Lecture 6: POS Tags and HMMs

USC VSoE CSCI 544: Applied Natural Language Processing

Jonathan May -- 梅約納

September 8, 2017

based on slides of Nathan Schneider / Sharon Goldwater / Philipp Koehn

Reminders

- HW1 is due today, by 11:59pm (modulo late days)
- HW3 will come out this coming Wednesday (9/13)
- HW2 is due next Friday (9/15)
- Jon will be out of town next week (no office hours): guest lecturer Marjan Ghazvininejad will discuss syntactic parsing (lec. 8 & 9)
- No class 9/22 (2 weeks from today)
- Start thinking about the midterm (10/6; 4 weeks from today)
  - Don't let the reading slip: make sure you have the latest schedule
  - https://www.isi.edu/~jonmay/cs544_fa17_web/ (or go to jonmay.net)

What is Part of Speech (POS) Tagging?

- Given a string

  This is a simple sentence

- Identify parts of speech (syntactic categories)

  This/DET is/VB a/DET simple/ADJ sentence/NOUN

Why do we care about POS tagging?

- First step toward full syntactic analysis (which is a first step toward full semantic analysis)
  - simpler and faster than full syntactic parsing
  - often good features for other tasks (e.g. sentiment classification, word sense disambiguation)
- Good pedagogical tool for me: illustrates **Hidden Markov Models** (HMMs) which are used for other sequence labeling tasks

Other sequence labeling tasks

- **Named Entity Recognition** (NER): label words as beginning to **persons (PER)**, **organizations (ORG), locations (LOC)**, or none of the above
  - Barack/PER Obama/PER spoke/N from/N the/N White/LOC House/LOC today/N ./N
- **Information field segmentation**: Given specific text type (e.g. classified ad), find which words belong to which "fields" for db creation (price/size/location, author/title/year)
  - 3BR/SIZE apt/TYPE in/N West/LOC Adams/LOC ,/N near/LOC USC/LOC ./N Bright/FEAT ,/N well/FEAT maintained/FEAT ...

Sequence Labeling: Key Features

- In all of these, deciding the correct label depends on
  - The word to be labeled
    - NER: **Smith** is probably a person
    - POS: **chair** is probably a noun
  - The labels of surrounding words
    - NER: if following word is an organization (e.g. **Corp.**), then this word is more likely to be an organization
    - POS: if preceding word is a modal verb (e.g. **will**), then this word is more likely to be a verb
- HMM combines these sources probabilistically

Parts of Speech

- Open class words ("content words")
  - nouns, verbs, adjectives, adverbs
  - mostly content-bearing. refer to objects, actions, features in the world
  - *open class* = there is no limit to what they are or can describe so new ones are added all the time (**email**, **website**, **defenestrate**)
- Closed class words ("function words")
  - pronouns, determiners, prepositions, connectives
  - there are a limited number of these
  - mostly functional: to tie the concepts of a sentence together

How Many Parts of Speech Are There?

- Linguistic and practical considerations
- If we're being empirical (we are), corpus annotators decide
  - proper nouns vs common nouns?
  - singluar vs plural nouns?
  - past and present tense verbs?
  - auxiliary and main verbs?
- Commonly used tag sets for English usually have 40-100 tag types. The Penn Treebank has 45 tags.

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	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, sing.	*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*			

J&M Fig 5.6: Penn Treebank POS tags

POS Tags in other languages

- Morphologically rich languages often have compound morphosyntactic tags

  Noun+A3sg+P2sg+Nom

- Hundreds or thousands of possible combinations
- Predicting these requires more complex methods than what's in today's lecture
  - e.g. soft morphological segmentation (with FST?) + disambiguation

Why is POS tagging hard?

