Part-of-Speech Tagging

Chapter 8 (8.1-8.4.6)

Outline

- Parts of speech (POS)
- Tagsets
- POS Tagging
  - Rule-based tagging
  - Probabilistic (HMM) tagging

Garden Path Sentences

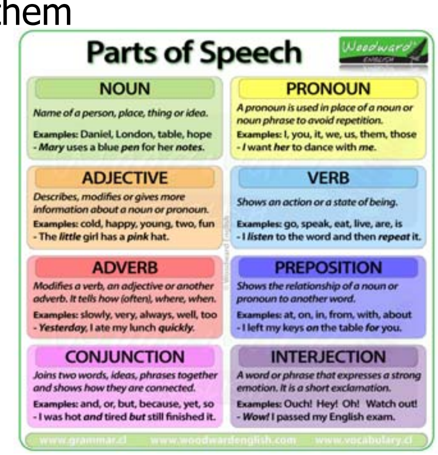
- The old dog the footsteps of the young

Parts of Speech

- Traditional parts of speech
  - Noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction, etc
  - Called: parts-of-speech, lexical categories, word classes, morphological classes, lexical tags...
  - Lots of debate within linguistics about the number, nature, and universality of these
    - We'll completely ignore this debate.

Parts of Speech

- Traditional parts of speech
  - ~ 8 of them



POS examples

- N noun *chair, bandwidth, pacing*
- V verb *study, debate, munch*
- ADJ adjective *purple, tall, ridiculous*
- ADV adverb *unfortunately, slowly*
- P preposition *of, by, to*
- PRO pronoun *I, me, mine*
- DET determiner *the, a, that, those*

POS Tagging

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

WORD	tag
the	
koala	
put	
the	
keys	
on	
the	
table	

POS Tagging

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

WORD	tag
the	**DET**
koala	
put	
the	
keys	
on	
the	
table	

POS Tagging

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

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

POS Tagging

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

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

POS Tagging

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

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

POS Tagging

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

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

POS Tagging

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

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

POS Tagging

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

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

POS Tagging

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

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

Why is POS Tagging Useful?

- First step of many practical tasks, e.g.
- Speech synthesis (aka text to speech)
  - How to pronounce "lead"?
  - OBject objECT
  - CONtent conTENT
- Parsing
  - Need to know if a word is an N or V before you can parse
- Information extraction
  - Finding names, relations, etc.
- Language modeling
  - Backoff

Why is POS Tagging Difficult?

- Words often have more than one POS: *back*
  - The *back* door = adjective
  - On my *back* =
  - Win the voters *back* =
  - Promised to *back* the bill =

Why is POS Tagging Difficult?

- Words often have more than one POS: *back*
  - The *back* door = adjective
  - On my *back* = noun
  - Win the voters *back* =
  - Promised to *back* the bill =

Why is POS Tagging Difficult?

- Words often have more than one POS: *back*
  - The *back* door = adjective
  - On my *back* = noun
  - Win the voters *back* = adverb
  - Promised to *back* the bill =

Why is POS Tagging Difficult?

- Words often have more than one POS: *back*
  - The *back* door = adjective
  - On my *back* = noun
  - Win the voters *back* = adverb

  - Promised to *back* the bill = verb
- The POS tagging problem is to determine the POS tag for a particular instance of a word.

POS Tagging

- Input: Plays well with others
- Ambiguity: NNS/VBZ UH/JJ/NN/RB IN NNS
- Output: Plays/VBZ well/RB with/IN others/NNS

Penn Treebank POS tags

POS tagging performance

- How many tags are correct? (Tag accuracy)
  - About 97% currently
  - But baseline is already 90%
    - Baseline is performance of stupidest possible method
      - Tag every word with its most frequent tag
      - Tag unknown words as nouns
  - Partly easy because
    - Many words are unambiguous
    - You get points for them (*the, a,* etc.) and for punctuation marks!

Deciding on the correct part of speech can be difficult even for people

- Mrs/NNP Shaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG
- All/DT we/PRP gotta/VBN do/VB is/VBZ go/VB around/IN the/DT corner/NN
- Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

How difficult is POS tagging?

