Part-of-Speech Tagging: Hidden Markov Models

Katharina Kann — CSCI/LING5832

Word Classes: Parts of Speech

- Traditional parts of speech:
  - Noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction, etc.
  - Lots of names for this notion: Part of speech, lexical category, word class, lexical tag...
  - Lots of debate within linguistics about the number, nature, and universality of these categories

Word Classes: Parts of Speech

- Sources of evidence:
  - Morphological evidence
    - walk, walking, walked, walks
    - probably a verb

Word Classes: Parts of Speech

- Sources of evidence:
  - Morphological evidence
    - walk, walking, walked, walks
    - probably a verb
  - Distributional evidence
    - The crash, A crash, Two crashes, The big crash…
    - probably a noun

Penn Treebank POS Tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	“to”	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential ‘there’	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one’s*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

POS Tagging

- The process of assigning a part of speech or lexical class marker to each word in a text.
- Often a useful first step in an NLP pipeline.
- Fast and accurate taggers are widely available for many languages.
  - (Now we will learn how to make one!)

POS Tagging

- The process of assigning a part of speech or lexical class marker to each word in a text.
- This is our first example of a **sequence labeling** task:
  - Assigning a category label to each element of a sequence.

POS Tagging

WORD	tag
the	**DET**
koala	**N**
put	**V**
the	**DET**
keys	**N**
on	**P**
the	**DET**
table	**N**

POS Tagging

- Many words have more than one part of speech: back
  - The **back** door = JJ
  - On my **back** = NN
  - Win the voters **back** = RB
  - Promised to **back** the bill = VB

POS Tagging

- Many words have more than one part of speech: back
  - The **back** door = JJ
  - On my **back** = NN
  - Win the voters **back** = RB
  - Promised to **back** the bill = VB
- The POS tagging problem is to determine the tag for a particular instance of a word **in context**.
  - The context is usually a sentence.

POS Tagging

- Note this is *distinct* from the task of word sense disambiguation.
    - “…**backed** the car into a pole”
    - “…**backed** the wrong candidate”

Measuring Ambiguity

		87-tag Original Brown		45-tag Treebank Brown	
Unambiguous (1 tag)		**44,019**		**38,857**	
Ambiguous (2–7 tags)		**5,490**		**8844**	
Details:	2 tags	4,967		6,731	
	3 tags	411		1621	
	4 tags	91		357	
	5 tags	17		90	
	6 tags	2	(*well*, *beat*)	32	
	7 tags	2	(*still*, *down*)	6	(*well*, *set*, *round*, *open*, *fit*, *down*)
	8 tags			4	(*'s*, *half*, *back*, *a*)
	9 tags			3	(*that*, *more*, *in*)

Methods for POS Tagging

- Rule-based tagging
- Probabilistic sequence models
    - HMM (hidden Markov model) tagging
    - RNNs (recurrent neural networks)
    - [Transformers]

Methods for POS Tagging

- Rule-based tagging
- Probabilistic sequence models
  - HMM (hidden Markov model) tagging
  - RNNs (recurrent neural networks)
  - [Transformers]

- Trivial baselines can often do well, too

POS Tagging as Sequence Labeling

- We are given a sentence (an “observation” or “sequence of observations”).
    - I should book the flight
- What is the best sequence of tags that corresponds to this sequence of observations?
- Probabilistic view:
    - Consider all possible sequences of tags and assign a probability to each
    - Out of this universe of sequences, choose the tag sequence that is most probable given the observation sequence of our *n* words.

Probabilistic Approach

- We want out of all sequences of n tags $t_1 \ldots t_n$ the single tag sequence such that $P(t_1 \ldots t_n | w_1 \ldots w_n)$ is highest.

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

Probabilistic Approach

- We want out of all sequences of n tags $t_1 \ldots t_n$ the single tag sequence such that $P(t_1 \ldots t_n | w_1 \ldots w_n)$ is highest.

