Americans, Democrats

Criteria for part-of-speech tagging: Verbs

Three different criteria might be considered.

- ***Distributional*** criteria: Where can the verbs appear?

  Different types of verbs have different distributional properties. For example, base form verbs can appear as infinitives: “to jump”, ”to learn”.

- ***Morphological*** criteria: What form does the word have? (E.g. -tion, -ize). What affixes can it take? (E.g. -s, -ing, -est).

  words that end in -ate or -ize tend to be verbs, and ones that end in -ing are often the present participle of a verb (automate, equalize; rising, washing)

- ***Notional(or semantic)*** criteria: What sort of concept does the word refer to?

  Verbs refer to actions (observe, think, give).

Common POS Categories - Verbs

- **VB** - verb, base form
  - Examples: ask, bring, fire, see, take
- **VBD** - verb, past tense
  - Examples: pleaded, swiped, registered, saw
- **VBG** - verb, present participle or gerund
  - Examples: stirring, focusing, approaching, erasing

POS tags in Penn Treebank

Table 2
The Penn Treebank POS tagset.

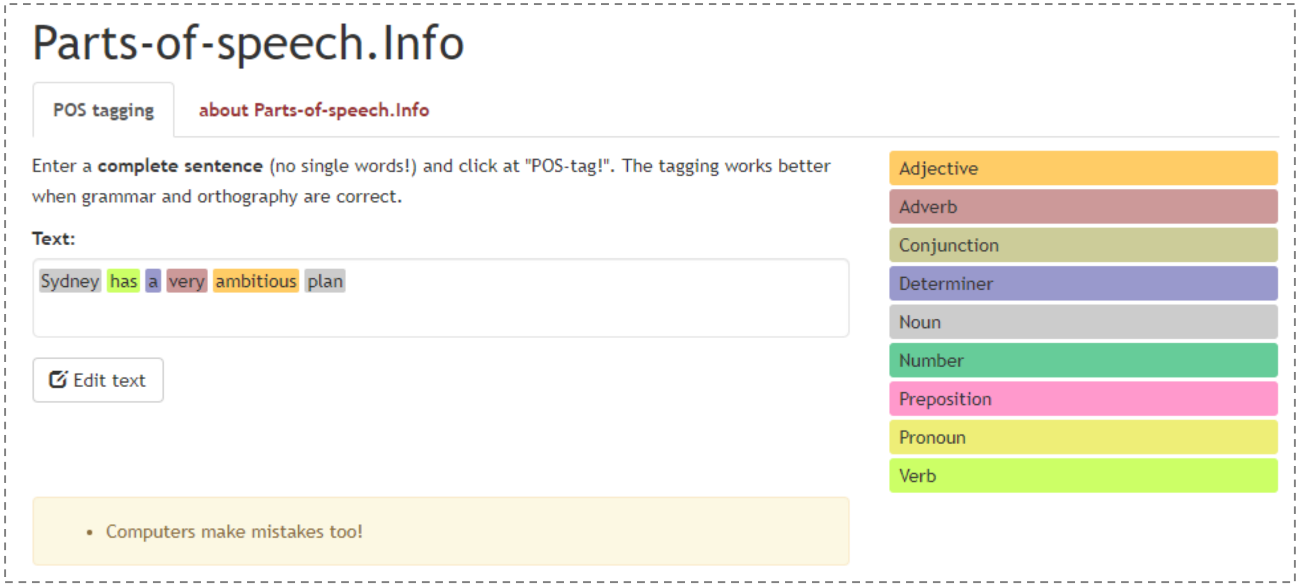
1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	‘	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	”	Right close double quote

Marcus, et al. (1993)

Part-of-Speech Tagging

Example of POS inference

Sydney	has	a	very	ambitious	plan
NNP	VBZ	DT	RB	JJ	NN



https://parts-of-speech.info/ , StandfordNLP(2018)

POS Tagging: Issue

Given an input text, tag each word correctly:

There/ was/ still/ lemonade/ in/ the/ bottle/

- (Tag sets are quite counterintuitive!)
  - In the above, the *bottle* is a noun not a verb
    - *but how does our tagger tell?*
  - The *still* could be an adjective or an adverb
    - *which seems more likely?*

The purpose of POS Tagging

Essential ingredient in natural language applications

- Useful in and of itself (more than you'd think)
  - Text-to-speech: record, lead
  - Lemmatization: saw[v] see, saw[n] saw
  - Linguistically motivated word clustering

- Useful as a pre-processing step for parsing

- Useful as features to downstream systems.

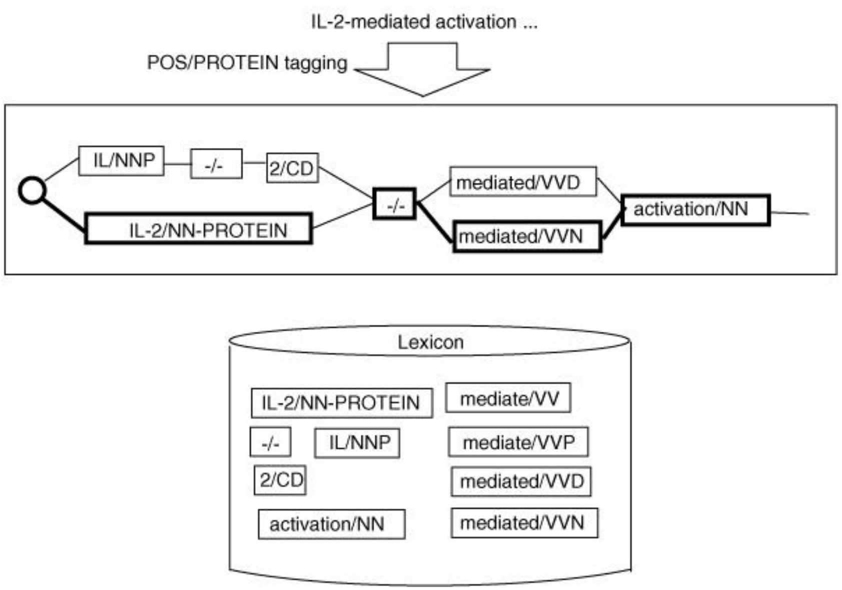
0 LECTURE PLAN

Lecture 6: Part of Speech Tagging

1. Part-of-Speech Tagging
2. **Baseline Approaches**
    1. Lexicon-based Methods
    2. Rule-based Methods
3. Probabilistic Approaches
    1. Hidden Markov Model
    2. Conditional Random Field
4. Deep Learning Approaches

Lexicon-based POS Tagging

- Just look up the word in a dictionary and look up the part of speech tag.
- Annotate text with POS and negation information.
- Identify words present on lexicon
- Aggregate results



"Cannot handle unknown words."

Rule-based POS Tagging

Basic idea:

- Old POS taggers used to work in two stages, based on hand-written rules:
  1. Identifies a set of possible POS for each word in the sentence (based on a lexicon),
  2. Uses a set of hand-crafted rules in order to select a POS from each of the lists for each word

Rule-based POS Tagging

Approach:

- Assign each token all its possible tags.
- Apply rules that eliminate all tags for a token that are inconsistent with its context.

Example

the	DT (determiner)		the	DT (determiner)	
can	MD (modal)	$\Rightarrow$	can	MD (modal)	X
	NN (sg noun)			NN (sg noun)	√
	VB (base verb)			VB (base verb)	X

- Assign any unknown word tokens a tag that is consistent with its context (eg, the most frequent tag).

THE UNIVERSITY OF SYDNEY

Rule-based POS Tagging

- Rule-based tagging often used a large set of hand-crafted context-sensitive rules.

Example (schematic):

Example

the	DT (determiner)		the	DT (determiner)	
can	MD (modal)	$\Rightarrow$	can	MD (modal)	X
	NN (sg noun)			NN (sg noun)	√
	VB (base verb)			VB (base verb)	X

"Cannot eliminate all POS ambiguity."

2 Baseline Approaches

N-gram with POS Tagging

One simple strategy (a.k.a. unigram tagging) *:* just assign to each word its most common tag. (So still will always get tagged as an adverb — never as a noun, verb or adjective.) only consider one token at a time.

Surprisingly, even this crude approach typically gives around 90% accuracy. (State-of-the-art is 96–98%).

Can we do better? We'll look briefly at bigram, trigram tagging, then at Hidden Markov Model tagging.

N-gram with POS Tagging

Bigram* *Tagging* :** We can do much better by looking at pairs of adjacent tokens. For each word (e.g. ***still), tabulate the frequencies of each possible POS given the POS of the **preceding word**.

still	DT	MD	JJ	. . .
NN	8	0	6	
JJ	23	0	14	
VB	1	12	2	
RB	6	45	3	

Given a new text, tag the words from left to right, assigning each word the most likely tag given the preceding one.

Could also consider ***trigram*** (or more generally n-gram) tagging, etc. But the frequency matrices would quickly get very large, and also (for realistic corpora) too ‘sparse’ to be really useful.

THE UNIVERSITY OF SYDNEY

Problems with N-gram Tagging

***Bigram* Tagging issue**

One incorrect tagging choice might have unintended effects:

	The	still	smoking	remains	of	the	campfire
Intended:	DT	RB	VBG	NNS	IN	DT	NN
Bigram:	DT	JJ	NN	VBZ	. . .		

No lookahead: choosing the 'most probable' tag at one stage might lead to highly improbable choice later.

	The	still	was	smashed
Intended:	DT	NN	VBD	VBN
Bigram:	DT	JJ	VBD?	

We'd prefer to find the overall most likely tagging sequence given the bigram frequencies. This is what the ***Hidden Markov Model (HMM)approach achieves***.

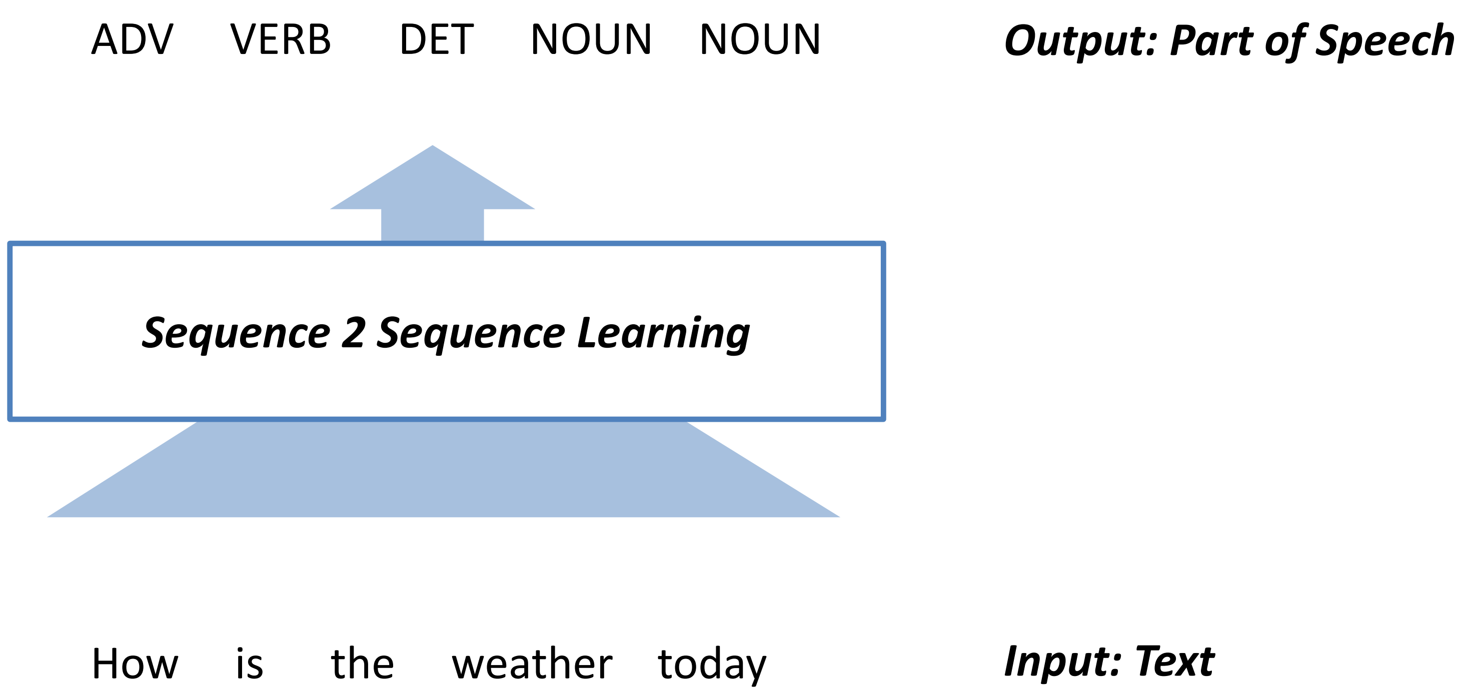
0 LECTURE PLAN

Lecture 6: Part of Speech Tagging

1. Part-of-Speech Tagging
2. Baseline Approaches
   1. Lexicon-based Methods
   2. Rule-based Methods
3. **Probabilistic Approaches**
   1. Hidden Markov Model
   2. Conditional Random Field
4. Deep Learning Approaches

3 Probabilistic Approaches

Sequence Tagging



Hidden Markov Model (HMM)

Hidden Markov Model

Hidden

What is ‘hidden’?