- Ambiguity
  - glass of **water/NOUN** vs **water/VERB** the plants
  - **lie/VERB** down vs tell a **lie/NOUN**
  - **wind/VERB** down vs a mighty **wind/NOUN** (homographs)

time	flies	like	an	arrow
NOUN	VERB	MODAL	DET	NOUN
VERB	NOUN			
ADJ	NOUN	VERB		

- Sparse data
  - Words we haven't seen before (at all, in this context)
  - Word-Tag pairs we haven't seen before (e.g. if we verb a noun)

Relevant knowledge for POS tagging

- Remember, we want a model that decides tags based on
  - The word itself
    - Some words may be only nouns e.g. **arrow**
    - Some words are ambiguous e.g. **like**, **flies**
    - Probabilities may help if one tag is more likely than another
  - Tags of surrounding words
    - Two determiners rarely follow each other
    - Two base form verbs rarely follow each other
    - Determiner is almost always followed by adjective or noun
- What might be a problem with putting this information in the models from last lecture?

A Probabilistic Model for Tagging

- We have a word sequence and we want a tag sequence for those words:
  - $P(T|W)$ ...guess how we're going to represent this again
- $P(T|W) = P(W, T)/P(W)$; $P(W, T) = P(T|W)P(W)$
- $P(W, T) = P(W|T)P(T)$
- $P(T|W) = P(W|T)P(T)/P(W)$
- $\text{argmax}_T\ P(T|W) = \text{argmax}_T P(W|T)P(T)/P(W)$
- $\text{argmax}_T\ P(T|W) = \text{argmax}_T P(W|T)P(T)$
- Note, btw, that $P(W|T)P(T) = P(W, T)$

Simplifying Assumptions

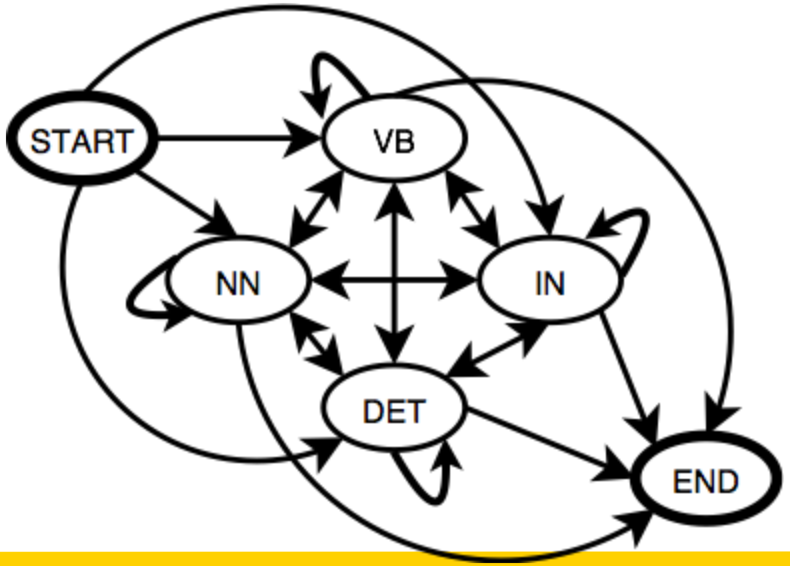
- We want P(W|T) and P(T) where W and T are sequences of length N
- Assumption 1: Each tag is conditioned only on the previous tag (a **bigram** model; this is why it's called "Markov")
  - $P(T) = \prod_{i=1}^{N} P(t_i|t_{i-1})$
- Assumption 2: Each word is conditioned only on its tag
  - $P(W|T) = \prod_{i=1}^{N} P(w_i|t_i)$
- Put it all together:
  - $P(W,T) = \prod_{i=1}^{N} P(t_i|t_{i-1})P(w_i|t_i) \times P(</s> \mid t_n)$ where t_0=<s>
- Notice the similarity to Naive Bayes, except the tag sequence is unknown

Quiz 1

- "walk" becomes "walked" in the past tense. What kind of morphology is this an example of?
  - inflectional
  - derivational
  - reduplicative

Connection to Probabilistic FSA

- One way to view this model: sentences are generated by walking through **states** in a graph. Each state represents a tag



- Probability of moving between states x and y (**transition probability**) is $P(t = y|t = x)$

Example Transition Probabilities

$t_{i-1} \backslash t_i$	NNP	MD	VB	JJ	NN	...
<s>	0.2767	0.0006	0.0031	0.0453	0.0449	...
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	...
MD	0.0008	0.0002	0.7968	0.0005	0.0008	...
VB	0.0322	0.0005	0.0050	0.0837	0.0615	...
JJ	0.0306	0.0004	0.0001	0.0733	0.4509	...
...	...	...	...	...	...	...