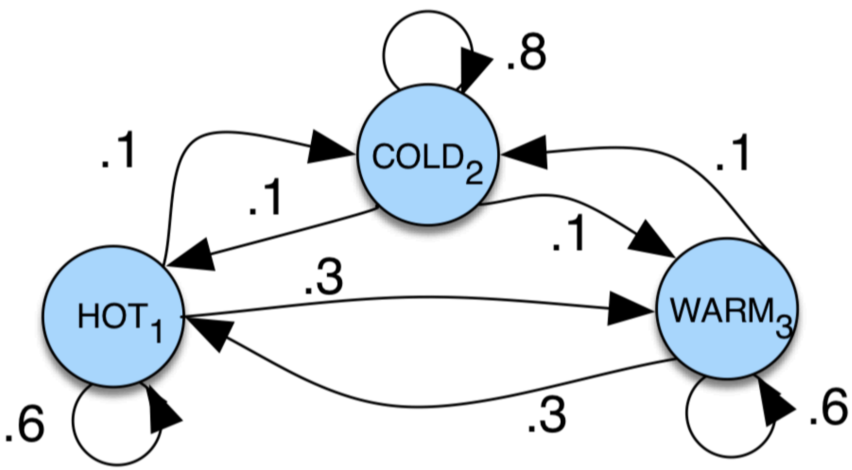
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

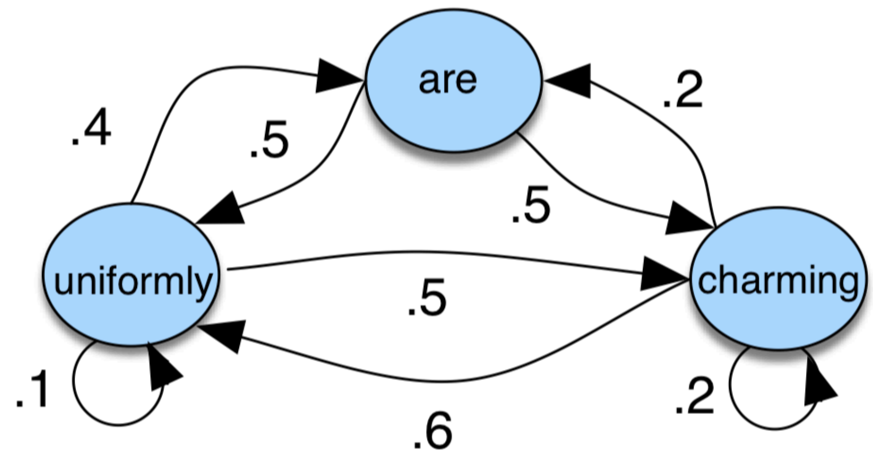
Markov Chains

Markov Chain

- A model that tells us something about the probabilities of sequences of random variables (*states*).
- Each of those variables can take on values from a predefined set.

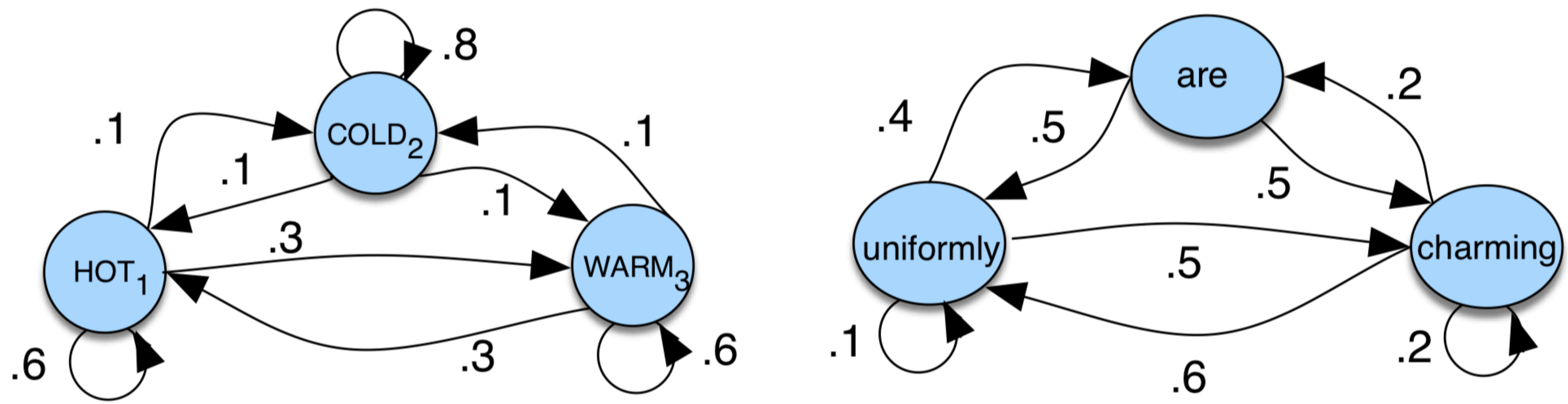
Markov Chain





J&M, Ch. 8

Markov Chain



J&M, Ch. 8

- Markov assumption: only the last state matters!