Markov Model

What is ‘Markov Model’?

Markov Model

Andrei Andreyevich Markov

The purpose of introducing Markov Chain
An example of statistical investigation in the text of `Eugene Onyegin' illustrating coupling of `tests' in chains.

- A stochastic model used to model randomly changing system
  - Assumes the Markov property
- A stochastic process has the Markov property if the ***conditional probability distribution of future states*** of the process ***depends only upon the present state***, not on the events that occurred before it.

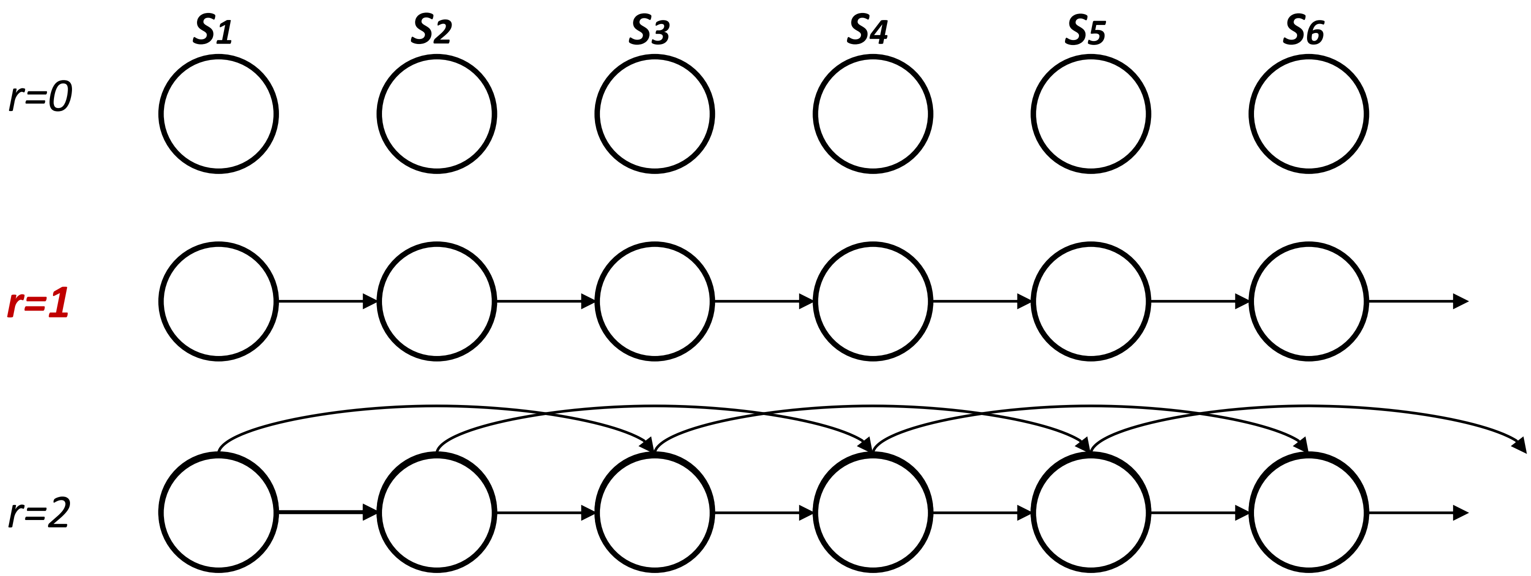
Markov Model (MM): Markov Property

- Assumption: last ***k*** states are sufficient
  - ***(r=1)*** First-order Markov Process – ***Most Commonly used***

    $$P(S_t \mid S_{t-1}, \ldots, S_0) = P(S_t \mid S_{t-1})$$

  - *(r=2)* Second-order Markov Process

    $$P(S_t \mid S_{t-1}, \ldots, S_0) = P(S_t \mid S_{t-1}, S_{t-2})$$



3 Probabilistic Approaches

Markov Model (MM): Example

- Classify (Sydney) weather into three states
  - State 1: Rainy
  - State 2: Cloudy
  - State 3: Sunny

State1

State2

State3

- Assume that we examined the weather of Sydney for a year, and found following weather change pattern.

		Tomorrow		
		Rainy	Cloudy	$\boxed{\text{Sunny}}$
Today	$\boxed{\text{Rainy}}$	0.4	0.3	0.3
	Cloudy	0.2	0.6	0.2
	Sunny	0.1	0.1	0.8

$$S_{ij} = P(S_t = j \mid S_{t-1} = i)$$

$$S_{rainysunny} = 0.3$$

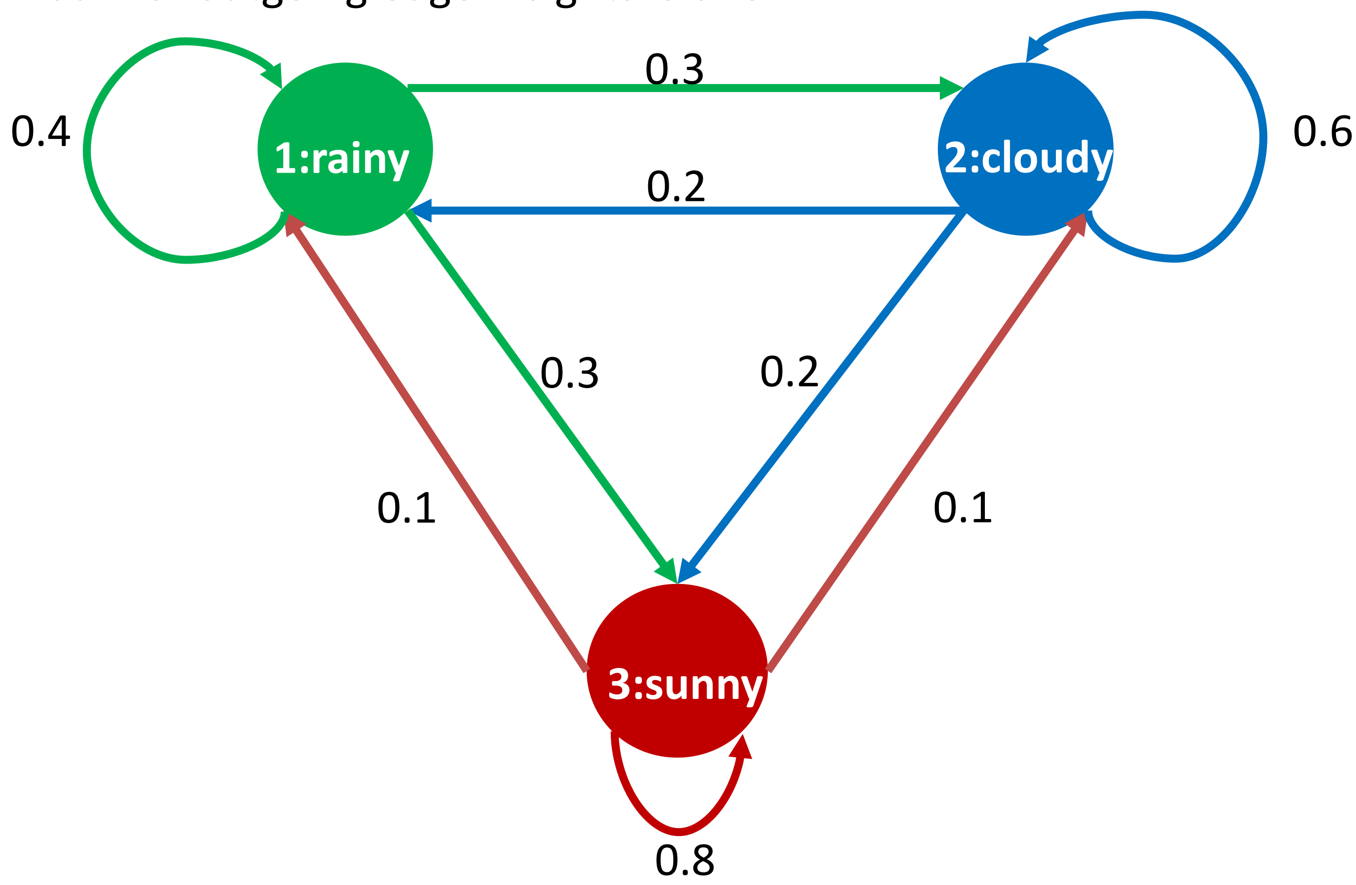
Assumption: Tomorrow weather depends only on today's!

Markov Model (MM): Example

Visual illustration with diagram

- Each state corresponds to one observation
- Sum of outgoing edge weights is one

		Tomorrow		
		Rainy	Cloudy	Sunny
Today	Rainy	0.4	0.3	0.3
	Cloudy	0.2	0.6	0.2
	Sunny	0.1	0.1	0.8



Markov Model (MM): Example

State Transition Matrix

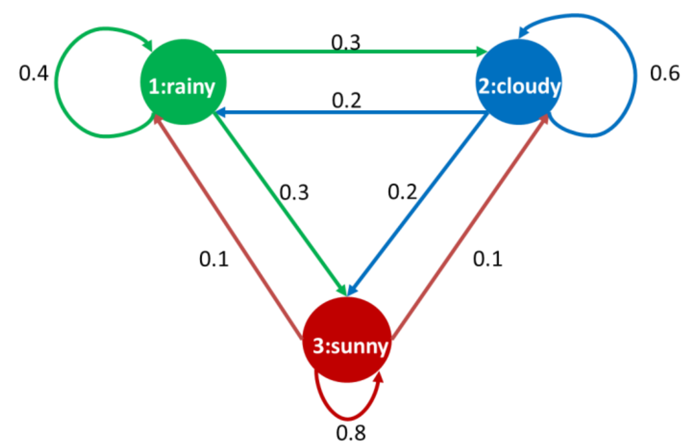
$$S_{ij} = P(S_t = j \mid S_{t-1} = i) \qquad 1 \leq i,\ \ j \geq N$$

$$S_{ij} \geq 0$$

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1N} \\ S_{21} & S_{22} & \cdots & S_{2N} \\ S_{31} & S_{32} & \cdots & S_{3N} \\ \vdots & \vdots & \vdots & \vdots \\ S_{N1} & S_{N2} & \cdots & S_{NN} \end{bmatrix}$$

		Tomorrow		
		Rainy	Cloudy	Sunny
Today	Rainy	0.4	0.3	0.3
	Cloudy	0.2	0.6	0.2
	Sunny	0.1	0.1	0.8

		Time ***t+1***		
		S_1	S_2	S_3
Time ***t***	S_1	0.4	0.3	0.3
	S_2	0.2	0.6	0.2
	S_3	0.1	0.1	0.8

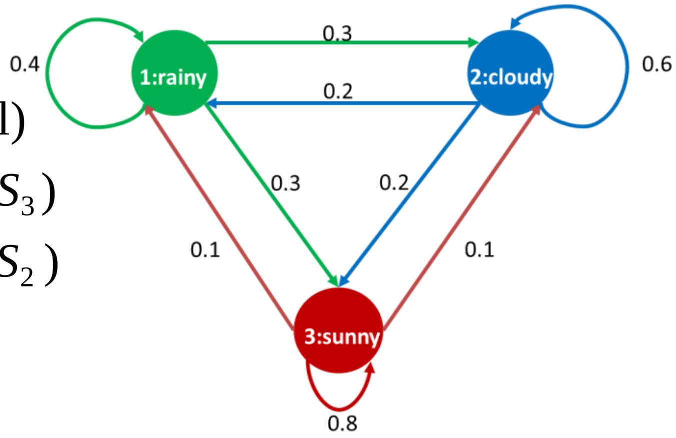


Markov Model (MM): Example

Sequence Probability

- Question: What is the probability that the weather for the next 7 days will be "sun-sun-rain-rain-sun-cloudy-sun" when today is sunny?
  - S1: Rainy
  - S2: Cloudy
  - S3: Sunny

$$\begin{aligned}
P(O \mid \text{model}) &= P(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \mid \text{model}) \\
&= P(S_3) \cdot P(S_3 \mid S_3) \cdot P(S_3 \mid S_3) \cdot P(S_1 \mid S_3) \\
&\quad \cdot P(S_1 \mid S_1) P(S_3 \mid S_1) P(S_2 \mid S_3) P(S_3 \mid S_2) \\
&= \pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \\
&= 1 \cdot (0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\
&= 1.536 \times 10^{-4}
\end{aligned}$$



3 Probabilistic Approaches

Hidden Markov Model (HMM)

Hidden Markov Model

Hidden

What is 'hidden'?

