Natural Language Processing

Info 159/259

Lecture 10: LLM Wrap up, Sequence labeling (for POS)

Many slides & instruction ideas borrowed from:

David Bamman, Mohit Iyyer, Greg Durrett & Diyi Yang

Logistics

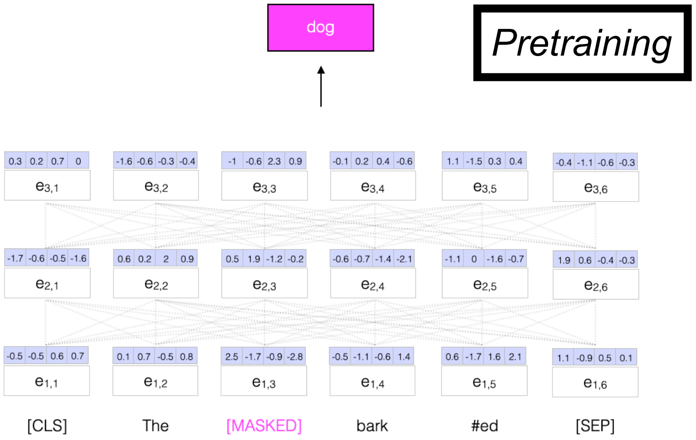
- Exam1 is being graded and reviewed.
- No homework this week
  - Homework 4 will be released towards end of the week.
- AP1 is due this Sunday March 3
- Quiz 4 will be out this Friday afternoon (Due Monday).
- Today: Wrapping up LLMs, Sequence Tagging

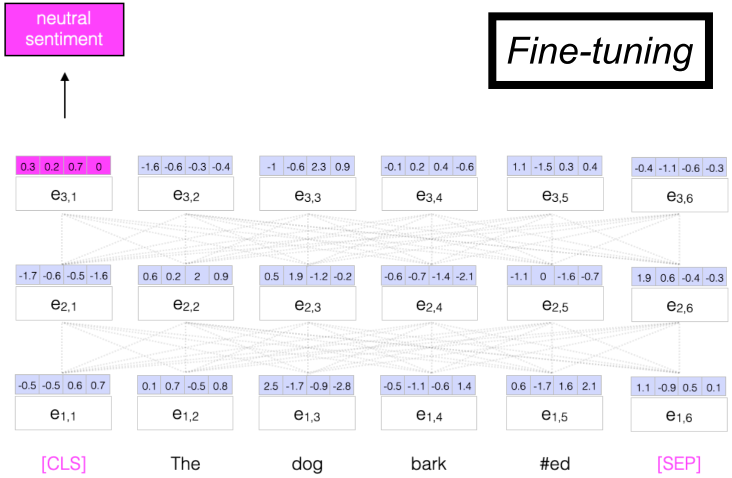
Evolution of Paradigm

Before 2014	Fully Supervised (feature Engineering)
2014-2019	Architecture Engineering
2019-2021	Pretrain+Finetune: Objective Engineering
2021-…	Pretrain, prompt, predict: Prompt Engineering

Pretrain + Fine-tune

- The LLM backbone gets trained with its objectives
- The backbone gets fine-tuned for specific task in supervised manner





Everything is language modeling

The director of *2001: A Space Odyssey* is ______________

The French translation of “cheese” is ______________

The sentiment of “I really hate this movie” is _____________

In Context Learning

- Provide the pattern; LLM is expected to continue with it.
- Use the off-the-shelf model:
  - No Gradient update and parameter change.