Table excerpted from J&M draft 3rd edition, Fig. 8.5

- Probabilities estimated from WSJ corpus showing, e.g.:
  - Proper Nouns (NNP) often begin sentences: P(NNP|<s>) = 0.28
  - Modal Verbs (MD) nearly always followed by bare verbs (VB)
  - Adjectives (JJ) are often followed by nouns

Example Transition Probabilities

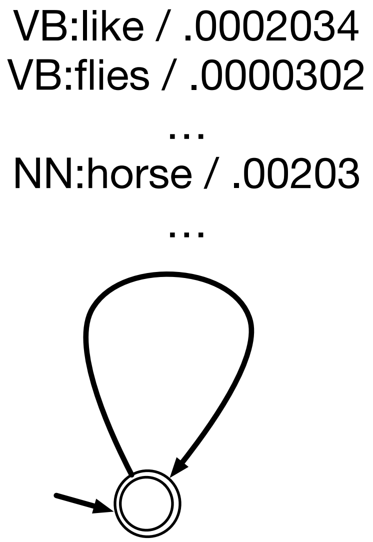
$t_{i-1} \backslash t_i$	NNP	MD	VB	JJ	NN	...
<s>	0.2767	0.0006	0.0031	0.0453	0.0449	...
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	...
MD	0.0008	0.0002	0.7968	0.0005	0.0008	...
VB	0.0322	0.0005	0.0050	0.0837	0.0615	...
JJ	0.0306	0.0004	0.0001	0.0733	0.4509	...
...	...	...	...	...	...	...

Table excerpted from J&M draft 3rd edition, Fig. 8.5

- Table is incomplete in 2 ways. How?
  - All categories should be represented
  - Sum of rows should = 1

Connection to Probabilistic FST

- Tag FSA outputs a tag; tag-to-word FST generates a word based on the tag



- weight on x:y = $P(w = y|t = x)$, i.e. the **emission probability**

Example Emission Probabilities

$t_i \backslash w_i$	Janet	will	back	the	...
NNP	0.000032	0	0	0.000048	...
MD	0	0.308431	0	0	...
VB	0	0.000028	0.000672	0	...
DT	0	0	0	0.506099	...
...	...	...	...	...	...

Table excerpted from J&M draft 3rd edition, Fig. 8.6

- MLE probabilities from tagged WSJ corpus showing, e.g.
  - 0.0032% of proper nouns are *Janet*: P(Janet|NNP) = 0.000032
  - About half of determiners are *the*
  - *the* can also be a proper noun (Annotation error?)
- Again, in full table, rows would sum to 1

What can we do with this model?

- This is a model of the joint probability P(T, W)
- So, if we have a word sequence and a tag sequence, we can get a probability for it

$$P(W,T) = \prod_{i=1}^{N} P(t_i|t_{i-1})P(w_i|t_i) \times P(</s> \mid t_n)$$

- E.g. P(This/DET is/VB a/DET simple/JJ sentence/NN)?

<s>	This	is	a	simple	sentence	</s>
<s>	DET	VB	DET	JJ	NN	</s>
	P(DET|<s>)	P(VB|DET)	P(DET|VB)	P(JJ|DET)	P(NN|JJ)	P(</s>|NN)
	p(This|DET)	P(is|VB)	P(a|DET)	P(simple|JJ)	P(sentence|NN)	

How to tag an unlabeled sequence?