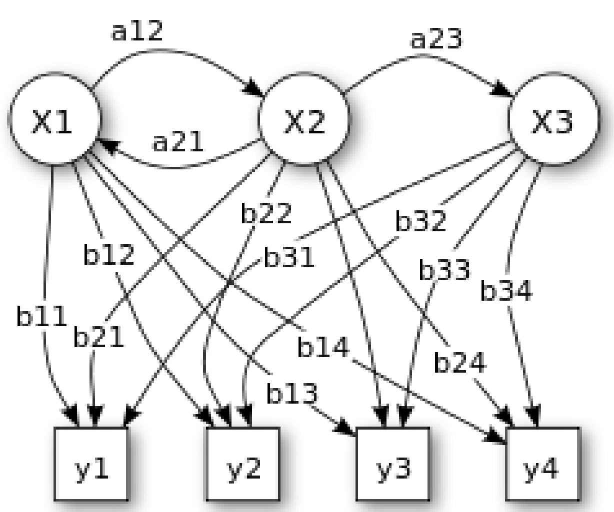
Markov Model

What is 'Markov Model'?

Future is independent of the past given the present

Hidden Markov Model (HMM)

Hidden Markov Models (HMMs) are a class of probabilistic graphical model that allow us to ***predict a sequence of unknown (hidden) variables*** from a set of observed variables.



hidden

x states

y possible observations

a state transition probabilities

b output probabilities

- States are ***hidden***
- ***Observable events*** linked to states
- Each state has ***observation probabilities*** to determine the observable event

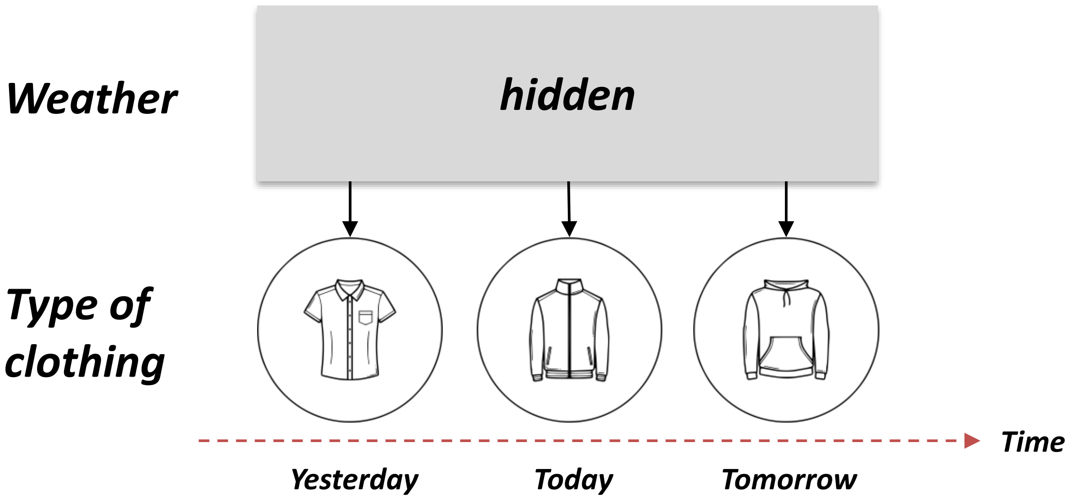
Hidden Markov Model (HMM)

Predicting the **weather** ***(hidden variable)*** based on the type of ***clothes that someone wears (observed)***

- ***Weather (hidden variable): sunny, cloudy, rainy***
- ***Observed variables** are the **type of clothing worn***

The arrows represent:

- *Transitions from a hidden state to another hidden state*
- *Transitions from a hidden state to an observed variable*



One or more observations allow us to make an inference about a sequence of hidden states

Hidden Markov Model (HMM)

Predicting the **weather** ***(hidden variable)*** based on the type of ***clothes that someone wears (observed)***

In order to compute the joint probability of a sequence of hidden states, we need to assemble three types of information:

1. ***Initial state information*** *(a.k.a.* ***prior probability****) - The initial probability of transitioning to a hidden state.*
2. ***Transition data*** *— the probability of transitioning to a new state conditioned on a present state*
3. ***Emission data*** *– the probability of transitioning to an observed state conditioned on a hidden state*

Priors

Rainy	0.6
Cloudy	0.3
Sunny	0.1

Transitions

		Tomorrow		
		Rainy	Cloudy	Sunny
Today	Rainy	0.6	0.3	0.1
	Cloudy	0.4	0.3	0.2
	Sunny	0.1	0.4	0.5

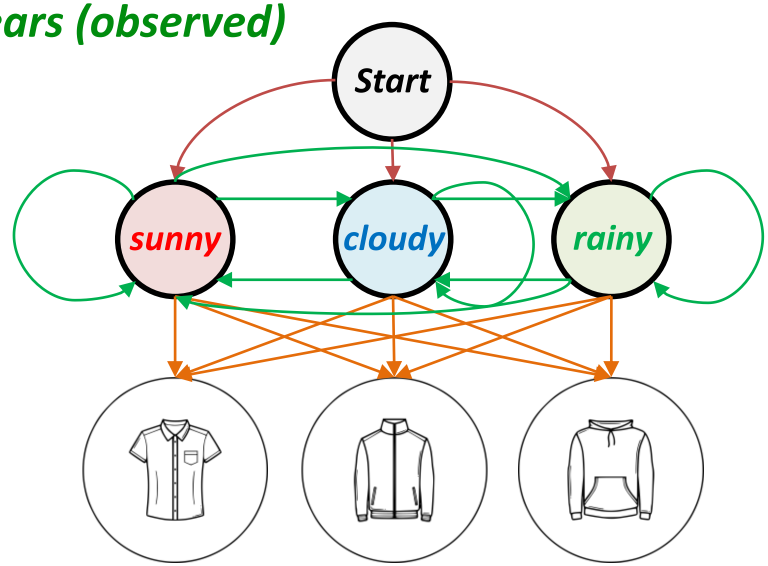
Emissions

	Rainy	Cloudy	Sunny
Shirts	0.8	0.19	0.01
Jacket	0.5	0.4	0.1
Hoodies	0.01	0.2	0.79

3 Probabilistic Approaches

Hidden Markov Model (HMM)

Predicting the **weather *(hidden variable)*** based on the type of ***clothes that someone wears (observed)***



Priors

Rainy	0.6
Cloudy	0.3
Sunny	0.1

Transitions

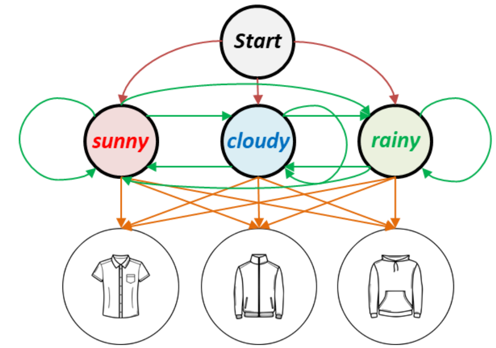
		Tomorrow		
		Rainy	Cloudy	Sunny
Today	Rainy	0.6	0.3	0.1
	Cloudy	0.4	0.3	0.2
	Sunny	0.1	0.4	0.5

Emissions

	Rainy	Cloudy	Sunny
Shirts	0.8	0.19	0.01
Jacket	0.5	0.4	0.1
Hoodies	0.01	0.2	0.79

3 Probabilistic Approaches

Hidden Markov Model (HMM)



We had the list of clothes that Caren wears for three days

Shirts ***Hoodie*** ***Hoodie***

Assume the weather was 'cloudy – cloudy – sunny '

1. Calculate the probability that Caren could wear that clothing (with the weather condition 'cloudy – cloudy – sunny')

*P(shirts|cloudy)*P(hoodie|cloudy)*P(hoodie|sunny)*

2. Calculate the probability that weathers were 'cloudy – cloudy – sunny'

P(prior_cloudy) P(cloudy|cloudy) * P(sunny|cloudy)*

*P(shirts|cloudy)*P(hoodie|cloudy)*P(hoodie|sunny)* P(prior_cloudy)* P(cloudy|cloudy) * P(sunny|cloudy)*

This is the probability when we assume the weather (cloudy – cloudy – sunny – rainy')

Hidden Markov Model (HMM)

The previous was the only probability when we assume the weather (cloudy – cloudy – sunny – rainy')

This is a complete set of 3^3=27 cases of weather states for three days:

{x1=s1=sunny, x2=s1=sunny, x3=s1=sunny}, {x1=s1=sunny, x2=s1=sunny, x3=s2=cloudy}, {x1=s1=sunny, x2=s1=sunny, x3=s3=rainy}, {x1=s1=sunny, x2=s2=cloudy, x3=s1=sunny}, {x1=s1=sunny, x2=s2=cloudy, x3=s2=cloudy}, {x1=s1=sunny, x2=s2=cloudy, x3=s3=rainy}, {x1=s1=sunny, x2=s3=rainy, x3=s1=sunny}, {x1=s1=sunny, x2=s3=rainy, x3=s2=cloudy}, {x1=s1=sunny, x2=s3=rainy, x3=s3=rainy}, {x1=s2=cloudy, x2=s1=sunny, x3=s1=sunny}, {x1=s2=cloudy, x2=s1=sunny, x3=s2=cloudy}, {x1=s2=cloudy, x2=s1=sunny, x3=s3=rainy}, {x1=s2=cloudy, x2=s2=cloudy, x3=s1=sunny}, {x1=s2=cloudy, x2=s2=cloudy, x3=s2=cloudy}, {x1=s2=cloudy, x2=s2=cloudy, x3=s3=rainy}, {x1=s2=cloudy, x2=s3=rainy, x3=s1=sunny}, {x1=s2=cloudy, x2=s3=rainy, x3=s2=cloudy}, {x1=s2=cloudy, x2=s3=rainy, x3=s3=rainy}, {x1=s3=rainy, x2=s1=sunny, x3=s1=sunny}, {x1=s3=rainy, x2=s1=sunny, x3=s2=cloudy}, {x1=s3=rainy, x2=s1=sunny, x3=s3=rainy}, {x1=s3=rainy, x2=s2=cloudy, x3=s1=sunny}, {x1=s3=rainy, x2=s2=cloudy, x3=s2=cloudy}, {x1=s3=rainy, x2=s2=cloudy, x3=s3=rainy}, {x1=s3=rainy, x2=s3=rainy, x3=s1=sunny}, {x1=s3=rainy, x2=s3=rainy, x3=s2=cloudy}, {x1=s3=rainy, x2=s3=rainy, x3=s3=rainy}.

Easy but slow solution: Exhaustive enumeration!

HMM: The two main questions

Evaluation

- *What is the* ***probability*** *that the observations were generated by a given model?*

Decoding

- *Given a model and a sequence of observations, what is* ***the most likely state observation****?*

Hidden Markov Model (HMM): Evaluation

Do we need to calculate this much all the time?

This is a complete set of 3^3=27 cases of weather states for three days:

{x1=s1=sunny, x2=s1=sunny, x3=s1=sunny}, {x1=s1=sunny, x2=s1=sunny, x3=s2=cloudy},
{x1=s1=sunny, x2=s1=sunny, x3=s3=rainy}, {x1=s1=sunny, x2=s2=cloudy, x3=s1=sunny},
{x1=s1=sunny, x2=s2=cloudy, x3=s2=cloudy}, {x1=s1=sunny, x2=s2=cloudy, x3=s3=rainy},
{x1=s1=sunny, x2=s3=rainy, x3=s1=sunny}, {x1=s1=sunny, x2=s3=rainy, x3=s2=cloudy},
{x1=s1=sunny, x2=s3=rainy, x3=s3=rainy}, {x1=s2=cloudy, x2=s1=sunny, x3=s1=sunny},
{x1=s2=cloudy, x2=s1=sunny, x3=s2=cloudy}, {x1=s2=cloudy, x2=s1=sunny, x3=s3=rainy},
{x1=s2=cloudy, x2=s2=cloudy, x3=s1=sunny}, {x1=s2=cloudy, x2=s2=cloudy, x3=s2=cloudy},
{x1=s2=cloudy, x2=s2=cloudy, x3=s3=rainy}, {x1=s2=cloudy, x2=s3=rainy, x3=s1=sunny},
{x1=s2=cloudy, x2=s3=rainy, x3=s2=cloudy}, {x1=s2=cloudy, x2=s3=rainy, x3=s3=rainy},
{x1=s3=rainy, x2=s1=sunny, x3=s1=sunny}, {x1=s3=rainy, x2=s1=sunny, x3=s2=cloudy},
{x1=s3=rainy, x2=s1=sunny, x3=s3=rainy}, {x1=s3=rainy, x2=s2=cloudy, x3=s1=sunny},
{x1=s3=rainy, x2=s2=cloudy, x3=s2=cloudy}, {x1=s3=rainy, x2=s2=cloudy, x3=s3=rainy},
{x1=s3=rainy, x2=s3=rainy, x3=s1=sunny}, {x1=s3=rainy, x2=s3=rainy, x3=s2=cloudy},
{x1=s3=rainy, x2=s3=rainy, x3=s3=rainy}.

