Natural Language Processing

CSCI 5832— Lecture 5

Jim Martin

NLP@CU

Office Hours

- Tuesday: 2:30 – 4:00
- Thursday: 1:00 – 2:00
  - By Zoom

Today

Parts of Speech

- Part-of-speech tagging
- Hidden Markov Models (HMMs)

Word Classes: Parts of Speech

- Traditional parts of speech
  - Noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction, etc.
    - There are various names for this notion
      - Part of speech, lexical category, word class, morphological class, lexical tag...

Word Classes: Parts of Speech

- Three sources of evidence
  1. Semantics
  2. Morphological evidence
  3. Distributional evidence

What’s a noun?

Word Classes: Parts of Speech

- Three sources of evidence
  1. ~~Semantics~~
  2. Morphological evidence
  3. Distributional evidence

Word Classes: Parts of Speech

- Three sources of evidence
  1. ~~Semantics~~
  2. Morphological evidence
     1. walk, walking, walked, walks
        - Probably a verb!
  3. Distributional evidence

Word Classes: Parts of Speech

- Three sources of evidence
  1. ~~Semantics~~
  2. Morphological evidence
     1. walk, walking, walked, walks
        - probably a verb
  3. Distributional evidence
     1. The crash, A crash, Two crashes, The big crash…
        - probably a noun since nouns follow determiners and adjectives

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.
  - Knowing the part of speech of the words in an input is a valuable signal for further processing
- Fast and accurate taggers are widely available for many languages

POS Tagging

- The process of assigning a part of speech or lexical class marker to each word in a text.

- The is our first example of a *sequence labeling task*
  - *Assigning a category label to each element of a sequence.*

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

POS Tagging

- Words can 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
  - Usually for a sentence

POS Tagging

- Note this is distinct from the task of identifying which *sense* of a word is being used given a particular part of speech. That's called word sense disambiguation.
  - "backed" is a verb in both of these examples
    - "… *backed* the car into a pole"
    - "… *backed* the wrong candidate"

How Hard is POS Tagging? 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*)

How Hard is POS Tagging? Measuring Ambiguity

- By a wide margin, most words in the vocabulary have only a single tag associated with them. So what's the problem?

Details:	2 tags	4,967	6,731

- The words that do have more than one tag are also the most frequently occurring ones.
  - In fact, there's a good correlation between number of tags and word frequency.

Methods for POS Tagging

1. Rule-based tagging
2. Probabilistic sequence models
    - HMM (Hidden Markov Model) tagging
    - Neural sequence models

POS Tagging as Sequence Labeling

- Given a sentence (an “observation” or “sequence of observations”)
  - *Secretariat is expected to race tomorrow*
- What is the best sequence of tags that corresponds to this sequence of observations?
- Probabilistic view
  - Consider all possible sequences of tags given the words and assign a probability to each tag sequence.
  - Out of this space of possible sequences, choose the tag sequence that is most probable given the observation sequence of n words $w_1 \dots w_n$.

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.

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

- Hat ^ means "our estimate of the best one"
- Argmax_x f(x) means "the x such that f(x) is maximized"

Towards HMMs

- This equation gives us our starting point

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

How do we make this operational? How to compute this value? Two steps

  - Use Bayes rule to transform this equation into a new set of equations
  - Use independence assumptions to make computing these tractable

Using Bayes Rule

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

Know this.

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

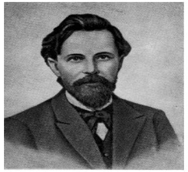
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \frac{P(w_1^n|t_1^n)P(t_1^n)}{P(w_1^n)}$$

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

Likelihood and Prior

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

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$



$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-1})$$

POS Tagging: Two Kinds of Probabilities

- Tag transition probabilities $p(t_i|t_{i-1})$
  - What's the probability that a noun will follow a determiner? Assume we have tagged data.
    - That/DT flight/NN
    - The/DT yellow/JJ hat/NN
  - Compute P(NN|DT) by counting in a labeled corpus:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

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

Two Kinds of Probabilities

- Word likelihood probabilities $p(w_i|t_i)$.
  - What's the probability that we'll see a particular word given a particular word class?
    - For example, that the tag VBZ (3sg Pres Verb) will be the word "is"
  - Compute P(is|VBZ) by counting in