Markov Chain

$$P(q_1, \ldots q_T) = P(q_T | q_1, \ldots, q_{T-1})$$
$$* P(q_{T-1} | q_1, \ldots, q_{T-2})$$
$$* \ldots$$
$$* P(q_1)$$

Markov assumption:

$$P(q_T | q_1, \ldots, q_{T-1}) = P(q_T | q_{T-1})$$

Markov Chain

$$P(q_1, \ldots q_T) = P(q_T | q_1, \ldots, q_{T-1})$$
$$* P(q_{T-1} | q_1, \ldots, q_{T-2})$$
$$* \ldots$$
$$* \boxed{P(q_1)}$$

?

Markov assumption:

$$P(q_T | q_1, \ldots, q_{T-1}) = P(q_T | q_{T-1})$$

Markov Chain

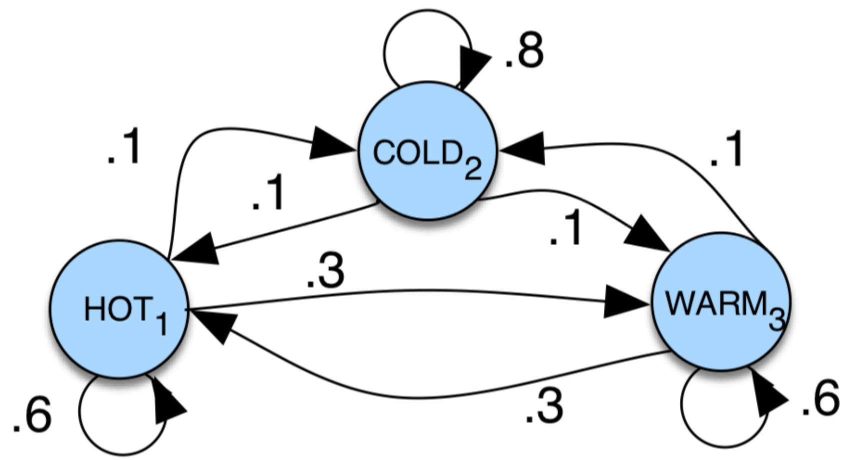
- We further have an **initial probability distribution** over states.
  - The probability that each state is the first state!

Markov Chain

- We further have an **initial probability distribution** over states.
    - The probability that each state is the first state!
- For example: $\pi = [0.1, 0.7, 0.2]$ for the states *hot*, *warm*, and *cold*

In-Class Exercise

- π = [0.1, 0.7, 0.2] for the states *hot*, *warm*, and *cold*



- **Compute the probability of:**
  1. **Warm, hot, warm**
  2. **cold, hot, cold, hot**

Formal Definition: Markov Chain

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, ..., \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

J&M, Ch. 8

Hidden Markov Models

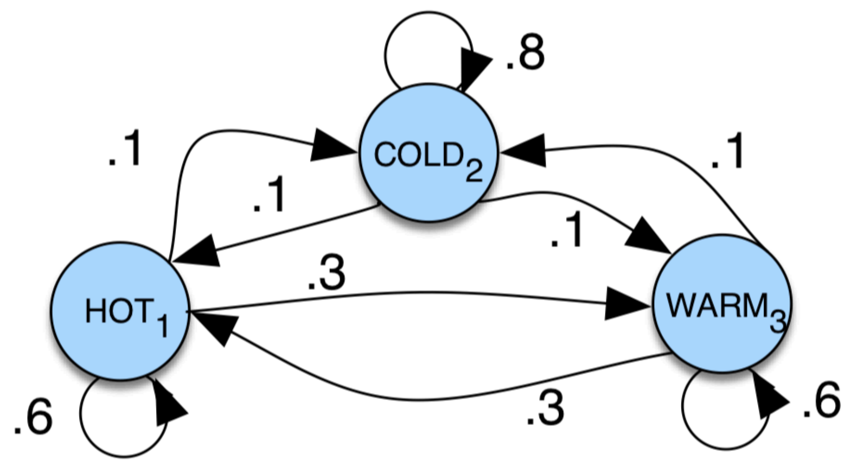
Hidden Markov Model

- Markov chains can compute a probability for a sequence of *observable* events.
- However, we often don't observe the events we are interested in.
  - Hidden Markov models!