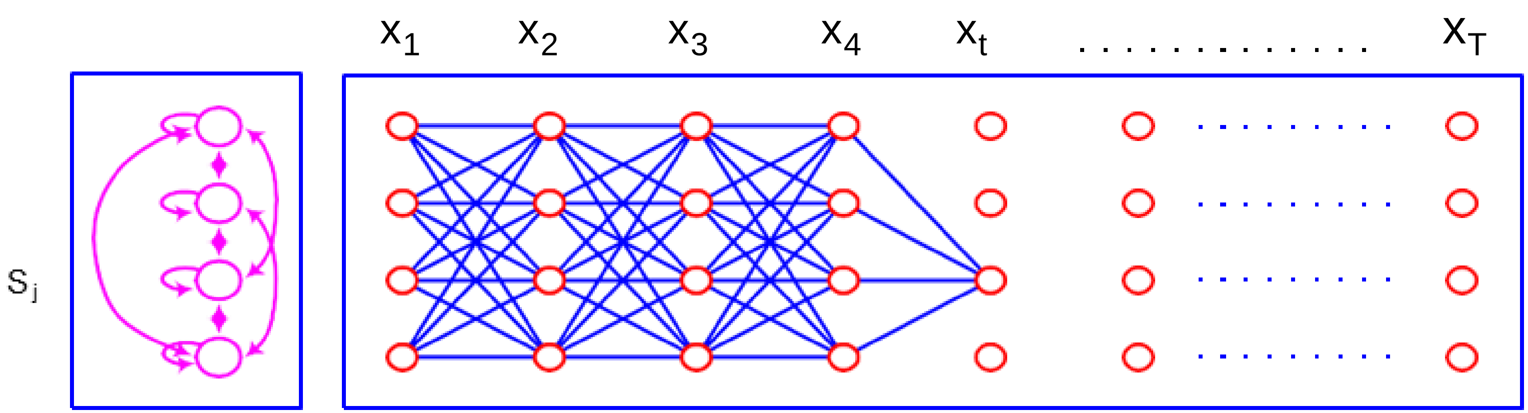
Any Efficient Solution?

Yes: Dynamic Programming technique

HMM Evaluation: Forward Algorithm

Key Idea

- *Span a lattice of N states and T times*
- *Keep the sum of probabilities of all the paths coming to each state i at time t*



Forward Probability

$$\alpha_t(j) = P(x_1 x_2 \ldots x_t, q_t = S_j \mid \lambda)$$

$$= \sum_{Q_t} P(x_1 x_2 \ldots x_t, Q_t = q_1 \ldots q_t \mid \lambda)$$

$$= \sum_{i=1} \alpha_{t-1}(i) a_{ij} b_j(x_t)$$

HMM Evaluation: Forward Algorithm

Initialization

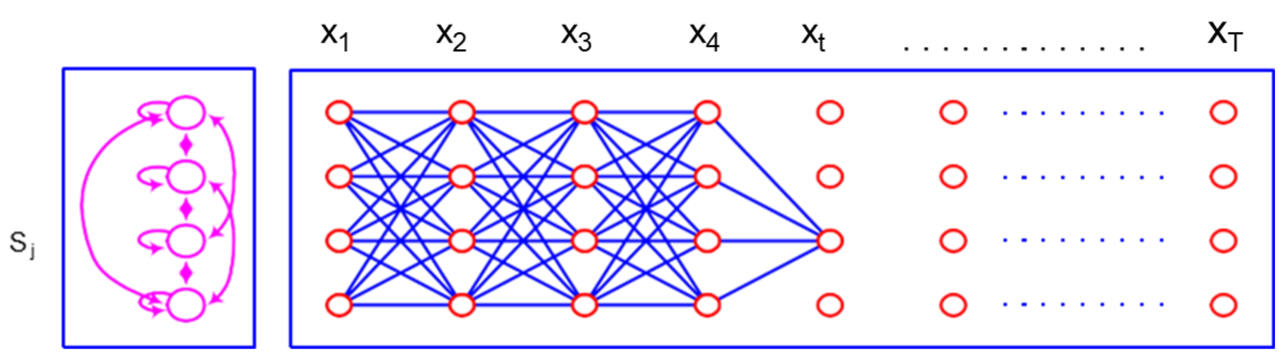
$$\alpha_1(i) = \pi_i b_i(\mathbf{x}_1) \qquad 1 \le i \le N$$

Induction

$$\alpha_t(j) = \sum_{i=1} \alpha_{t-1}(i) a_{ij} b_j(\mathbf{x}_t) \qquad 1 \le j \le N,\ t = 2, 3, \dots,\ T$$

Termination

$$P(\mathbf{X} \mid \lambda) = \sum \alpha_T(i)$$



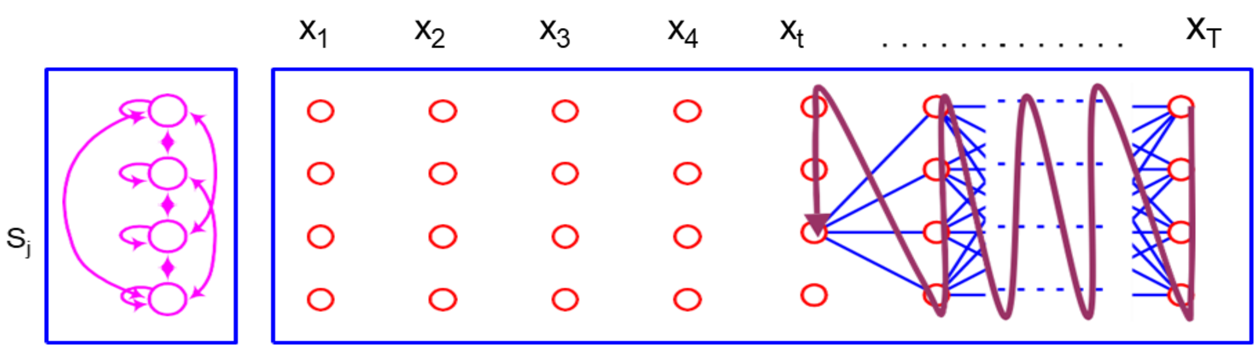
HMM Evaluation: Backward Algorithm

Initialization

$$\beta_T(i) = 1 \qquad 1 \le i \le N$$

Induction

$$\beta_t(i) = \sum a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j) \qquad 1 \le i \le N,\ t = T-1, T-2, \ldots, 1$$



HMM: The two main questions

Evaluation

- *What is the **probability** that the observations were generated by a given model? **[Solved]***

Decoding

- *Given a model and a sequence of observations, what is the most likely state observation?*

HMM Decoding

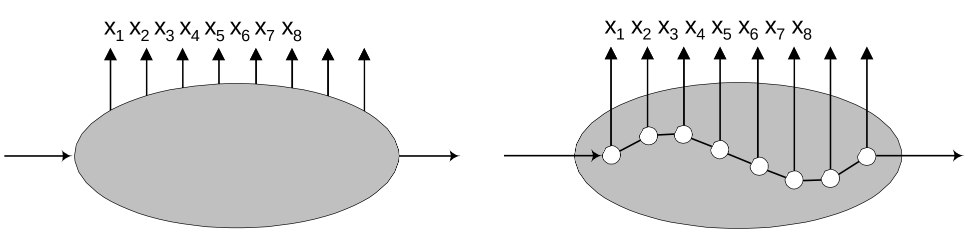
The most likely state observation

- *Given a model* λ
- *Observation sequence:* $X = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$

$$P(X, Q \mid \lambda) = ?$$

$$Q^* = \arg\max_Q P(X, Q \mid \lambda) = \arg\max_Q P(X \mid Q, \lambda) P(Q \mid \lambda)$$

– (A path or state sequence: $Q = q_1, \ldots, q_T$)



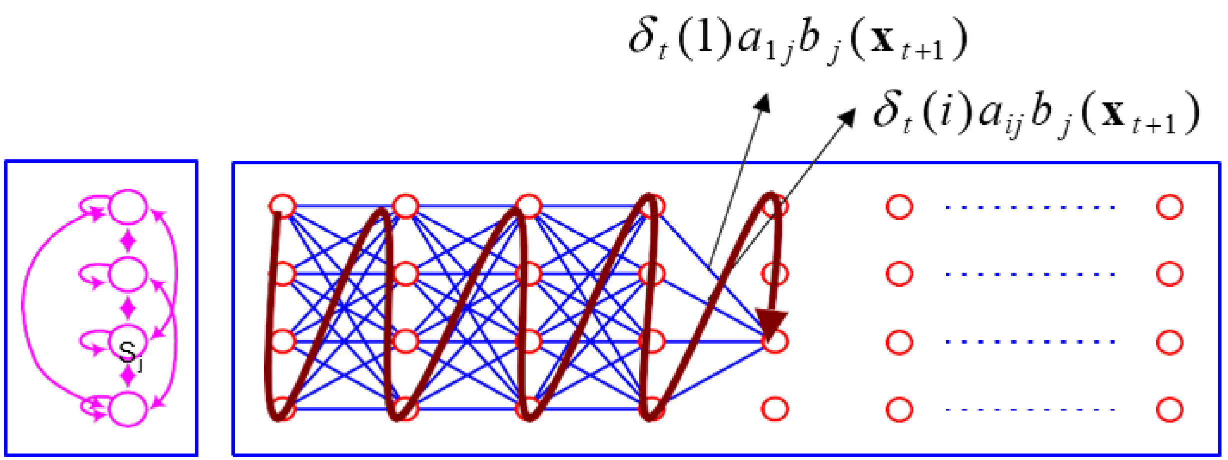
HMM Decoding: Viterbi Path Idea

Key Idea

- *Span a lattice of N states and T times*
- *Keep the probability and the previous node of the most probable path coming to each state i at time t*

Recursive path selection

- *Path probability* $\delta_{t+1}(j) = \max_{1 \le i \le N} \delta_t(i) a_{ij} b_j(\mathbf{x}_{t+1})$
- *Path node* $\psi_{t+1}(j) = \arg\max_{1 \le i \le N} \delta_t(i) a_{ij}$



Viterbi Algorithm

Introduction

$$\delta_1(i) = \pi_i b_i(\mathbf{x}_1)$$
$$\psi_1(i) = 0$$

Recursion

$$\delta_{t+1}(j) = \max_{1 \le i \le N} \delta_t(i) a_{ij} b_j(\mathbf{x}_{t+1})$$
$$\psi_{t+1}(j) = \arg\max_{1 \le i \le N} \delta_t(i) a_{ij}$$

Termination

$$P^* = \max_{1 \le i \le N} \delta_T(i)$$
$$q_T^* = \arg\max_{1 \le i \le N} \delta_T(i)$$

Path Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, \ldots, 1$$

HMM: The two main questions

Evaluation

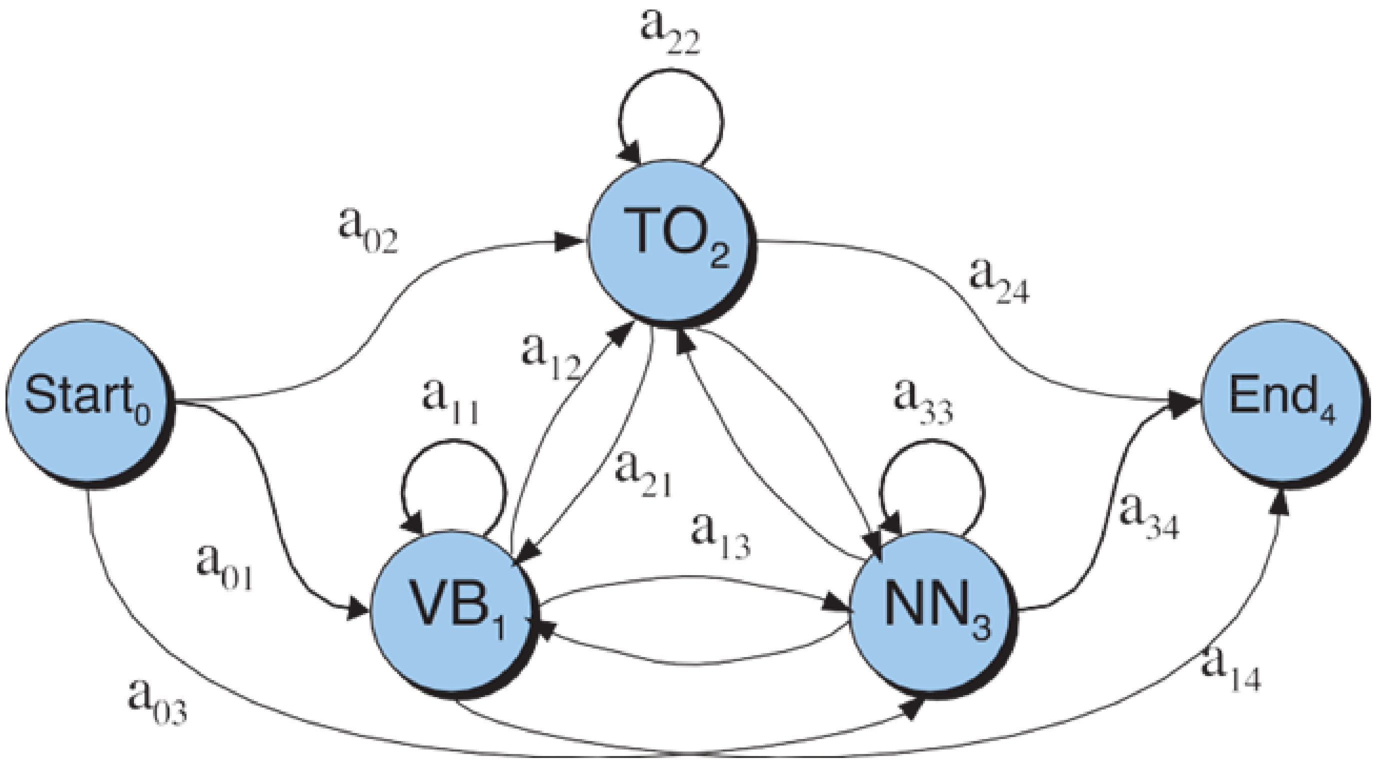
- *What is the **probability** that the observations were generated by a given model? **[Solved]***