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:
cheese =>
```

task description

prompt

Brown et al. (2020, “Language Models are Few-Shot Learners”
https://arxiv.org/pdf/2005.14165.pdf

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
cheese =>
```

task description

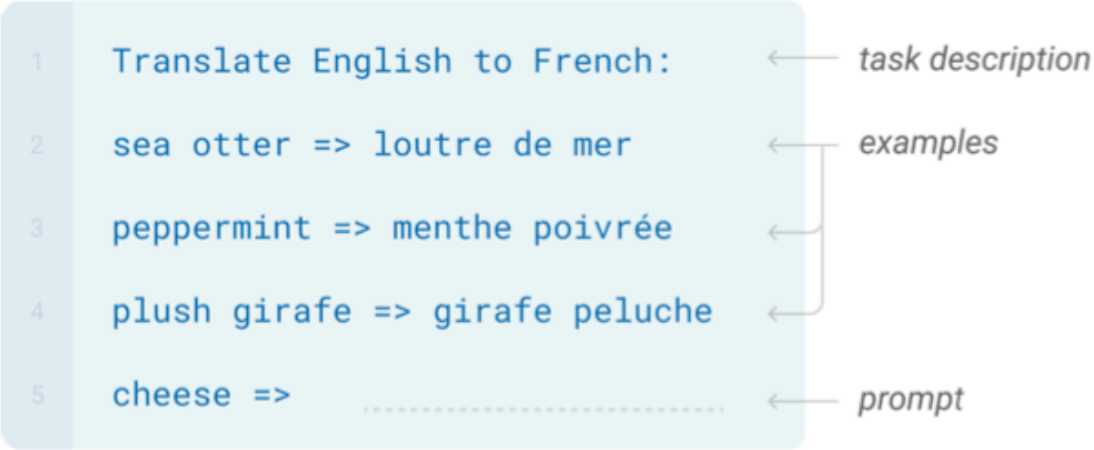
example

prompt

Brown et al. (2020, “Language Models are Few-Shot Learners”
https://arxiv.org/pdf/2005.14165.pdf

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Brown et al. (2020, “Language Models are Few-Shot Learners”
https://arxiv.org/pdf/2005.14165.pdf

Prompt engineering

- Manual prompt design: encoding domain knowledge into prompt templates that are likely to generate a response in the output space.

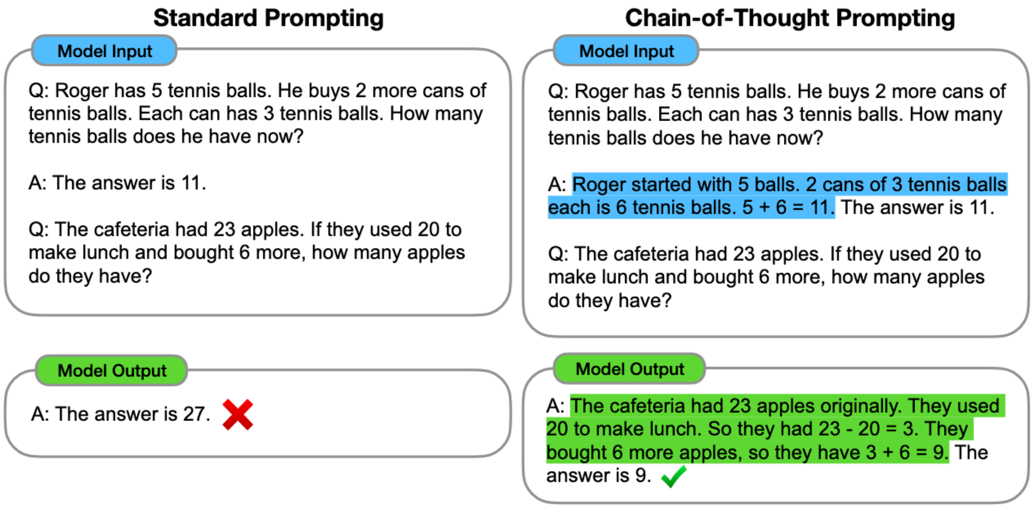
Chain-of-thought Prompting

- Tasks that require multi-step reasoning.
    - Computation: entirely on the LM.
- One/few shot learning: not enough
- Improves with breaking down the task.

```
4621012097 + 3367370272 = 7988382369
7263297356 + 3675827524 = 10939124880
4764893393 + 9123518451 = 13888411844
5692118231 + 1499193323 = 7191311554
8504625225 + 5470236074 = ?
```

Chain-of-thought Prompting

- Tasks that require multi-step reasoning.
- One/few shot learning: not enough
- Improves with breaking down the task.