- Let's say we're just given "This is a simple sentence"
- Recall, in this formulation, we derived P(T, W) in order to solve $\text{argmax}_T\, P(T)P(W|T)$
- So try "DT DT DT DT DT", "DT DT DT DT NN", ...
  - There are 45 tags. How many sequences will we try?
  - 45^5=184,528,125

Greedy Algorithm

tags ordered by frequency for each word

<s>	one	dog	bit	</s>
<s>	CD	NN	NN	</s>
	NN	VB	VBD	
	PRP			

- Simplest: just choose the most likely tag for each word, i.e. $\text{argmax}_i P(w_i|t_i)$
  - Since we don't consider tag context we get the wrong answer
- Simple: At time i, choose $\text{argmax}_i P(t_i|t_{i-1})P(w_i|t_i)$
  - Since t_{i-1} and w_i are determined, O(|T| x N) runtime – same as above
  - This uses tag context and gets a better result because P(VBD|NN) and P(</s>|VBD) are high

Greedy Algorithm

tags ordered by
frequency for each word

<s>	one	dog	bit	</s>
<s>	CD	NN	NN	</s>
	NN	VB	VBD	
	PRP			

- Greedy ($\text{argmax}_i P(t_i|t_{i-1})P(w_i|t_i)$) is still suboptimal
  - You commit to a tag before considering <u>subsequent</u> tags
  - It could be the case that ALL possible next tags have low transition probabilities
  - E.g. a tag that is unlikely to be at the end of the sentence could be selected at the wrong time when going left to right

The Viterbi Algorithm

- A **dynamic programming** algorithm
  - Break down a problem into smaller parts
  - Compute small parts once and re-use later on
- Yes, that Viterbi
  - All USC CS courses are required to present the Viterbi Algorithm
  - Kidding! But it comes up a lot because it's very useful
- Optimal global solution
  - Will be slower than greedy algorithm, but is <u>guaranteed</u> to return the proper argmax $\prod_{i=1}^{N} P(t_i|t_{i-1})P(w_i|t_i)\times P(</s>\mid t_n)$

Viterbi as a Decoder

- The problem of finding the best tag sequence for a given word sequence is sometimes called decoding
- This is because, like spelling correction, etc., HMM can also be viewed as a noisy channel model:
  - Someone wants to send us a sequence of tags P(T)
  - During transmission, "noise" converts each tag to a word P(W|T)
  - We try to decode the observed words back to the original tags
- Decoding is a general term in NLP for inferring hidden variables in a test instance (e.g. finding correct spelling of a misspelled word, determining topic or sentiment of an input, determining the underlying syntactic tree)

Viterbi Intuition

tags ordered by
frequency for each word

<s>	one	dog	bit	</s>
<s>	CD	NN	NN	</s>
	NN	VB	VBD	
	PRP			

- Suppose we have already calculated
  a) the best tag sequence for <s> ... bit that ends in NN
  b) the best tag sequence for <s> ... bit that ends in VBD
- Then, the best sequence would be either
  - sequence a) extended to include </s> or
  - sequence b) extended to include </s>

Viterbi Intuition

tags ordered by
frequency for each word

<s>	one	dog	bit	</s>
<s>	CD	NN	NN	</s>
	NN	VB	VBD	
	PRP			

- But to get
  a) the best tag sequence for <s> ... bit that ends in NN
- Then we have to extend one of:
  - The best tag sequence for <s> ... dog that ends in NN
  - The best tag sequence for <s> ... dog that ends in VB
- And so on...

Viterbi High-Level Picture

- Want to find $\text{argmax}_T P(T|W) = \text{argmax}_T P(T,W) = \text{argmax}_T P(T)P(W|T)$
- Intuition: the best path of length i ending in state t must include the best path of length i-1 to the previous state. So,
  - find the best path of length i-1 to each state
  - consider extending each of these by 1 step, to state t
  - take the best of these options as the best path to state t

Quiz 2

- In Naive Bayes we model P(Y| X1 X2 X3) as P(Y|X1, X2, X3) by applying...
  A. The Law of Total Probability
  B. Bayes' rule
  C. The Bag of Words Assumption
  D. The Naive Bayes Assumption

Viterbi Algorithm

- use a chart v to store partial results as we go
  - T x N table for T possible tags and length N sentence
  - v[t, i] is the probability of the best state sequence for $w_1 \ldots w_i$ that ends in state t
- fill columns left to right, with
  - $v[t, i] = \max_{t'} v[t', i-1] \times P(t|t') \times P(w_i|t_i)$
  - note, the max is over each possible previous tag t'
- also keep a backtrace table b
  - $b[t, i] = \text{argmax}_{t'} v[t', i-1] \times P(t|t') \times P(w_i|t_i)$
  - b can be used afterward to find the chain of tags

Example

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N					
V					
D					
P					
A					

Suppose W = the doctor is in. Our chart is initially empty.