- About 11% of the word types in the Brown corpus are ambiguous with regard to part of speech
- But they tend to be very common words. E.g., *that*
  - I know *that* he is honest = IN
  - Yes, *that* play was nice = DT
  - You can't go *that* far = RB
- 40% of the word tokens are ambiguous

Review

- Backoff/Interpolation
- Parts of Speech
  - What?
- Part of Speech Tagging
  - What?
  - Why?
  - Easy or hard?
  - Evaluation

Open vs. Closed Classes

- Closed class: ***why?***
  - Determiners: a, an, the
  - Prepositions: of, in, by, …
  - Auxiliaries: may, can, will had, been, …
  - Pronouns: I, you, she, mine, his, them, …
  - Usually function words (short common words which play a role in grammar)
- Open class: ***why?***
  - English has 4: Nouns, Verbs, Adjectives, Adverbs
  - Many languages have these 4, but not all!

Open vs. Closed Classes

- Closed class: a small fixed membership
  - Determiners: a, an, the
  - Prepositions: of, in, by, …
  - Auxiliaries: may, can, will had, been, …
  - Pronouns: I, you, she, mine, his, them, …
  - Usually function words (short common words which play a role in grammar)
- Open class: new ones can be created all the time
  - English has 4: Nouns, Verbs, Adjectives, Adverbs
  - Many languages have these 4, but not all!

Open Class Words

- Nouns
  - Proper nouns (Pittsburgh, Pat Gallagher)
    - English capitalizes these.
  - Common nouns (the rest).
  - Count nouns and mass nouns
    - Count: have plurals, get counted: goat/goats, one goat, two goats
    - Mass: don't get counted (snow, salt, communism) (*two snows)
- Adverbs: tend to modify things
  - Unfortunately, John walked home extremely slowly yesterday
  - Directional/locative adverbs (here,home, downhill)
  - Degree adverbs (extremely, very, somewhat)
  - Manner adverbs (slowly, slinkily, delicately)
- Verbs
  - In English, have morphological affixes (eat/eats/eaten)

Closed Class Words

Examples:

- prepositions: *on, under, over, …*
- particles: *up, down, on, off, …*
- determiners: *a, an, the, …*
- pronouns: *she, who, I, ..*
- conjunctions: *and, but, or, …*
- auxiliary verbs: *can, may should, …*
- numerals: *one, two, three, third, …*

Prepositions from CELEX

of	540,085	through	14,964	worth	1,563	pace	12
in	331,235	after	13,670	toward	1,390	nigh	9
for	142,421	between	13,275	plus	750	re	4
to	125,691	under	9,525	till	686	mid	3
with	124,965	per	6,515	amongst	525	o'er	2
on	109,129	among	5,090	via	351	but	0
at	100,169	within	5,030	amid	222	ere	0
by	77,794	towards	4,700	underneath	164	less	0
from	74,843	above	3,056	versus	113	midst	0
about	38,428	near	2,026	amidst	67	o'	0
than	20,210	off	1,695	sans	20	thru	0
over	18,071	past	1,575	circa	14	vice	0

POS Tagging
Choosing a Tagset

- There are so many parts of speech, potential distinctions we can draw
- To do POS tagging, we need to choose a standard set of tags to work with
- Could pick very coarse tagsets
  - N, V, Adj, Adv.
- More commonly used set is finer grained, the "Penn TreeBank tagset", 45 tags
- Even more fine-grained tagsets exist

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*			

Using the Penn Tagset

- The/? grand/? jury/? commmented/? on/? a/? number/? of/? other/? topics/? ./?

Using the Penn Tagset

- The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

Recall POS Tagging Difficulty

- Words often have more than one POS: *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 POS tag for a particular instance of a word.

These examples from Dekang Lin

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*)

Tagging Whole Sentences with POS is Hard too

- Ambiguous POS contexts
  - E.g., Time flies like an arrow.
- Possible POS assignments
  - Time/[V,N] flies/[V,N] like/[V,Prep] an/Det arrow/N
  - Time/N flies/V like/Prep an/Det arrow/N
  - Time/V flies/N like/Prep an/Det arrow/N
  - Time/N flies/N like/V an/Det arrow/N
  - …..

How Do We Disambiguate POS?