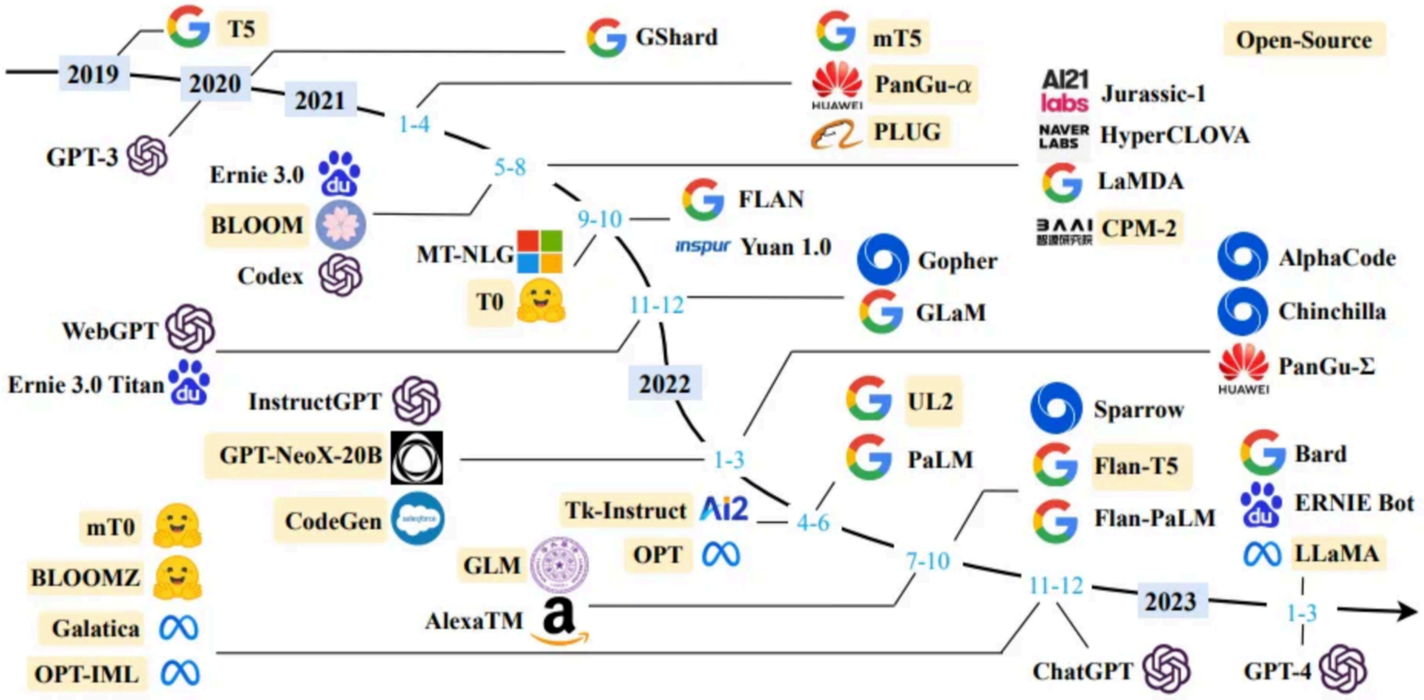
Wei et al, 2022

Chain-of-thought Prompting

	MultiArith	GSM8K
Zero-Shot	**17.7**	**10.4**
Few-Shot (2 samples)	33.7	15.6
Few-Shot (8 samples)	33.8	15.6
Zero-Shot-CoT	**78.7**	**40.7**
Few-Shot-CoT (2 samples)	84.8	41.3
Few-Shot-CoT (4 samples : First) (*1)	89.2	-
Few-Shot-CoT (4 samples : Second) (*1)	90.5	-
Few-Shot-CoT (8 samples)	93.0	48.7
Zero-Plus-Few-Shot-CoT (8 samples) (*2)	**92.8**	**51.5**
Finetuned GPT-3 175B [Wei et al., 2022]	-	33
Finetuned GPT-3 175B + verifier [Wei et al., 2022]	-	55
PaLM 540B: Zero-Shot	**25.5**	**12.5**
PaLM 540B: Zero-Shot-CoT	**66.1**	**43.0**
PaLM 540B: Zero-Shot-CoT + self consistency	**89.0**	**70.1**
PaLM 540B: Few-Shot [Wei et al., 2022]	-	17.9
PaLM 540B: Few-Shot-CoT [Wei et al., 2022]	-	56.9
PaLM 540B: Few-Shot-CoT + self consistency [Wang et al., 2022]	-	74.4

Kojima et al, 2022

Explosion of LLMs



Classic NLP

- Sequence Modeling: POS tagging, Named Entity Recognition
- Syntactic and Dependency Parsing
- Lexical Semantics
- Discourse: Coreference Resolution

context

everyone likes	______________	
a bottle of	______________	is on the table
	______________	makes you drunk
a cocktail with	______________	and seltzer

Distribution

- Words that appear in similar contexts have similar representations (and similar meanings, by the distributional hypothesis).