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	a	cat	doctor	in	is	the	very
N	0	.5	.4	0	.1	0	0
V	0	0	.1	0	.9	0	0
D	.3	0	0	0	0	.7	0
P	0	0	0	1	0	0	0
A	0	0	0	.1	0	0	.9

Example

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N					
V					
D					
P					
A					

Suppose W = the doctor is in. Our chart is initially empty.

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Filling in the First Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0				
V	0				
D	.21				
P	0				
A	0				

$$v[N,\text{the}] = P(N|\text{<s>})*P(\text{the}|N) = .3*0=0$$

…

$$v[D,\text{the}] = P(D|\text{<s>})*P(\text{the}|D) = .3*.7=.21$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Second Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	?			
V	0				
D	.21				
P	0				
A	0				

$$v[N,doctor] = \max_{t'} v[t',the]*P(N|t')*P(doctor|N)$$

$$\max(0,0,.21*.9*.4.,0,0) = .0756$$

$$P(N|D)*P(doctor|N) = .9*.4$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Second Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756			
V	0				
D	.21				
P	0				
A	0				

$$v[N,doctor] = \max_{t'} v[t',the]*P(N|t')*P(doctor|N)$$

$$\max(0,0,.21*.9*.4.,0,0) = .0756$$

$$P(N|D)*P(doctor|N) = .9*.4$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Second Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756			
V	0	.00021			
D	.21				
P	0				
A	0				

$$v[V,doctor] = \max_{t'} v[t',the]*P(V|t')*P(doctor|V)$$

$$\max(0,0,.21*.01*.1,0,0) = .00021$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Second Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756			
V	0	.00021			
D	.21	0			
P	0	0			
A	0	0			

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Third Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756			
V	0	.00021			
D	.21	0			
P	0	0			
A	0	0			

$$v[N,is] = \max_{t'} v[t',doctor]*P(N|t')*P(is|N)$$

.0756 * .2 * .1 = .001512

.00021 * .3 * .1 = .0000063

0* .9 * .1 = 0

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Third Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756	.001512		
V	0	.00021	.027216		
D	.21	0	0		
P	0	0	0		
A	0	0	0		

$$v[V,is] = \max_{t'} v[t',doctor]*P(V|t')*P(is|V)$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

max(.0756*.4*.9,
.00021*.05*.9,
0,
0,
0) = .027216

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	$\boxed{.9}$	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Fourth Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756	.001512	0	
V	0	.00021	.027216	0	
D	.21	0	0	0	
P	0	0	0	.0054432	
A	0	0	0		

$$v[P,in] = \max_{t'} v[t',is]*P(P|t')*P(in|P)$$

$$\begin{aligned} \max(&.001512*.3*1,\\ &.027216*.2*1,\\ &0,\\ &0,\\ &0) = .0054432 \end{aligned}$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Fourth Column

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756	.001512	0	
V	0	.00021	.027216	0	
D	.21	0	0	0	
P	0	0	0	.0054432	
A	0	0	0	.00027216	

$$v[A,in] = \max_{t'} v[t',is]*P(A|t')*P(in|A)$$

max(.001512*.04*.1,
.027216*.1*.1,
0,
0,
0) = .00027216

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

End of sentence

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756	.001512	0	
V	0	.00021	.027216	0	
D	.21	0	0	0	.000027216
P	0	0	0	.0054432	
A	0	0	0	.00027216	

$$v[</s>] = \max_{t'} v[t',in]*P(</s>|t')$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

max(0,
0,
0,
.0054432*0,
.00027216*.1) = .000027216

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Completed Chart

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756	.001512	0	
V	0	.00021	.027216	0	
D	.21	0	0	0	.000027216
P	0	0	0	.0054432	
A	0	0	0	.00027216	

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Following the Backtraces

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756	.001512	0	
V	0	.00021	.027216	0	
D	.21	0	0	0	.000027216
P	0	0	0	.0054432	
A	0	0	0	.00027216	
	D	N	V	A	

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Do Transition and Emission Probabilities Need Smoothing?