- Many words have only one POS tag (e.g. is, Mary, smallest)
- Others have a single ***most likely*** tag (e.g. Dog is less used as a V)
- Tags also tend to *co-occur* regularly with other tags (e.g. Det, N)
- In addition to conditional probabilities of words $P(w_1|w_{n-1})$, we can look at POS likelihoods $P(t_1|t_{n-1})$ to disambiguate sentences and to assess sentence likelihoods

More and Better Features → Feature-based tagger

- Can do surprisingly well just looking at a word by itself:
  - Word — the: the → DT
  - Lowercased word — Importantly: importantly → RB
  - Prefixes — unfathomable: un- → JJ
  - Suffixes — Importantly: -ly → RB
  - Capitalization — Meridian: CAP → NNP
  - Word shapes — 35-year: d-x → JJ

Overview: POS Tagging Accuracies

Most errors on unknown words

- Rough accuracies:
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%
  - Maxent P(t|w): 93.7% / 82.6%
  - Upper bound: ~98% (human)

Rule-Based Tagging

- Start with a dictionary
- Assign all possible tags to words from the dictionary
- Write rules by hand to selectively remove tags
- Leaving the correct tag for each word.

Start With a Dictionary

- she:
- promised:
- to
- back:
- the:
- bill:

Start With a Dictionary

- she: PRP
- promised: VBN,VBD
- to TO
- back: VB, JJ, RB, NN
- the: DT
- bill: NN, VB

Assign Every Possible Tag

			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

Write Rules to Eliminate Tags

Eliminate VBN if VBD is an option when VBN|VBD follows “<start> PRP”

			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

POS tag sequences

- Some tag sequences are more likely occur than others
- POS Ngram view https://books.google.com/ngrams/graph?content=_ADJ_+_NOUN_%2C_ADV_+_NOUN_%2C+_ADV_+_VERB_

Existing methods often model POS tagging as a sequence tagging problem

POS Tagging as Sequence Classification

- We are 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
  - Out of this universe of sequences, choose the tag sequence which is most probable given the observation sequence of n words $w_1 \ldots w_n$.

How do you predict the tags?

- Two types of information are useful
  - Relations between words and tags
  - Relations between tags and tags
    - DT NN, DT JJ NN…

Getting to HMMs (Hidden Markov Models)

- 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"

Getting to HMMs

- This equation is guaranteed to give us the best tag sequence

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

- But how to make it operational? How to compute this value?
- Intuition of Bayesian classification:
  - Use Bayes rule to transform this equation into a set of other probabilities that are easier to compute

Using Bayes Rule

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

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

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

Statistical POS tagging

- What is the most likely sequence of tags for the given sequence of words w

$$\begin{aligned}\underset{\mathbf{t}}{\operatorname{argmax}} P(\mathbf{t}|\mathbf{w}) &= \underset{\mathbf{t}}{\operatorname{argmax}} \frac{P(\mathbf{t},\mathbf{w})}{P(\mathbf{w})} \\ &= \underset{\mathbf{t}}{\operatorname{argmax}} P(\mathbf{t},\mathbf{w}) \\ &= \underset{\mathbf{t}}{\operatorname{argmax}} P(\mathbf{t})P(\mathbf{w}|\mathbf{t})\end{aligned}$$

P(DT JJ NN | a smart dog) =

Statistical POS tagging

- What is the most likely sequence of tags for the given sequence of words w

$$\begin{aligned}\operatorname*{argmax}_{\mathbf{t}} P(\mathbf{t}|\mathbf{w}) &= \operatorname*{argmax}_{\mathbf{t}} \frac{P(\mathbf{t},\mathbf{w})}{P(\mathbf{w})} \\ &= \operatorname*{argmax}_{\mathbf{t}} P(\mathbf{t},\mathbf{w}) \\ &= \operatorname*{argmax}_{\mathbf{t}} P(\mathbf{t})P(\mathbf{w}|\mathbf{t})\end{aligned}$$

P(DT JJ NN | a smart dog) = P(DD JJ NN a smart dog) / P (a smart dog)
∝ P(DD JJ NN a smart dog)
= P(DD JJ NN) P(a smart dog | DD JJ NN)

Likelihood and Prior

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

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

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

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

Two Kinds of Probabilities

- 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
    - But P(DT|JJ) to be:
  - 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 (emission) probabilities $p(w_i|t_i)$
  - VBZ (3sg Pres verb) likely to be “is”
  - Compute P(is|VBZ) by counting in a labeled corpus:

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

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

Put them together

- Two independent assumptions
  - Approximate P(**t**) by a bi(or N)-gram model
  - Assume each word depends only on its POS tag