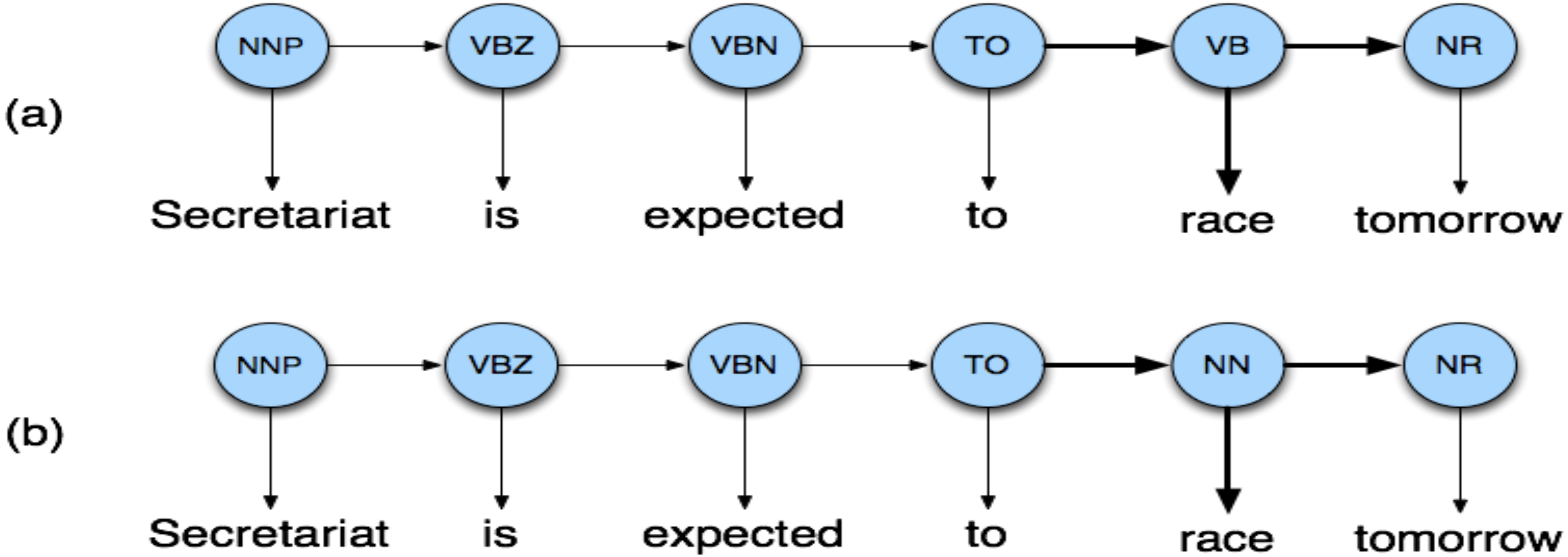
$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

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

Example: “race”

- Secretariat/NNP is/VBZ expected/VBN to/TO race/VB tomorrow/NR
- People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN

Disambiguating "race"