Decoding

- *Given a model and a sequence of observations, what is the most likely state observation? **[Solved]***

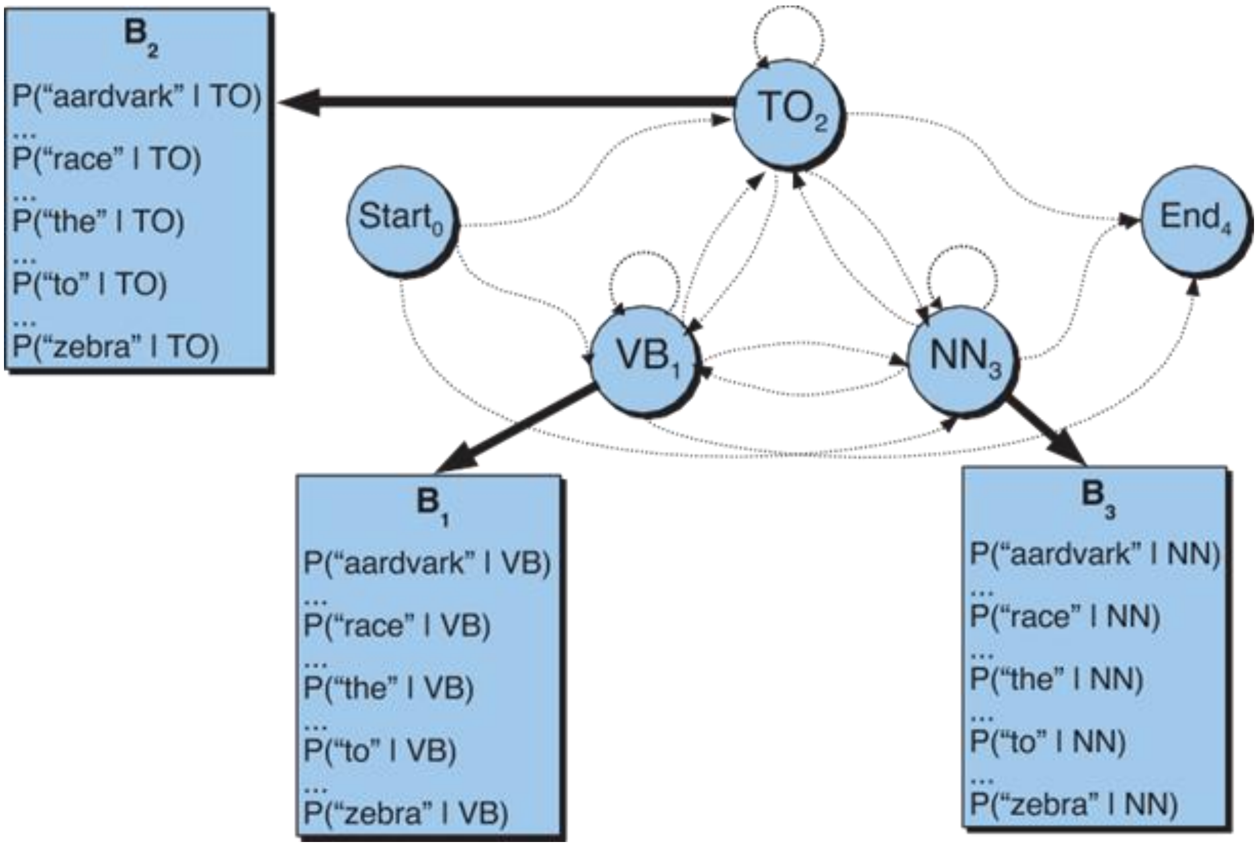
Hidden Markov Model (HMM) with POS Tagging

Transitions



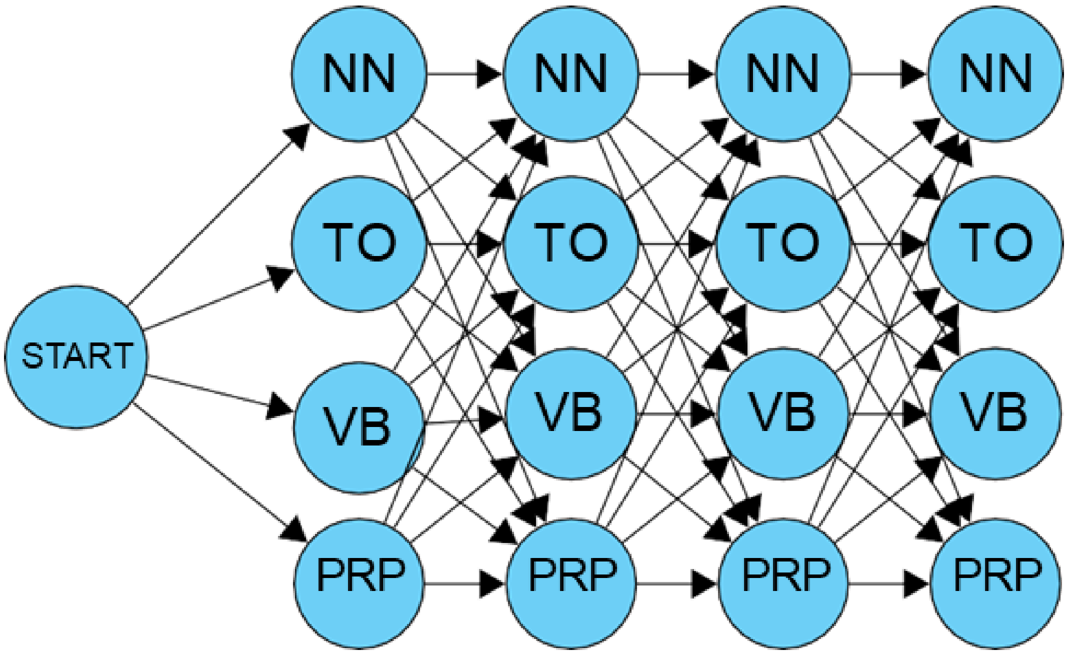
Hidden Markov Model (HMM) with POS Tagging

Emissions



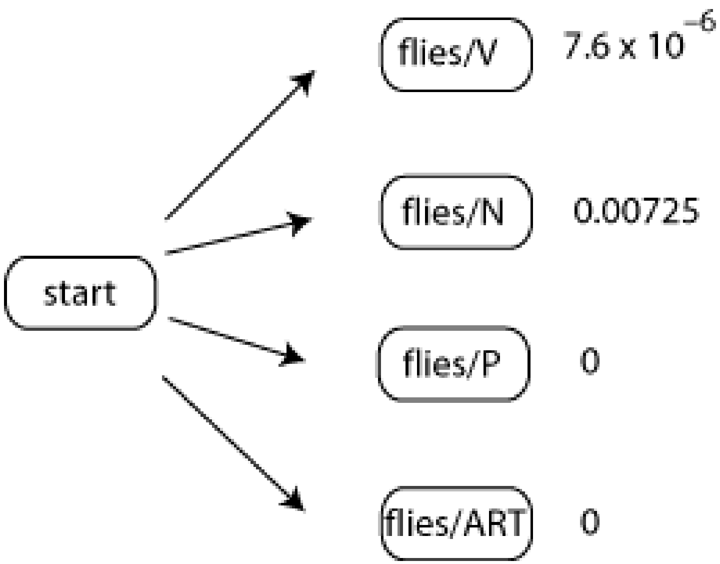
Hidden Markov Model (HMM) with POS Tagging

Trellis



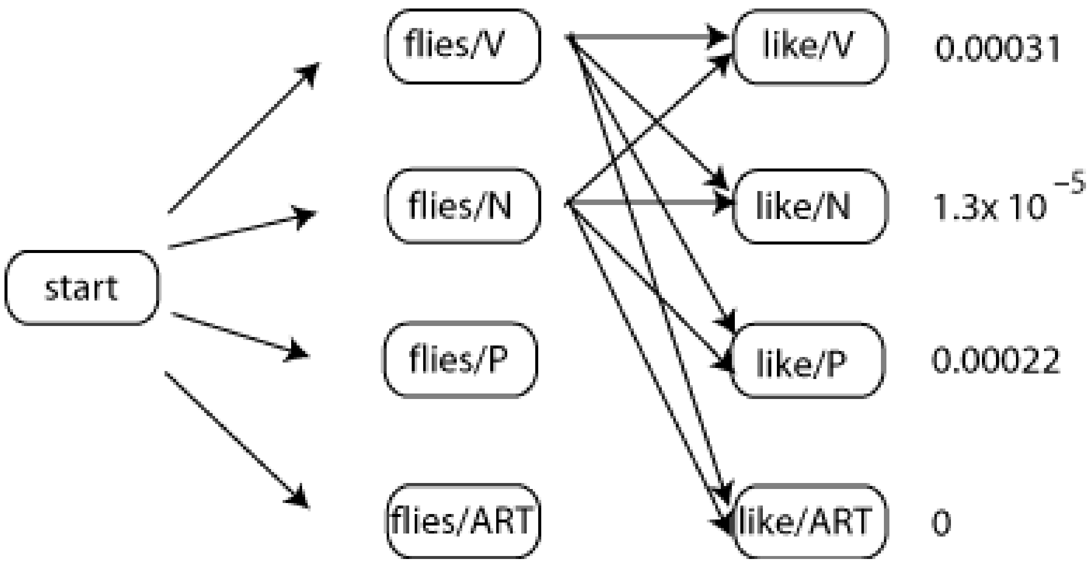
Hidden Markov Model (HMM) with POS Tagging

Simulating the Viterbi Algorithm



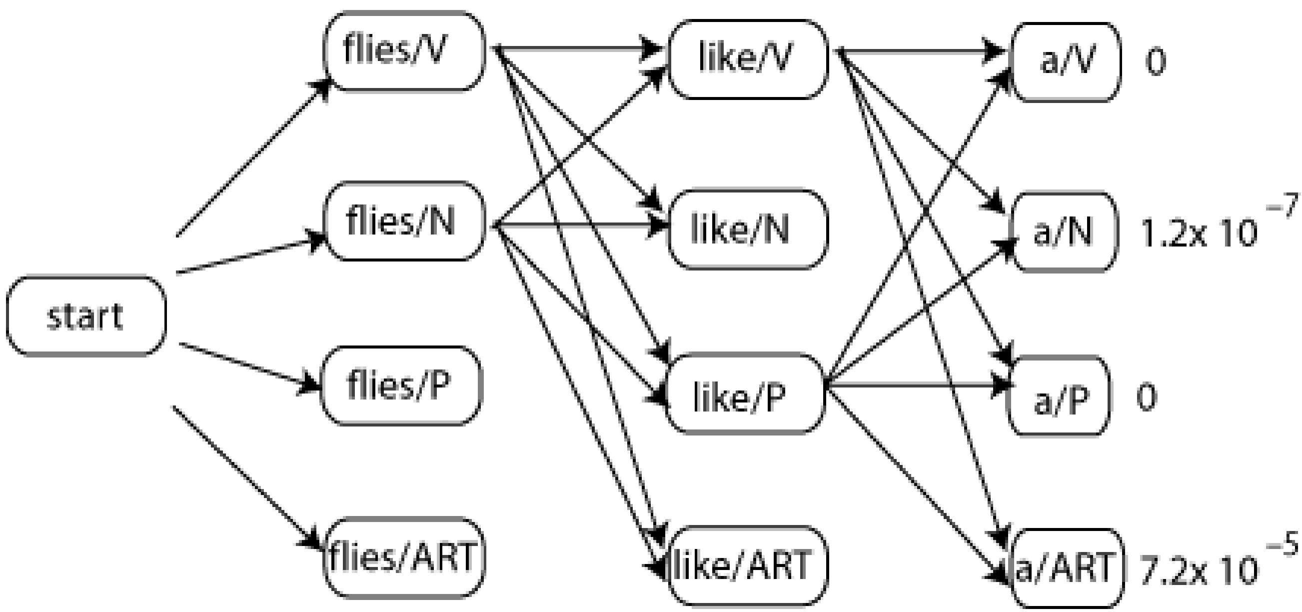
Hidden Markov Model (HMM) with POS Tagging