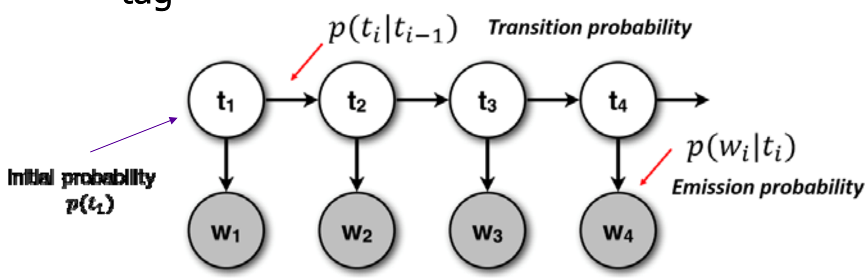


Table representation

Transition Matrix A

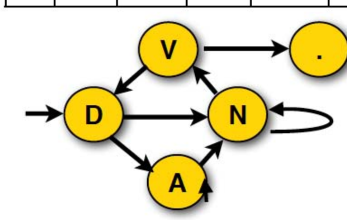
	D	N	V	A	.
D		0.8		0.2	
N		0.7	0.3		
V	0.6				0.4
A		0.8		0.2	
.					

Emission Matrix B

	the	*man*	*ball*	*throws*	*sees*	*red*	*blue*	.
D	1.0							
N		0.7	0.3					
V				0.6	0.4			
A						0.8	0.2	
.								1

Initial state vector π

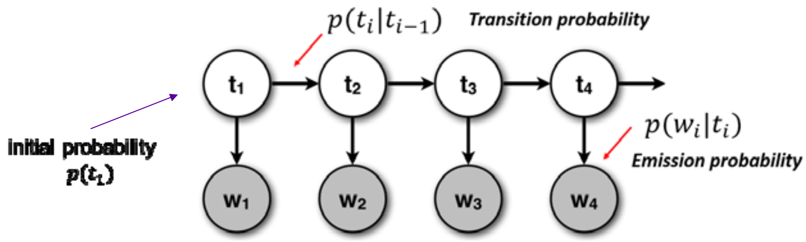
	D	N	V	A	.
π	1.0				



Let $\lambda = \{A, B, \pi\}$ represents all paramete

Prediction in generative model

- Inference: What is the most likely sequence of tags for the given sequence of words **w**

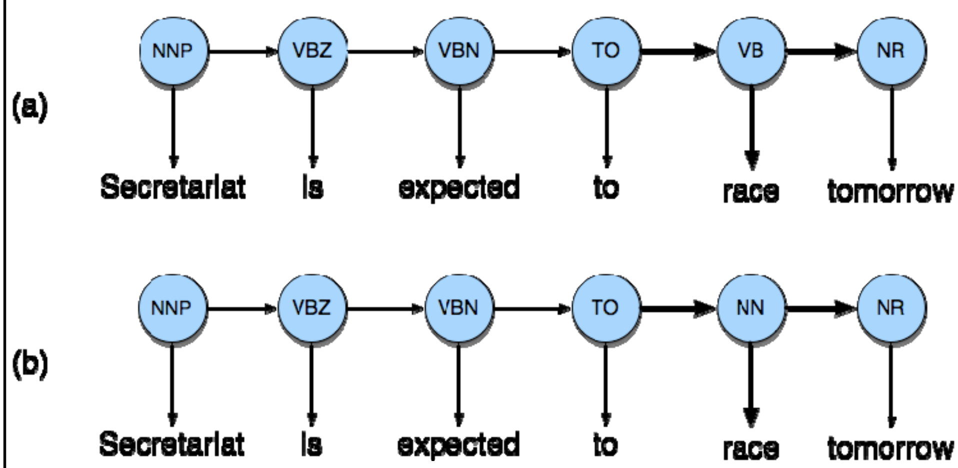


- What are the latent states that most likely generate the sequence of word **w**

Example: The Verb "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
- How do we pick the right tag?