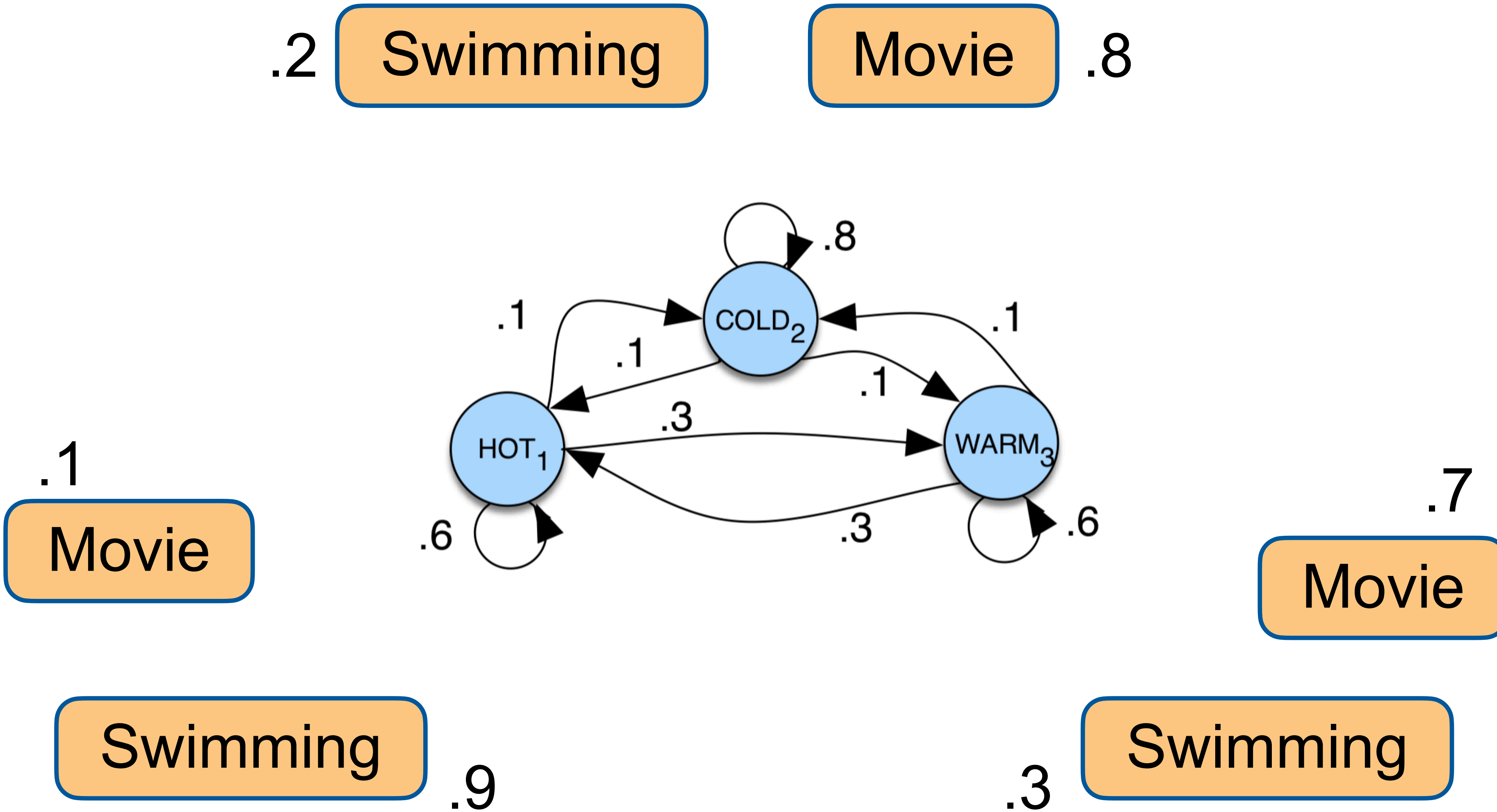
Hidden Markov Model

- Markov chains can compute a probability for a sequence of *observable* events.
- However, we often don't observe the events we are interested in.
    - Hidden Markov models!
- Example: part-of-speech tagging!