Simulating the Viterbi Algorithm



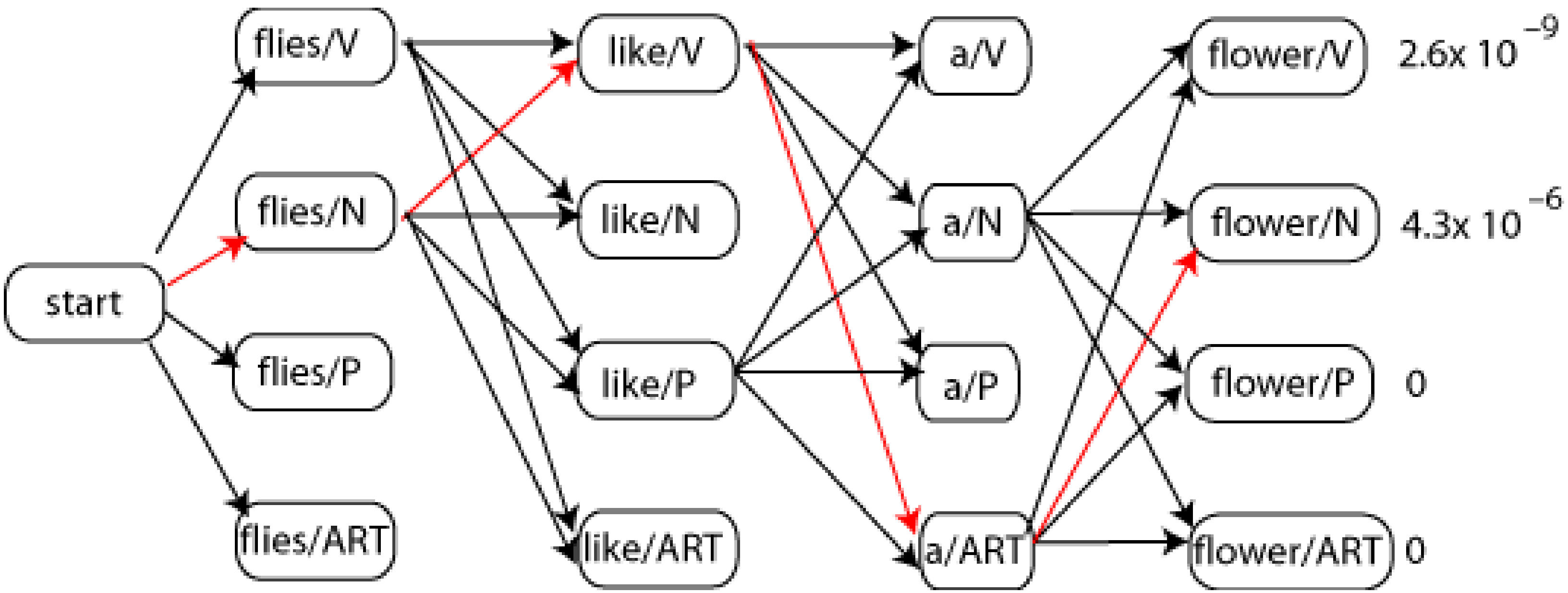
Hidden Markov Model (HMM) with POS Tagging

Simulating the Viterbi Algorithm



Hidden Markov Model (HMM) with POS Tagging

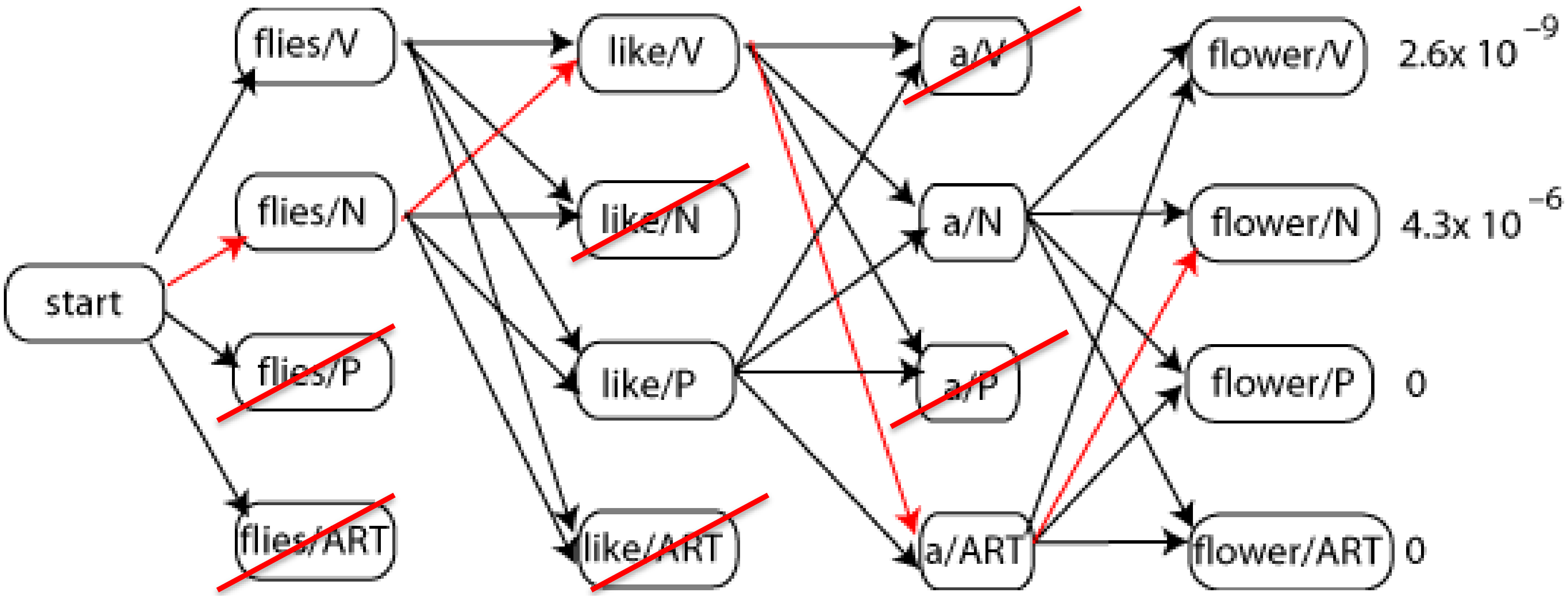
Simulating the Viterbi Algorithm



Hidden Markov Model (HMM) with POS Tagging

Beam Search

Get rid of unlikely candidates



Out-of-Vocab

HMM Tagger Issue: #1. Unknown (OOV) Words

How to handle if there are any unknown words

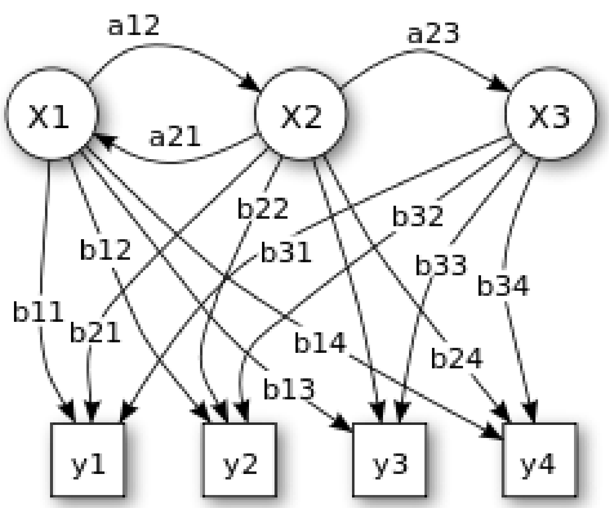
Solution 1: Use N-grams to predict the correct Tag

Solution 2: Use morphology (prefixes, suffixes) or hypenation

THE UNIVERSITY OF SYDNEY

HMM Tagger Issue: #2. Independency Problem

HMM is only dependent on every state and its corresponding observed object. The sequence labeling, in addition to having a relationship with individual words, also relates to such aspects as the observed sequence length, word context and others.



Maximum Entropy Markov Models

- Takes into account the ***dependencies between neighboring states and the entire observed sequence***, hence a better expression ability.
- Does not consider P(X), which reduces the modeling workload and learns the consistency between the target and the estimated function.

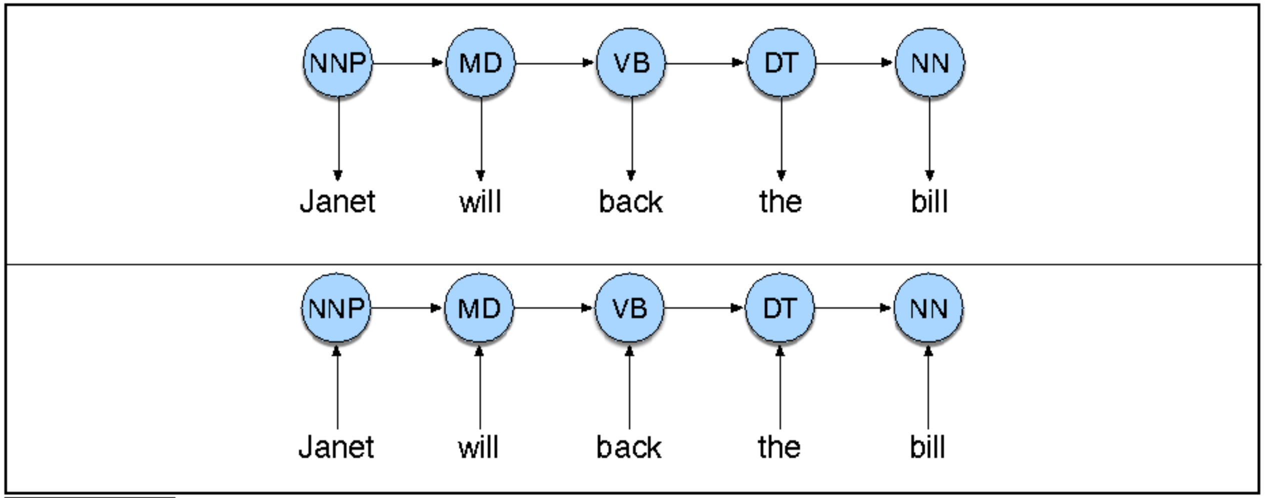
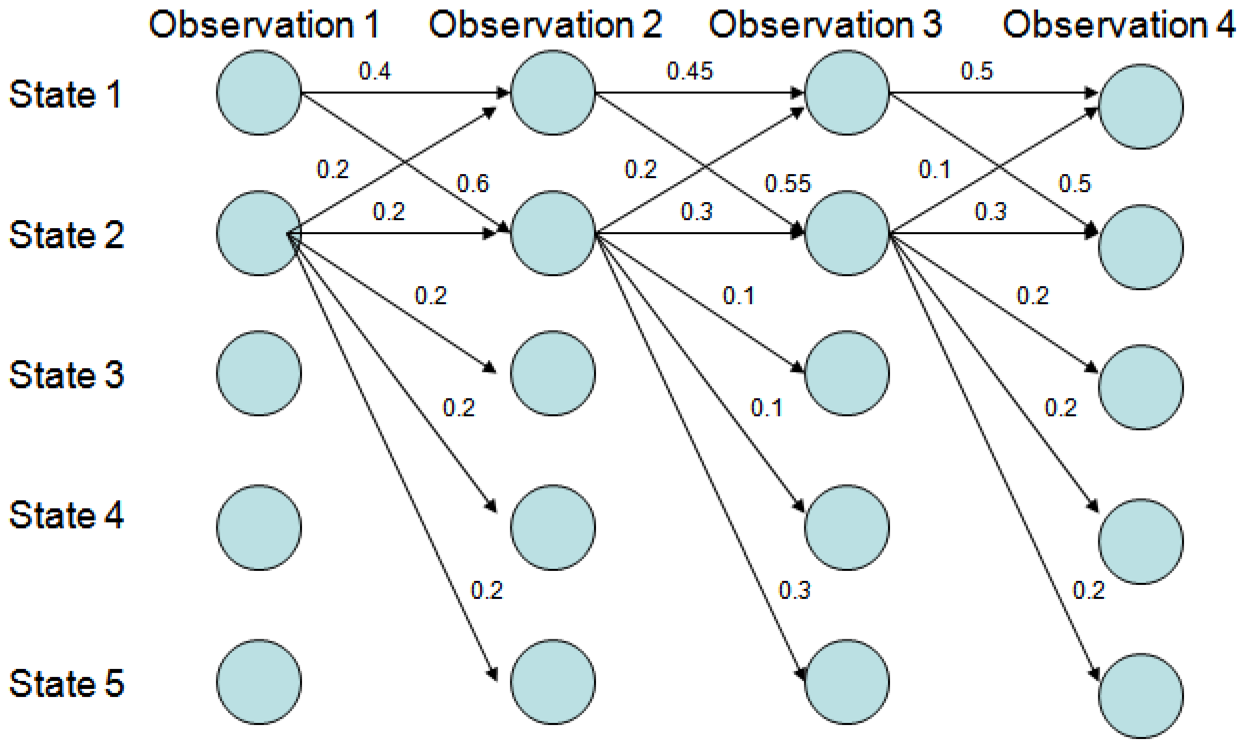
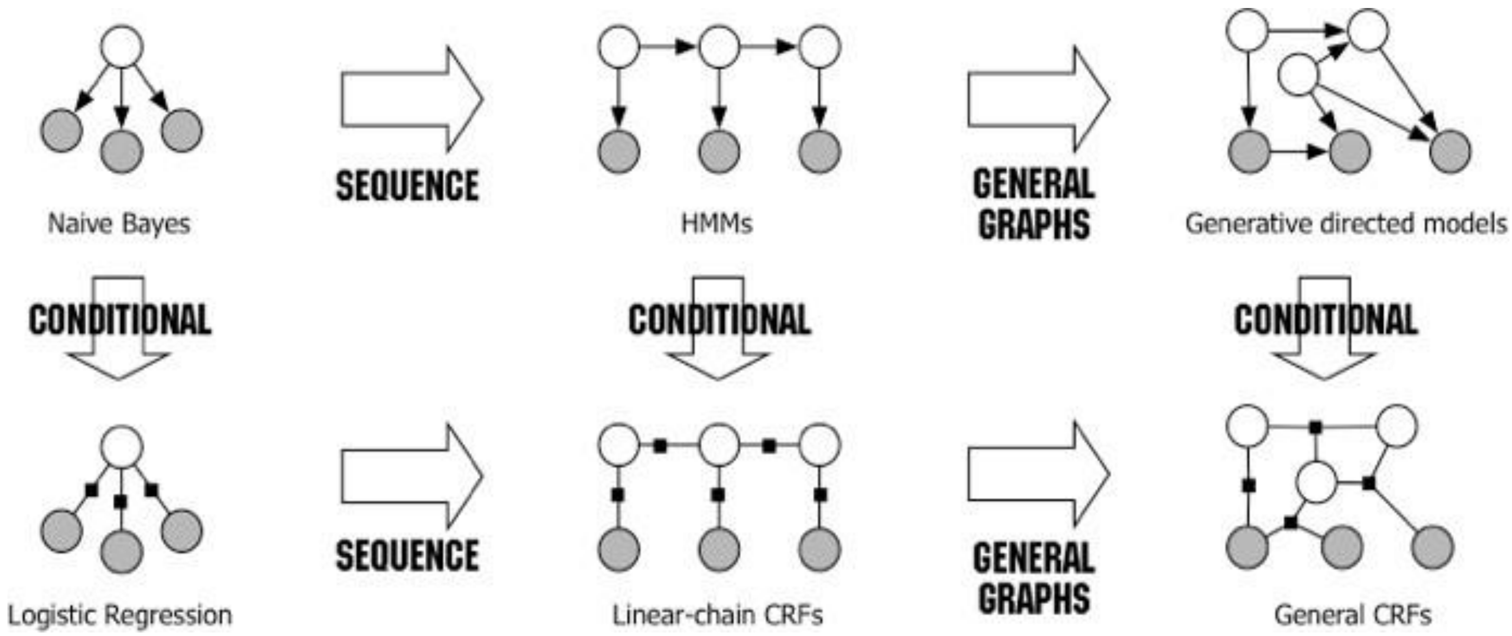