Disambiguating "race"



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 reading

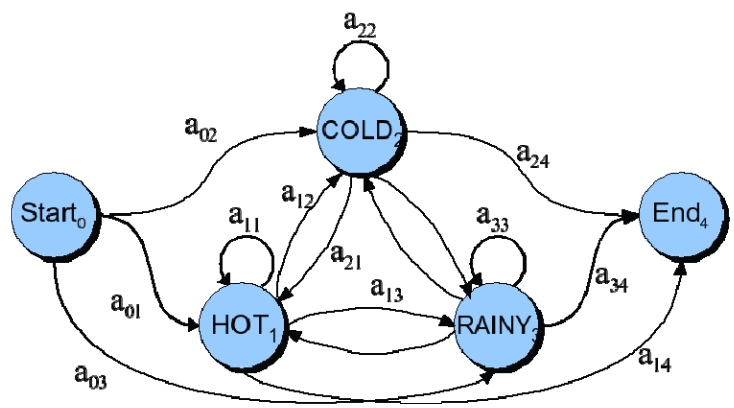
Hidden Markov Models

- What we've described with these two kinds of probabilities is a Hidden Markov Model (HMM)

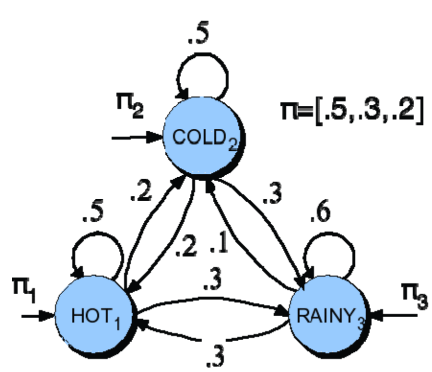
Definitions

- A weighted finite-state automaton adds probabilities to the arcs
  - The sum of the probabilities leaving any arc must sum to one
- A Markov chain is a special case of a WFSA in which the input sequence uniquely determines which states the automaton will go through
- Markov chains can't represent inherently ambiguous problems
  - Useful for assigning probabilities to unambiguous sequences

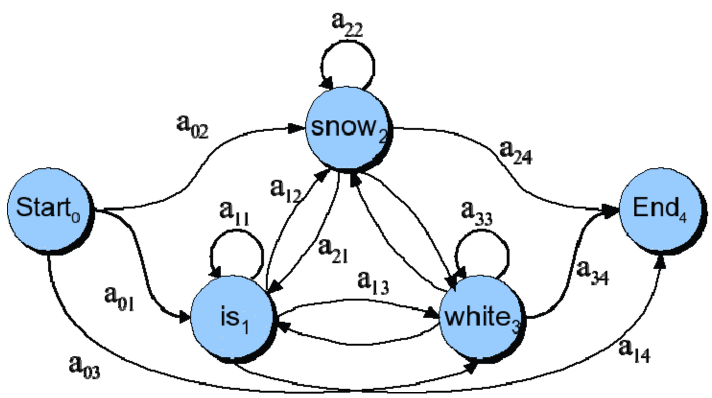
Markov Chain for Weather



Weather continued



Markov Chain for Words



Markov Chain: "First-order observable Markov Model"

- A set of states
  - $Q = q_1, q_2 \ldots q_{N;}$ the state at time t is q_t
- Transition probabilities:
  - a set of probabilities $A = a_{01}a_{02}\ldots a_{n1}\ldots a_{nn}$.
  - Each a_{ij} represents the probability of transitioning from state i to state j
  - The set of these is the transition probability matrix A
- Current state only depends on previous state

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

Markov Chain for Weather

- What is the probability of 4 consecutive rainy days?
- Sequence is rainy-rainy-rainy-rainy
- I.e., state sequence is 3-3-3-3
- P(3,3,3,3) =

Markov Chain for Weather

- What is the probability of 4 consecutive rainy days?
- Sequence is rainy-rainy-rainy-rainy
- I.e., state sequence is 3-3-3-3
- P(3,3,3,3) =
  - $\pi_3 a_{33} a_{33} a_{33} = 0.2 \times (0.6)^3 = 0.0432$

Review

- Tagsets
  - What?
  - Example(s)
- Baseline(s) for tagging evaluation
- Two types of probabilities for POS tagging
  - assumptions
- Markov Chain vs Hidden Markov Model

HMM for Ice Cream

- You are a climatologist in the year 2799
- Studying global warming
- You can't find any records of the weather in Pittsburgh for summer of 2018
- But you find a diary
- Which lists how many ice-creams someone ate every date that summer
- Our job: figure out how hot it was

Hidden Markov Model

- For Markov chains, the output symbols are the same as the states.
  - See **hot** weather: we're in state **hot**
- But in part-of-speech tagging (and other things)
  - The output symbols are **words**
  - But the hidden states are **part-of-speech tags**
- So we need an extension!
- A Hidden Markov Model is an extension of a Markov chain in which the input symbols are not the same as the states.
- This means we don't know which state we are in.