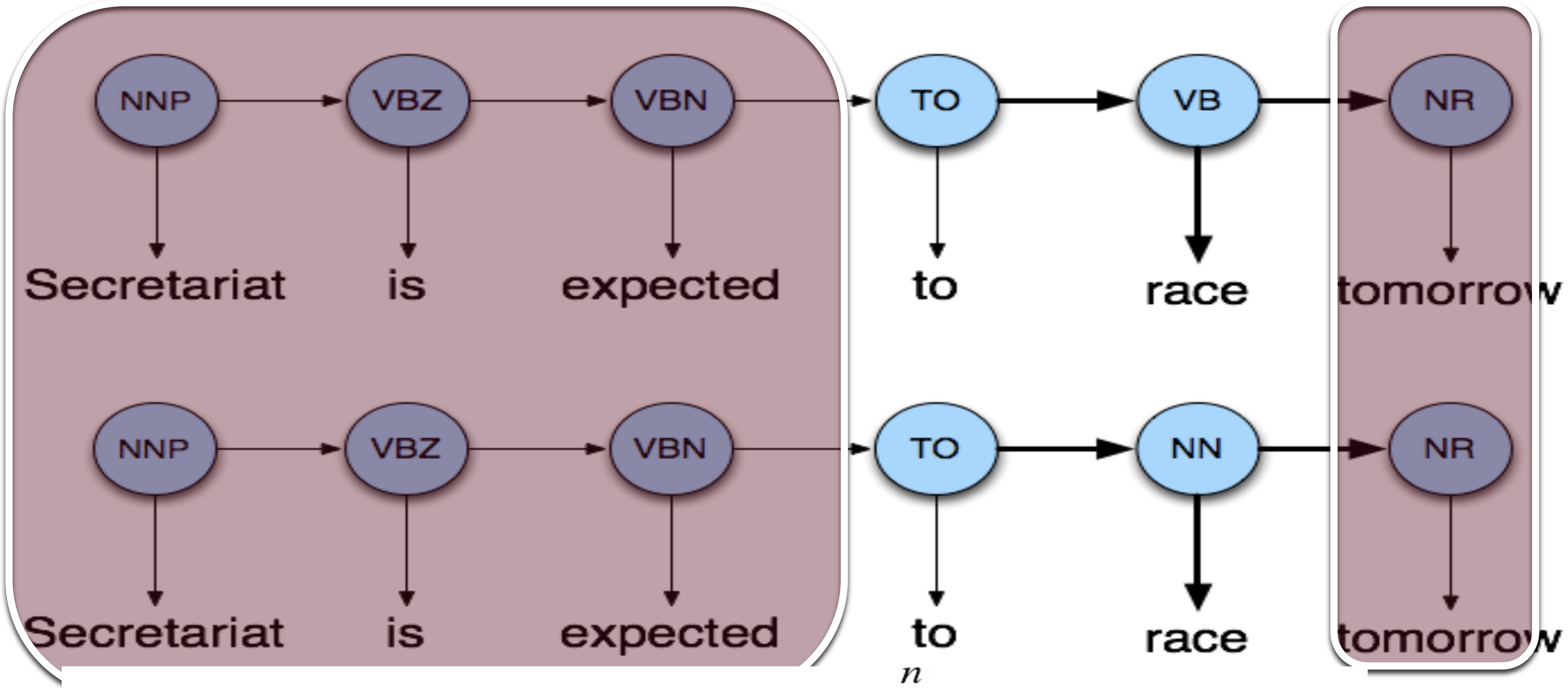


$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

Disambiguating "race"

(a)

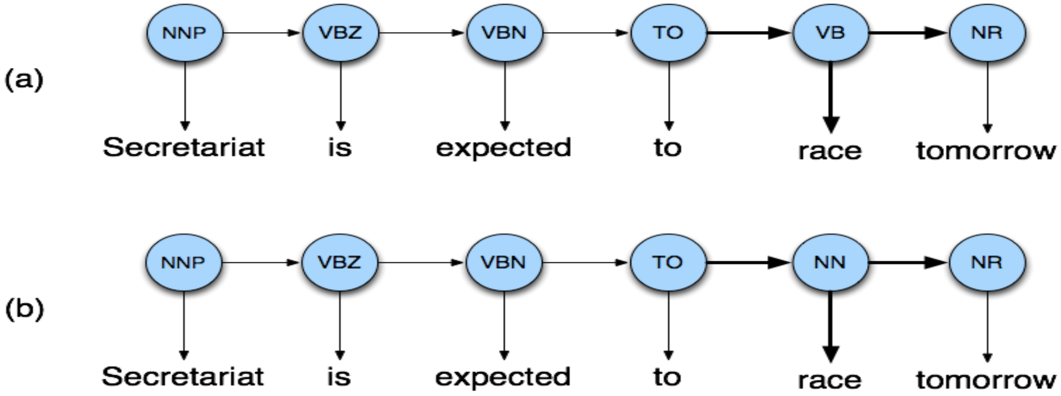
(b)



$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

Example

- P(NN|TO) = .00047
- P(VB|TO) = .83
- P(race|NN) = .00057
- P(race|VB) = .00012
- P(NR|VB) = .0027
- P(NR|NN) = .0012



- P(VB|TO)P(NR|VB)P(race|VB) = .00000027
- P(NN|TO)P(NR|NN)P(race|NN)=.00000000032
- So we (correctly) choose the verb tag for "race"

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?