Figure 9.11 A schematic view of the HMM (top) and MEMM (bottom) representation of the probability computation for the correct sequence of tags for the *back* sentence. The HMM computes the likelihood of the observation given the hidden state, while the MEMM computes the posterior of each state, conditioned on the previous state and current observation.

Maximum Entropy Markov Models: Label Bias Issue



Conditional Random Field

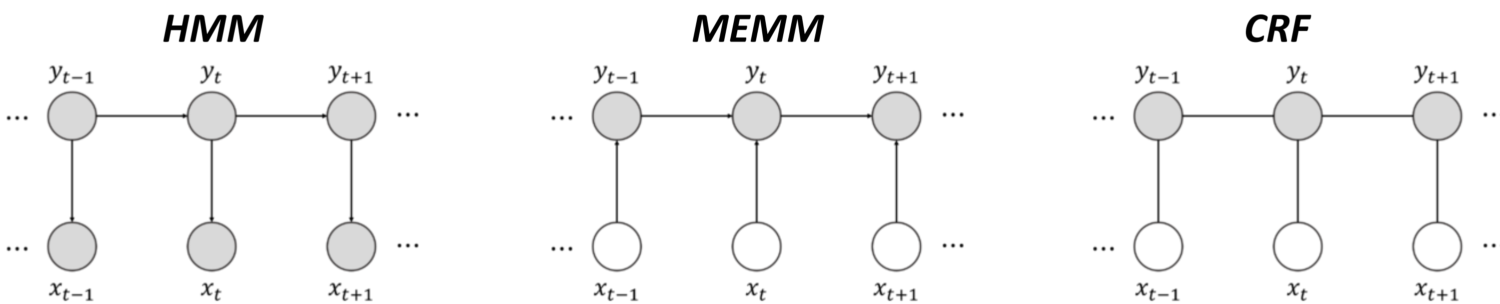


3 Probabilistic Approaches

Conditional Random Field

The CRF model has addressed ***the labeling bias issue*** and ***eliminated unreasonable hypotheses*** in HMM.

MEMM adopts *local variance normalization* while CRF adopts global variance normalization.



Conditional Random Field

Generative Model

- A model that generate observed data randomly
- Model the joint probability p(x,y)

Discriminative model

- Directly estimate the posterior probability p(y|x)
- Aim at modeling the “discrimination” between different outputs

Topological structure

HMM and MEMM are a directed graph, while CRF is an undirected graph.

Conditional Random Field (Comparison)

Global optimum or local optimum

HMM directly models the transition probability and the phenotype probability, and calculates the probability of co-occurrence.

MEMM establishes the probability of co-occurrence based on the transition probability and the phenotype probability. It calculates the conditional probability, and only adopts the local variance normalization, making it easy to fall into a local optimum.

CRF calculates the normalization probability in the global scope, rather than in the local scope as is the case with MEMM. It is an optimal global solution and resolves the labeling bias issue in MEMM.

Conditional Random Field: Advantages

- Compared with HMM: Since CRF does not have as strict independence assumptions as HMM does, it can accommodate any context information.

- Compared with MEMM: Since CRF computes the conditional probability of global optimal output nodes, it overcomes the drawbacks of label bias in MEMM.

However,

CRF is highly ***computationally complex at the training stage*** of the algorithm.
It makes it ***very difficult to re-train the model*** when newer data becomes available.

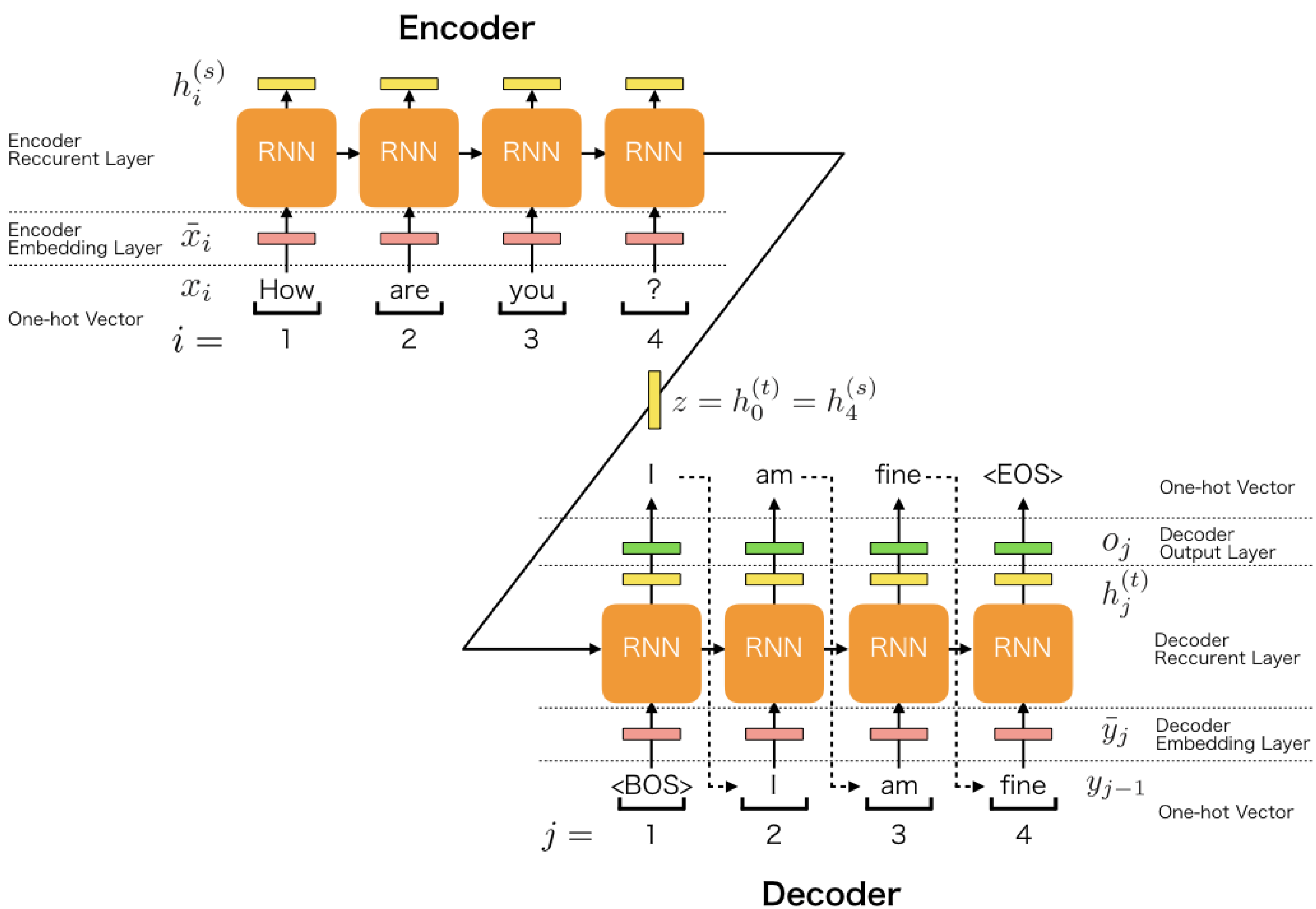
0

LECTURE PLAN

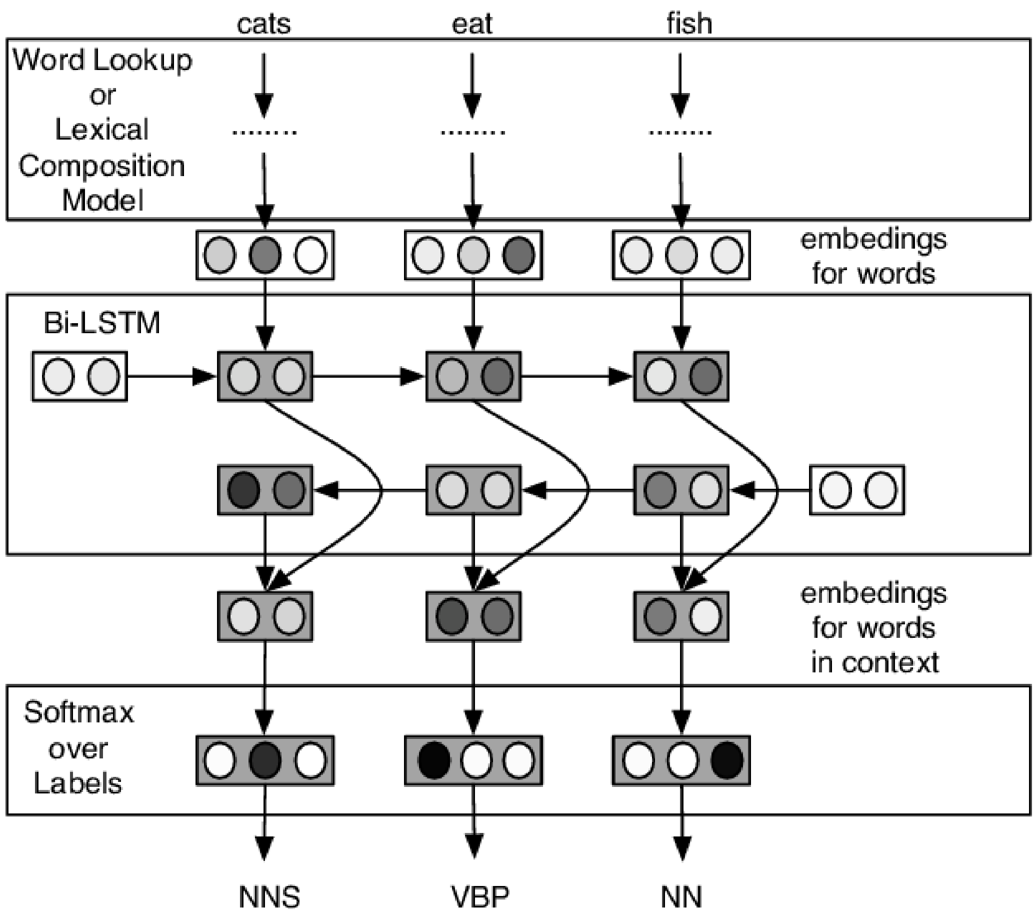
Lecture 6: Part of Speech Tagging

1. Part-of-Speech Tagging
2. Baseline Approaches
   1. Lexicon-based Methods
   2. Rule-based Methods
3. Probabilistic Approaches
   1. Hidden Markov Model
   2. Conditional Random Field
4. **Deep Learning Approaches**