Hidden Markov Model



Hidden Markov Model



Hidden Markov Model

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state q_i
$\pi = \pi_1, \pi_2, ..., \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

J&M, Ch. 8

Hidden Markov Model

- A first-order Markov model makes 2 simplifying assumptions:

Hidden Markov Model

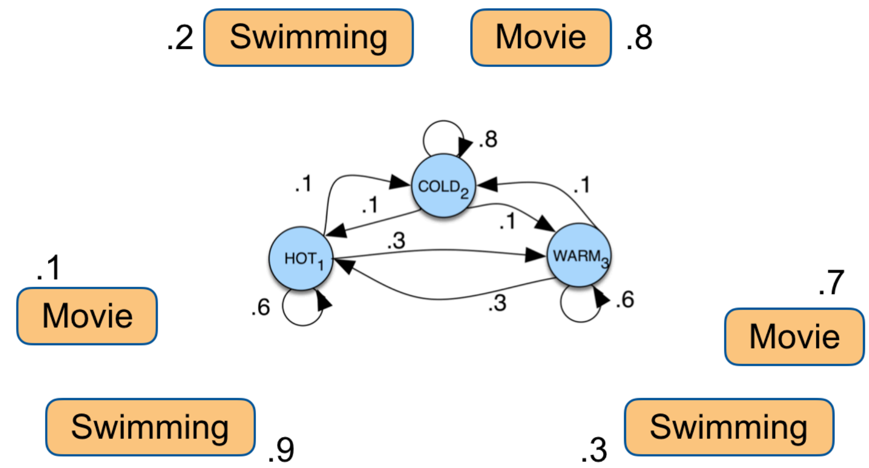
- A first-order Markov model makes 2 simplifying assumptions:
  - Markov assumption:

    $$P(q_i \mid q_1, \ldots, q_{i-1}) = P(q_i \mid q_{i-1})$$

Hidden Markov Model

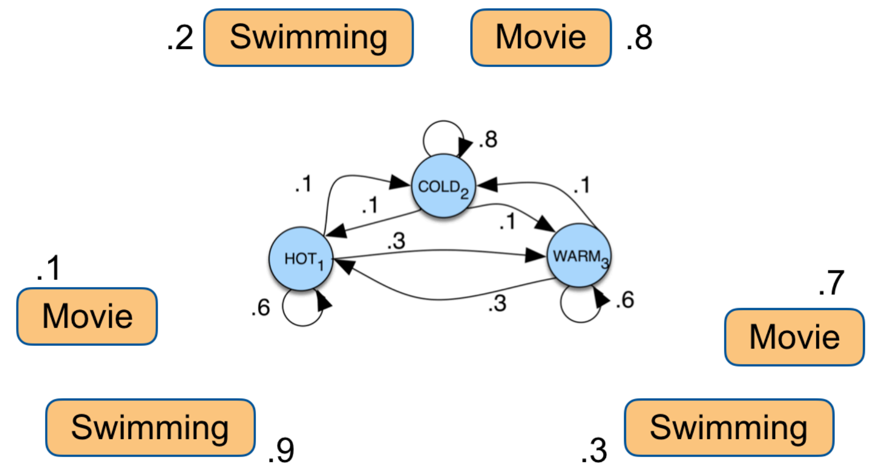
- A first-order Markov model makes 2 simplifying assumptions:
  - Markov assumption: $P(q_i \mid q_1, \ldots, q_{i-1}) = P(q_i \mid q_{i-1})$
  - Output independence: $P(o_i \mid q_1, \ldots, q_T, o_1, \ldots, o_T) = P(o_i \mid q_i)$

Example



- Given:
  - π = [0.1, 0.7, 0.2] for *hot*, *warm*, and *cold*
- Compute the probability of: [cold, warm, Movie, Movie]

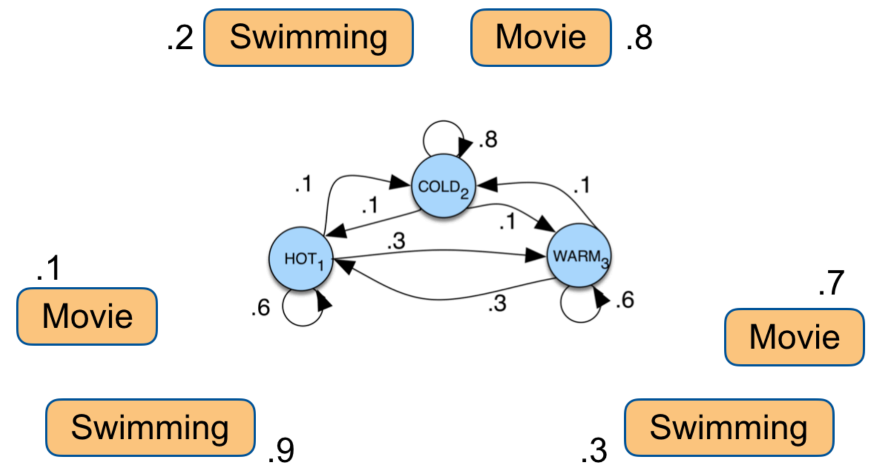
Example



- Given:
  - π = [0.1, 0.7, 0.2] for *hot*, *warm*, and *cold*
- Compute the probability of: [cold, warm, Movie, Movie]

$$P(c, w, M, M) = P(cold) * P(Movie \mid cold) * P(warm \mid cold) * P(Movie \mid warm)$$