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 $B = \{b_i(k)\}$

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

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

- Special initial probability vector π

Task

- Given
  - Ice Cream Observation Sequence: 1,2,3,2,2,2,3…
- Produce:
  - Weather Sequence: H,C,H,H,H,C…

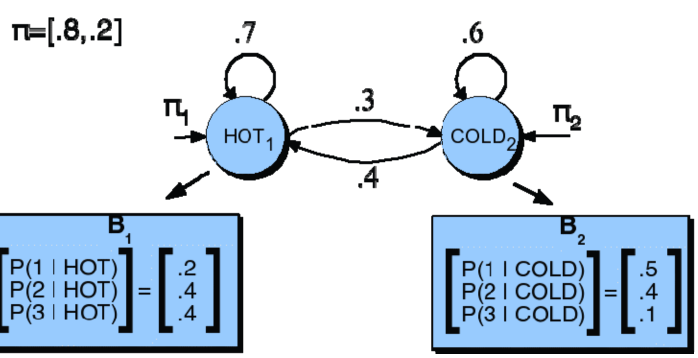
Weather/Ice Cream HMM

- Hidden States:
- Transition probabilities:
- Observations:

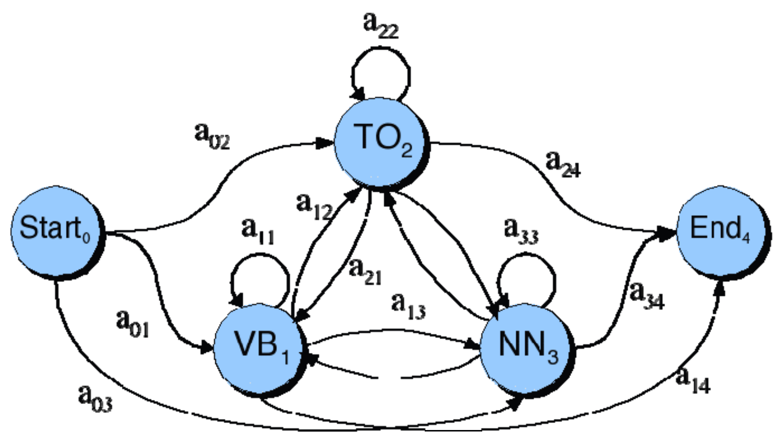
Weather/Ice Cream HMM

- Hidden States: {Hot,Cold}
- Transition probabilities (A Matrix) between H and C
- Observations: {1,2,3} # of ice creams eaten per day

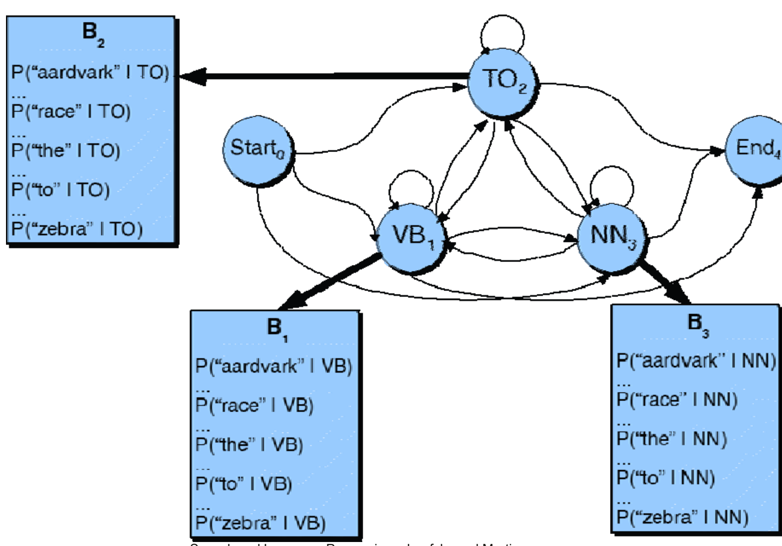
HMM for Ice Cream



Back to POS Tagging: Transition Probabilities



Observation Likelihoods



What can HMMs Do?