- **Emissions**: Yes, because if there is any word w in the test data such that P(w|t) = 0 for all tags t, then the whole joint probability P(W, T) will go to 0.
- **Transitions**: Not necessarily, but if any transition probabilities are estimated as 0, that tag bigram will never be predicted
- What are some transitions that should NEVER occur in a bigram HMM?
  - * -> <s>
  - </s> -> *
  - <s> -> </s>

Higher-Order HMMs

- Equations thus far have been for bigram HMMs, i.e. transitions $P(t_i|t_{i-1})$
- But we can increase the order of the n-gram (i.e. $n > 2$). e.g. trigram HMMs = $P(t_i|t_{i-1},t_{i-2})$ [see collins notes]
- As usual, smoothing the transition distributions becomes more important with higher-order models

What Else Can we do?

- Suppose you want to find the likelihood of an input sequence, P(W)
- The HMM models P(T, W); by law of total probability sum over all T and you get P(W)
  - Why do you want to do this?
  - If you have a high P(W) that means your model likes your data; if your data is good data, it's an indication you have a good model
- There are an exponential number of members of T
  - We can use the <u>forward</u> algorithm, which is very similar to Viterbi
  - Replace max with sum (no backpointers needed)

Second Column, Viterbi Algorithm

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756			
V	0				
D	.21				
P	0				
A	0				

$$v[N,doctor] = \max_{t'} v[t',the]*P(N|t')*P(doctor|N)$$

$$\max(0,0,.21*.9*.4.,0,0) = .0756$$

$$P(N|D)*P(doctor|N) = .9*.4$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	$\boxed{.9}$	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	$\boxed{.4}$	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Second Column, Forward Algorithm

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756			
V	0				
D	.21				
P	0				
A	0				

v[N,doctor] = $\sum_{t'}$ v[t',the]*P(N|t')*P(doctor|N)

$\sum$(0,0,.21*.9*.4.,0,0) = .0756

P(N|D)*P(doctor|N) = .9*.4

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	$\boxed{.9}$	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	$\boxed{.4}$	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Third Column, Forward

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756	.001512		
V	0	.00021			
D	.21	0			
P	0	0			
A	0	0			

$$v[N,is] = \max_{t'} v[t',doctor] * P(N|t') * P(is|N)$$

$$.0756 * .2 * .1 = .001512$$

$$.00021 * .3 * .1 = .0000063$$

$$0 * .9 * .1 = 0$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Third Column, Viterbi

v	w_1=the	w_2=doctor	w_3=is	w_4=in	</s>
N	0	.0756	.0015183		
V	0	.00021			
D	.21	0			
P	0	0			
A	0	0			

$$v[N,is] = \sum_{t'} v[t',doctor]*P(N|t')*P(is|N)$$

$$.0756 * .2 * .1 = .001512$$

$$+$$

$$.00021 * .3 * .1 = .0000063$$

$$+ \quad 0* .9 * .1 = 0$$

$$= .0015183$$

T->T	N	V	D	P	A	</s>
<s>	.3	.1	.3	.2	.1	0
N	.2	.4	.01	.3	.04	.05
V	.3	.05	.3	.2	.1	.05
D	.9	.01	.01	.01	.07	0
P	.4	.05	.4	.1	.05	0
A	.1	.5	.1	.1	.1	.1

T->W	doctor	in	is	the
N	.4	0	.1	0
V	.1	0	.9	0
D	0	0	0	.7
P	0	1	0	0
A	0	.1	0	0

Summary

- Part-of-speech tagging is a sequence labeling task
- HMM: A generative model of sentences using hidden state sequence
- HMM uses two sources of information to resolve ambiguity
  - The words themselves
  - The tags of nearby words
- Can be viewed as a probabilistic FSM
- Algorithms for computing probability efficiently:
- Greedy tagging: Fast but suboptimal
- Dynamic Programming algorithms to compute
  - Best tag sequence given words: Viterbi algorithm
  - Likelihood of corpus: Forward algorithm