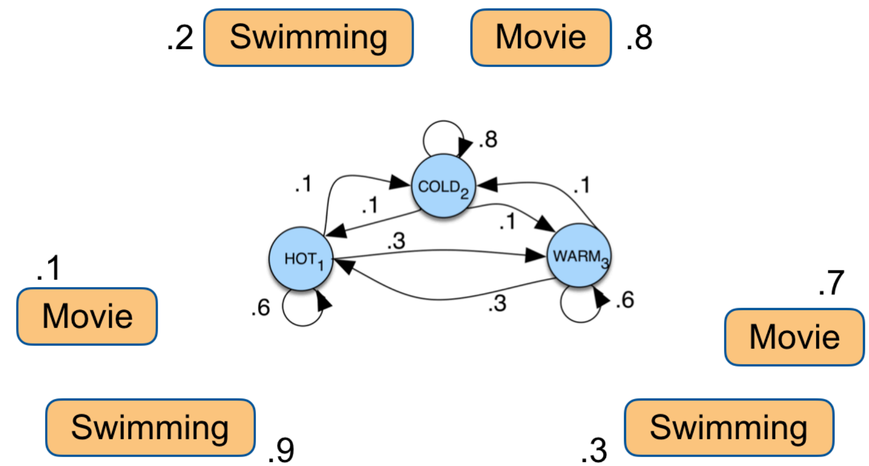
Example



- Given:
  - π = [0.1, 0.7, 0.2] for *hot*, *warm*, and *cold*
- Compute the probability of: [cold, warm, Movie, Movie]

$$P(c, w, M, M) = P(cold) * P(Movie \mid cold) *$$
$$P(warm \mid cold) * P(Movie \mid warm)$$
$$= 0.2 * 0.8 *$$
$$0.1 * 0.7$$

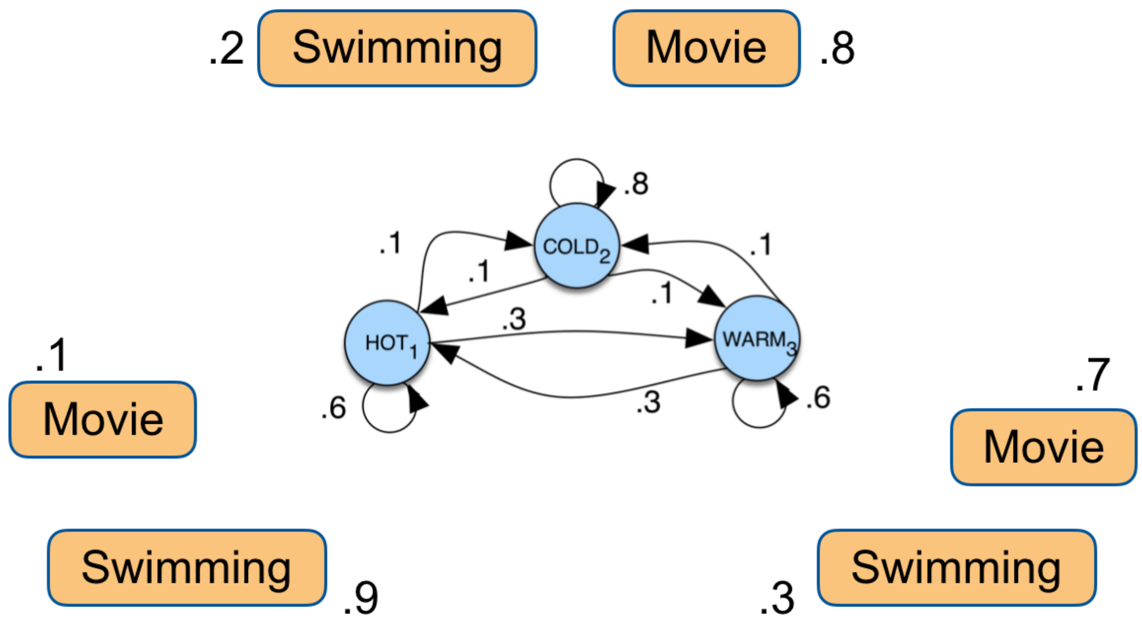
Example



- Given:
  - π = [0.1, 0.7, 0.2] for *hot*, *warm*, and *cold*
- Compute the probability of: [cold, warm, Movie, Movie]

$$
\begin{aligned}
P(c, w, M, M) &= P(cold) * P(Movie \mid cold) * \\
&\quad P(warm \mid cold) * P(Movie \mid warm) \\
&= 0.2 * 0.8 * \\
&\quad 0.1 * 0.7 \\
&= 0.0112
\end{aligned}
$$