- ***Likelihood***: Given an HMM λ and an observation sequence O, determine the likelihood P(O, λ): *language modeling*
- ***Decoding***: Given an observation sequence O and an HMM λ, discover the *best* hidden state sequence Q: Given seq of ice creams, what was the most likely weather on those days? *(tagging)*
- ***Learning***: Given an observation sequence O and the set of states in the HMM, learn the HMM *parameters*

Decoding

- Ok, now we have a complete model that can give us what we need. Recall that we need to get

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

- We could just enumerate all paths given the input and use the model to assign probabilities to each.
  - Not a good idea.
  - In practice: Viterbi Algorithm (dynamic programming)

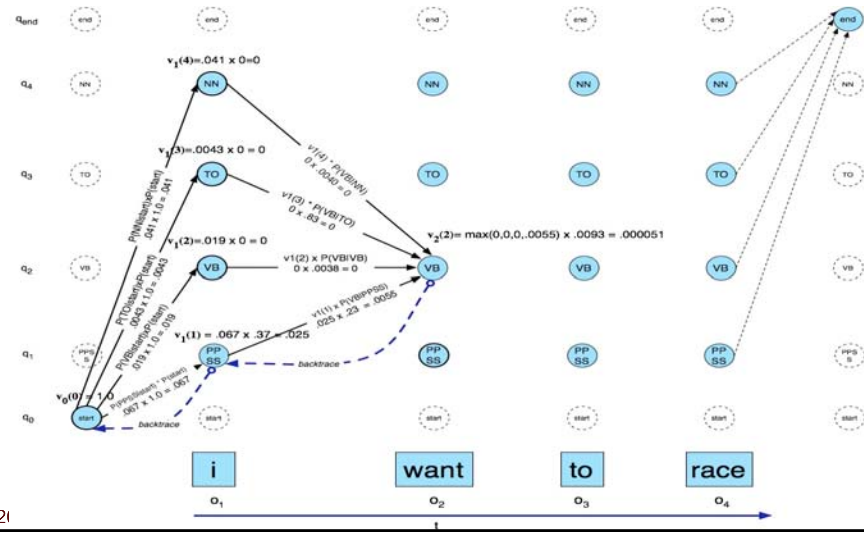
Viterbi Algorithm

- Intuition: since state transition out of a state only depend on the current state (and not previous states), we can record for each state the optimal path
- We record
  - Cheapest cost to state at step
  - Backtrace for that state to best predecessor

Viterbi Summary

- Create an array
  - With columns corresponding to inputs
  - Rows corresponding to possible states
- Sweep through the array in one pass filling the columns left to right using our transition probs and observations probs
- Dynamic programming key is that we need only store the MAX prob path to each cell (not all paths).

Viterbi Example



Another Viterbi Example

- Analyzing "Fish sleep"
  - Done in class

Evaluation

- So once you have your POS tagger running how do you evaluate it?
  - Overall error rate with respect to a gold-standard test set.
  - Error rates on particular tags
  - Error rates on particular words
  - Tag confusions...
- Need a baseline – just the most frequent tag is 90% accurate!

Error Analysis

- Look at a confusion matrix

	IN	JJ	NN	NNP	RB	VBD	VBN
IN	—	.2			.7		
JJ	.2	—	**3.3**	2.1	1.7	.2	**2.7**
NN		**8.7**	—				.2
NNP	.2	**3.3**	**4.1**	—	.2		
RB	**2.2**	2.0	.5		—		
VBD		.3	.5			—	**4.4**
VBN		**2.8**				**2.6**	—

- See what errors are causing problems
  - Noun (NN) vs ProperNoun (NNP) vs Adj (JJ)
  - Preterite (VBD) vs Participle (VBN) vs Adjective (JJ)

Evaluation

- The result is compared with a manually coded "Gold Standard"
  - Typically accuracy reaches 96-97%
  - This may be compared with result for a baseline tagger (one that uses no context).
- Important: 100% is impossible even for human annotators.

More Complex Issues

- Tag indeterminacy: when 'truth' isn't clear
  Caribbean cooking, child seat
- Tagging multipart words
  wouldn't --> would/MD n't/RB
- How to handle unknown words
  - Assume all tags equally likely
  - Assume same tag distribution as all other singletons in corpus
  - Use morphology, word length,….

Other Tagging Tasks

- Noun Phrase (NP) Chunking
- [the student] said [the exam] is hard
- Three tabs
  - B = beginning of NP
  - I = continuing in NP
  - O = other word
- Tagging result
  - The/B student/I said/O the/B exam/I is/0 hard/0

Summary

- Parts of speech
- Tagsets
- Part of speech tagging
- Rule-Based, HMM Tagging
  - Other methods later in course