$$30^{20}$$

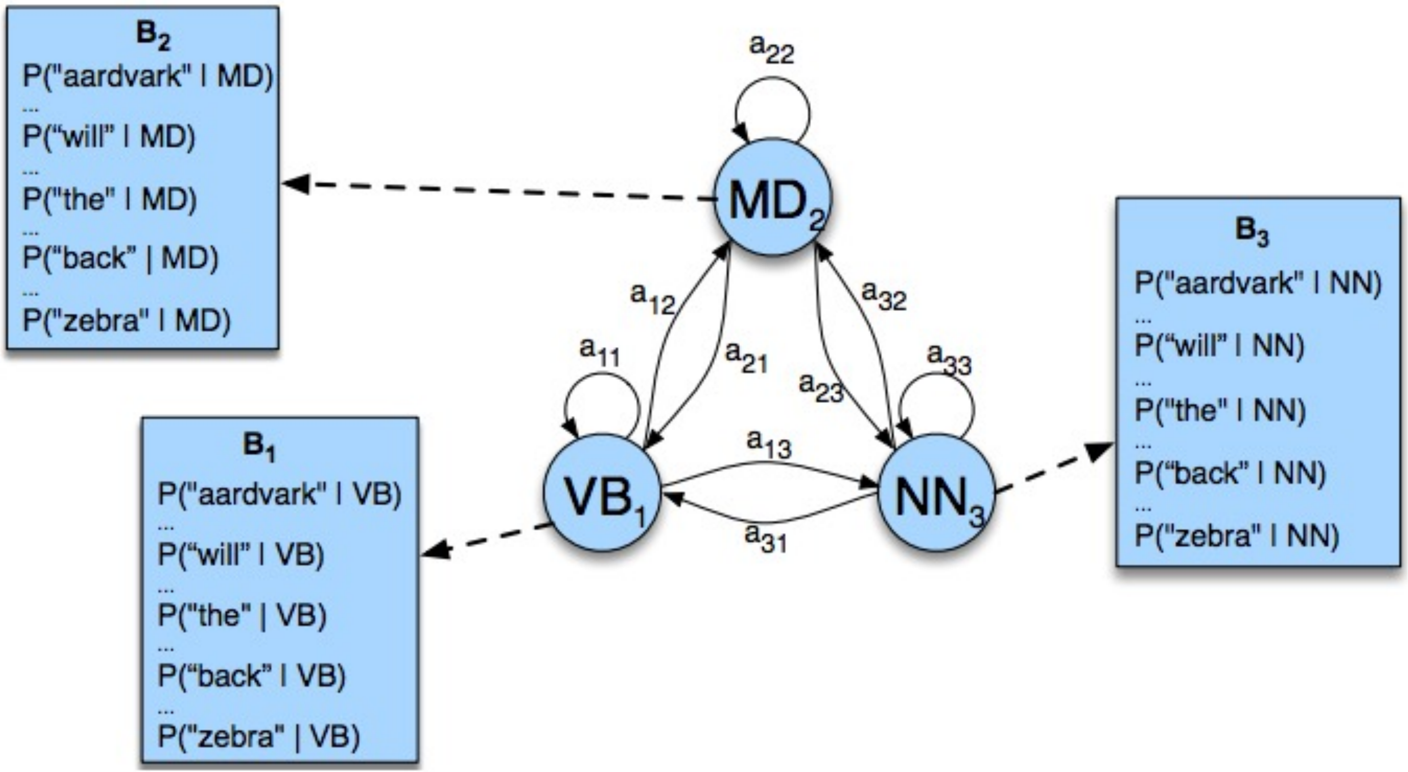
Hidden Markov Models

- What we’ve just described is called a Hidden Markov Model (HMM)
- This is an example of a *generative* model.
  - There is a hidden underlying generator of observable events
  - The hidden generator can be modeled as a network of states and transitions
  - We want to infer the underlying state sequence given the observed event sequence

HMM Tagging

- The hidden process is the unseen process of part of speech sequences
  - Modeled as states and state transitions
- The observations are the words that are emitted for each POS
  - Modeled as emissions from states

POS Tagging as an HMM



Hidden Markov Models

- States $Q = q_1, q_2 \ldots q_{N;}$
- Observations $O = o_1, o_2 \ldots o_{N;}$
  - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \ldots v_V\}$
- Transition probabilities
  - Transition probability matrix $A = \{a_{ij}\}$

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \le i, j \le N$$

- Observation likelihoods
  - Output probability matrix $B = \{b_i(k)\}$

$$b_i(k) = P(X_t = o_k \mid q_t = i)$$

- Special initial probability vector π

$$\pi_i = P(q_1 = i) \quad 1 \le i \le N$$

3 Problems

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

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
  - Used in model development... How do I know if some change I made to the model is making things better?

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
  - Useful for getting an HMM without having to hire annotators...

Solutions

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