Reminder: RNN/LSTM/GRU

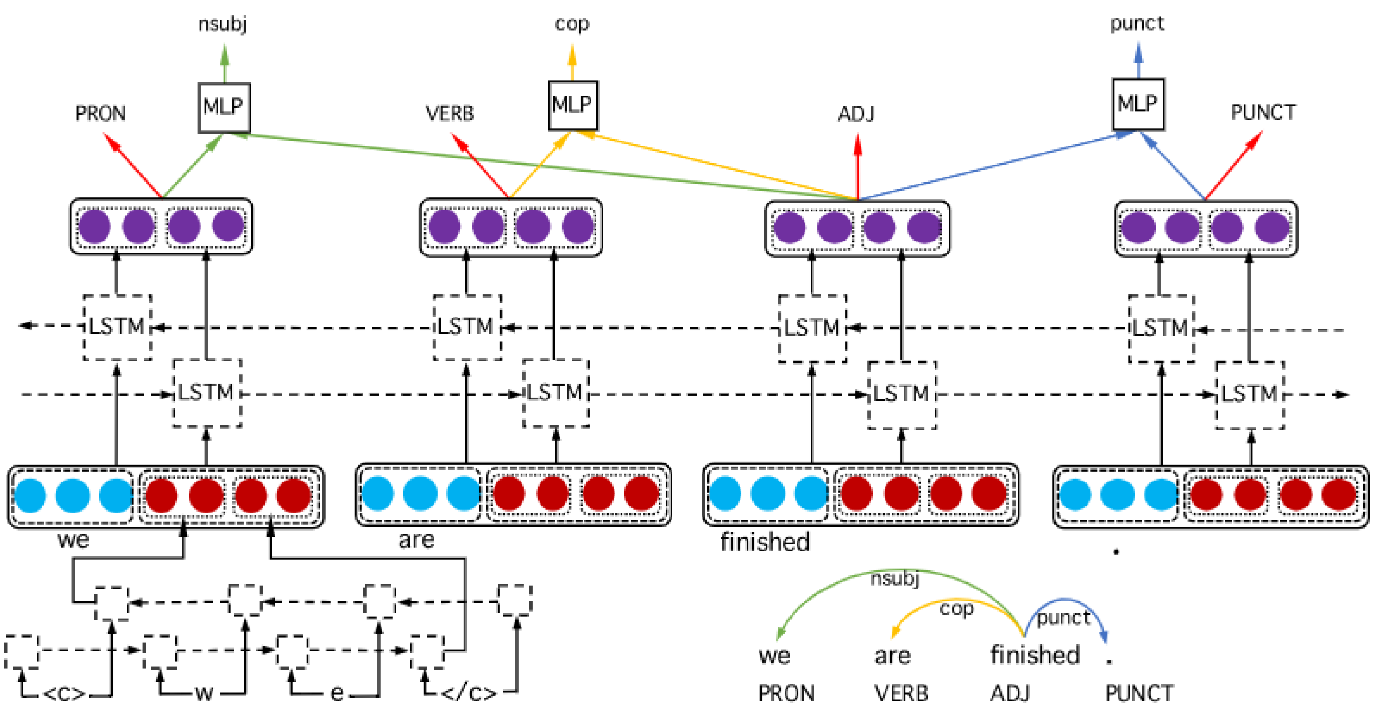


Do LSTMs really work so well for PoS tagging?



LSTM-based POS Tagging

llustration of LSTM-based joint POS tagging and graph-based dependency parsing.



Do LSTMs really work so well for PoS tagging?

(Horsmann and Zesch, 2017)

HMM POS Tagger

Lang. Group	Corpus Id	Word Ngrams ±1 All	Word Ngrams ±1 OOV	Top 750 Char Ngrams All	Top 750 Char Ngrams OOV	Clusters All	Clusters OOV	Best CRF All	Best CRF OOV	HunPos All	HunPos OOV
Germanic	Danish	90.9	53.3	90.3	69.3	89.5	67.6	96.1	82.4	94.9	74.2
	Dutch	86.5	66.9	85.0	71.7	88.0	77.7	90.7	83.7	89.9	80.6
	English	87.5	45.1	90.3	70.1	89.1	64.0	94.6	80.2	93.8	77.7
	German-1	88.5	62.4	90.3	77.7	90.8	73.7	94.6	84.6	94.4	83.7
	German-2	87.2	60.3	90.9	77.7	90.8	76.1	95.2	87.1	94.9	85.4
	German-3	86.3	58.5	91.7	76.8	91.6	77.6	94.4	85.0	94.4	83.9
	Icelandic	67.5	14.2	76.5	45.1	68.3	28.9	80.9	53.6	79.8	51.9
	Norwegian	92.4	77.1	91.6	80.6	92.8	82.7	96.1	89.7	95.5	86.5
	Swedish-1	91.1	70.6	92.9	82.2	92.3	79.9	96.3	90.3	95.6	85.9
	Swedish-2	78.7	29.7	87.2	67.3	81.4	48.8	91.0	74.6	91.4	77.6
Romanic	B-Portug.	86.9	62.8	87.8	73.6	89.7	76.0	92.8	83.8	93.3	84.2
	French-1	81.9	40.1	85.9	66.5	81.6	58.2	89.2	75.7	88.2	71.8
	French-2	95.4	67.3	93.8	74.5	91.9	79.3	97.7	88.2	97.4	82.4
	Italian	93.3	68.6	91.6	74.8	91.7	75.5	96.4	86.5	95.8	80.8
	Spanish	88.5	45.5	94.5	78.2	88.1	58.8	96.4	83.5	96.6	83.6
Slavic	Croatian-1	69.0	18.6	80.6	56.3	75.2	47.2	84.9	65.4	84.7	66.7
	Croatian-2	66.3	15.9	78.5	54.4	73.5	44.8	83.4	63.9	82.6	63.9
	Czech	64.1	14.4	79.2	56.0	75.2	39.2	83.1	62.9	81.7	60.9
	Polish	82.9	58.1	92.5	86.9	86.5	72.5	95.5	91.5	93.6	85.4
	Russian	83.7	53.7	93.0	83.5	88.2	70.9	95.5	87.5	94.6	83.6
	Slovak	67.7	14.9	80.5	57.8	65.6	31.9	83.5	63.8	82.9	61.6
	Slovene-1	72.6	17.4	83.5	55.6	72.4	39.4	86.4	62.5	82.6	59.6
	Slovene-2	65.4	12.1	78.2	50.5	73.0	39.0	83.0	59.4	86.2	59.5
Other	Afrikaans	95.7	75.0	95.3	80.3	95.8	81.9	97.8	89.6	97.3	85.5
	Finnish	62.6	10.0	77.1	48.5	67.8	33.8	82.3	56.7	81.3	55.8
	Hebrew	82.3	41.7	81.3	60.9	76.3	53.3	90.5	68.5	90.3	60.1
	Hungarian	72.7	13.9	86.7	63.3	72.0	31.7	89.9	69.6	89.4	69.5

Table 2: Accuracy of CRF taggers (10fold CV)

Lang. Group	Corpus Id	Word All	Word OOV	Char All	Char OOV	Word-Char All	Word-Char OOV	Word-Char+ All	Word-Char+ OOV	HunPos All	HunPos OOV
Germanic	Danish	94.9	72.7	95.0	79.1	96.4	82.5	96.9	83.4	94.9	74.2
	Dutch	91.1	82.3	90.3	83.6	91.6	85.7	92.5	87.1	89.9	80.6
	English	91.9	65.9	92.3	77.4	94.1	79.6	94.9	80.9	93.8	77.7
	German-1	93.6	78.3	94.1	84.5	95.6	87.6	96.0	88.3	94.4	83.7
	German-2	94.5	82.4	94.6	87.1	96.4	90.1	96.8	91.5	94.4	85.4
	German-3	93.8	80.3	94.0	84.9	95.8	88.6	96.4	89.8	94.4	83.9
	Icelandic	76.0	34.8	76.5	49.3	81.8	56.2	84.1	60.6	79.8	51.9
	Norwegian	95.8	86.2	95.7	88.2	96.6	90.3	96.9	90.3	95.5	86.5
	Swedish-1	94.9	81.4	95.3	86.7	96.2	89.0	96.7	89.8	95.6	85.9
	Swedish-2	86.5	54.3	88.9	74.3	91.8	78.5	92.5	80.4	91.4	77.6
Romanic	B-Portug.	93.3	82.4	93.9	87.4	95.0	90.3	95.1	90.8	93.3	84.2
	French-1	87.6	67.0	85.8	72.0	88.7	77.4	89.7	78.7	88.2	71.8
	French-2	97.5	80.4	97.4	83.4	98.1	87.7	98.3	88.7	97.4	82.4
	Italian	96.0	81.3	95.6	84.2	96.5	85.9	97.1	86.9	95.8	80.8
	Spanish	93.1	63.3	96.4	85.5	96.9	86.1	97.2	87.0	96.6	83.6
Slavic	Croatian-1	83.2	55.5	83.8	67.5	88.1	72.8	89.1	75.2	84.7	66.9
	Croatian-2	80.3	52.4	81.1	63.8	84.9	69.1	86.8	72.4	82.6	63.9
	Czech	79.4	49.1	81.0	62.7	85.8	68.7	87.7	72.4	81.7	60.9
	Polish	86.9	73.6	89.2	84.7	95.5	91.2	91.2	88.0	93.6	85.4
	Russian	91.3	73.2	94.6	85.8	95.3	86.9	96.0	88.4	94.6	83.6
	Slovak	78.7	44.9	80.6	65.0	85.3	69.7	86.6	71.4	82.9	61.6
	Slovene-1	81.9	44.5	83.9	61.1	86.0	62.6	87.9	65.7	82.6	59.6
	Slovene-2	79.9	47.9	82.0	63.4	85.8	67.4	87.5	70.1	86.2	59.5
Other	Afrikaans	97.3	82.8	97.1	85.8	97.8	88.4	98.0	90.0	97.3	85.5
	Finnish	76.7	42.7	78.0	57.6	82.0	58.9	83.6	61.2	81.3	55.8
	Hebrew	89.9	60.2	89.2	66.9	92.2	69.7	92.9	72.1	90.3	60.1
	Hungarian	84.7	53.3	88.0	73.1	91.2	76.9	92.0	79.0	89.4	69.5

Table 3: Accuracy of LSTM taggers (10fold CV)

Reference

Reference for this lecture

- Deng, L., & Liu, Y. (Eds.). (2018). Deep Learning in Natural Language Processing. Springer.
- Rao, D., & McMahan, B. (2019). Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning. " O'Reilly Media, Inc.".
- Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.
- Parikh, A 2018, Statistical NLP, lecture notes, New York University
- Cohen, S 2018, Processing Formal and Natural Languages, lecture notes, The University of Edinburgh
- Manning, C 2017, Natural Language Processing with Deep Learning, lecture notes, Stanford University
- M. Marcus, B. Santorini and M.A. Marcinkiewicz (1993). Building a large annotated corpus of English: The Penn Treebank. In Computational Linguistics, volume 19, number 2, pp. 313–330.
- Nguyen, D. Q., Dras, M., & Johnson, M. (2017). A novel neural network model for joint pos tagging and graph-based dependency parsing. arXiv preprint arXiv:1705.05952.
- Ling, W., Luís, T., Marujo, L., Astudillo, R. F., Amir, S., Dyer, C., ... & Trancoso, I. (2015). Finding function in form: Compositional character models for open vocabulary word representation. arXiv preprint arXiv:1508.02096.
- Horsmann, T., & Zesch, T. (2017). Do LSTMs really work so well for PoS tagging?–A replication study. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 727-736).