In-Class Exercise 2



- Given: π = [0.1, 0.7, 0.2] for the states *hot*, *warm*, and *cold*
- Compute the probability of:
  1. [warm, cold, cold, Swimming, Movie, Movie]
  2. [cold, hot, warm, Swimming, Movie, Movie]
  3. [hot, cold, warm, Swimming, Movie, Movie]

Back to Part-of-Speech Tagging

- Tag transition probabilities $p(t_i|t_{i-1})$
  - Determiners likely to precede adjs and nouns
  - That/DT flight/NN
  - The/DT yellow/JJ hat/NN
  - So we expect P(NN|DT) and P(JJ|DT) to be high
- Compute P(NN|DT) by counting in a labeled corpus: $P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$

Back to Part-of-Speech Tagging

- Tag transition probabilities $p(t_i|t_{i-1})$
  - Determiners likely to precede adjs and nouns
  - That/DT flight/NN
  - The/DT yellow/JJ hat/NN
  - So we expect P(NN|DT) and P(JJ|DT) to be high
- Compute P(NN|DT) by counting in a labeled corpus: $P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$

$$P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = .49$$

Back to Part-of-Speech Tagging

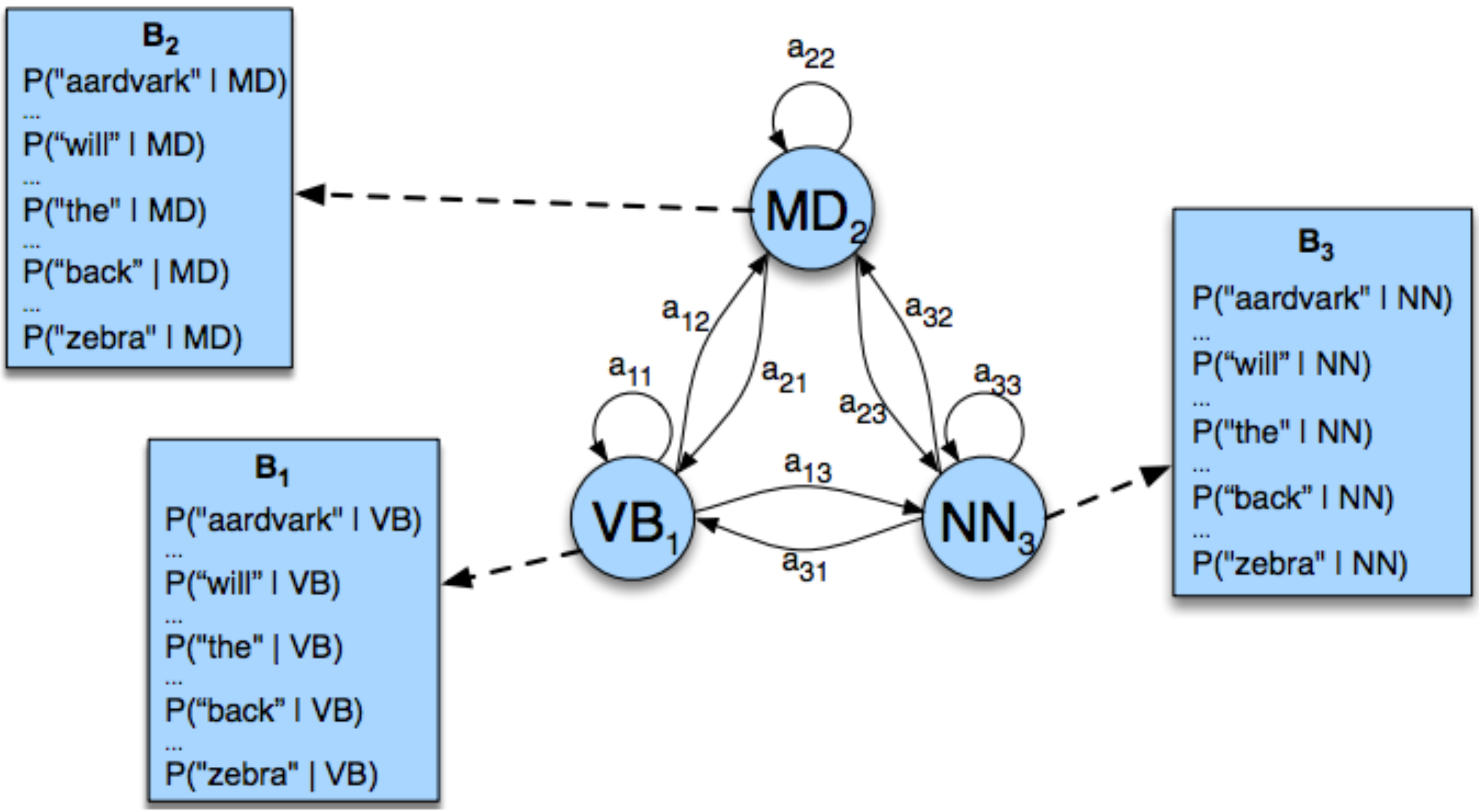
- Word likelihood probabilities $p(w_i|t_i)$
  - VBZ (3sg Pres Verb) likely to be “is”
  - Compute P(is|VBZ) by counting in a labeled corpus: $P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$

Back to Part-of-Speech Tagging

- Word likelihood probabilities $p(w_i|t_i)$
  - VBZ (3sg Pres Verb) likely to be "is"
  - Compute P(is|VBZ) by counting in a labeled corpus: $P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$

Back to Part-of-Speech Tagging



3 Problems

- Given this framework there are 3 problems that we can pose to an HMM:
  - Given an observation sequence and a model, what is the probability of that sequence?
  - Given an observation sequence and a model, what is the most likely state sequence?
  - Given an observation sequence, what are the best model parameters?

3 Problems

- Given this framework there are 3 problems that we can pose to an HMM:
  - Given an observation sequence and a model, what is the probability of that sequence?
  - Given an observation sequence and a model, what is the most likely state sequence?
  - Given an observation sequence, what are the best model parameters?

We did a small part of this in the in-class exercise! (Why?)

Question

- If there are 30 or so tags in the Penn set…
- …and the average sentence is around 20 words...
- …how many tag sequences do we have to enumerate to argmax over?

Question

- If there are 30 or so tags in the Penn set…
- …and the average sentence is around 20 words...
- …how many tag sequences do we have to enumerate to argmax over?

$\boxed{30^{20}}$

Problem 1

- The probability of a sequence given a model...

Computing Likelihood: Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.

- Used in sequence classification tasks
  - Word spotting in ASR, language identification, speaker identification, author identification, etc.

Problem 1

- The probability of a sequence given a model...

Computing Likelihood: Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.

- Used in sequence classification tasks
  - Word spotting in ASR, language identification, speaker identification, author identification, etc.
  - Train one model per class
  - Given an observation, pass it to each model and compute P(seq|model)
  - Argmax over models gives you the class

Problem 2

- Most probable state sequence given a model and an observation sequence

Decoding: Given as input an HMM $\lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, ..., o_T$, find the most probable sequence of states $Q = q_1 q_2 q_3 \ldots q_T$.

Problem 2

- Most probable state sequence given a model and an observation sequence

Decoding: Given as input an HMM $\lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, ..., o_T$, find the most probable sequence of states $Q = q_1 q_2 q_3 \ldots q_T$.

  - Typically used in sequence labeling problems, where the labels correspond to hidden states
    - As we'll see almost any problem can be cast as a sequence labeling problem

Problem 3

- Infer the best model parameters, given a model and an observation sequence...
    - That is, fill in the A and B tables with the right numbers...
        - The numbers that make the observation sequence most likely.

Problem 3

- Infer the best model parameters, given a model and an observation sequence...
  - That is, fill in the A and B tables with the right numbers...
    - The numbers that make the observation sequence most likely.
  - Useful for getting an HMM without supervision/annotators!

Solutions

- Problem 1: Forward
- Problem 2: Viterbi
- Problem 3: Forward-Backward
  - An instance of Expectation Maximization (EM)

Wrapping up

- Discussed today:
  - Part-of-speech tagging
  - Markov chains
  - Hidden Markov models

- On Wednesday: The Viterbi algorithm (guest lecture by Abhidip)