Parts of speech

- Parts of speech are categories of words defined distributionally by the morphological and syntactic contexts a word appears in.

Morphological distribution

POS often defined by distributional properties; verbs = the class of words that each combine with the same set of affixes

	-s	-ed	-ing
walk	walks	walked	walking
slice	slices	sliced	slicing
believe	believes	believed	believing
of	*ofs	*ofed	*ofing
red	*reds	*redded	*reding

Bender 2013

Morphological distribution

We can look to the function of the affix (denoting past tense) to include irregular inflections.

	-s	-ed	-ing
walk	walks	walked	walking
sleep	sleeps	slept	sleeping
eat	eats	ate	eating
give	gives	gave	giving

Bender 2013

Syntactic distribution

- Substitution test: if a word is replaced by another word, does the sentence remain grammatical?

Kim saw the	elephant	before we did
	dog	
	idea	
	*of	
	*goes	

Bender 2013

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Kim saw the	elephant	before we did
	*Sandy	
Kim *arrived the	elephant	before we did

both nouns but common vs. proper

both verbs but transitive vs. intransitive

Bender 2013

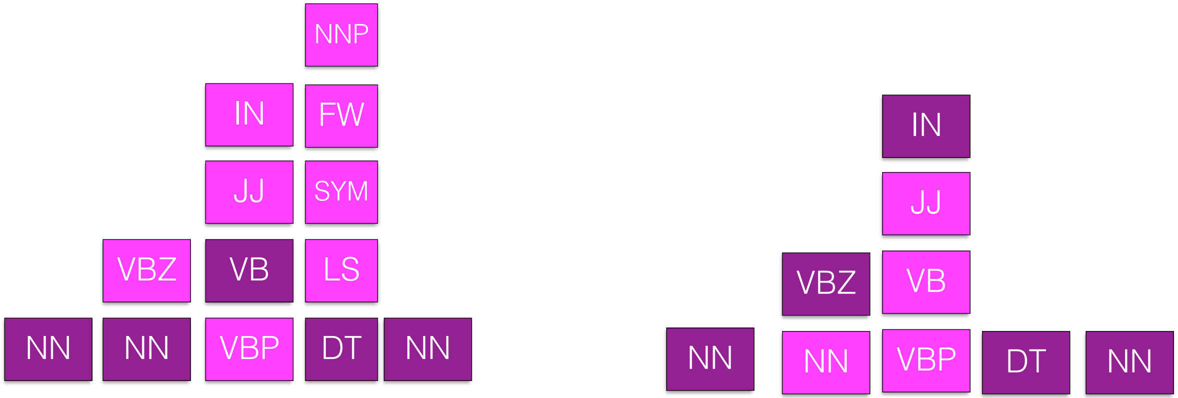
Nouns	People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural
Verbs	Actions, processes. Inflected for tense, aspect, number, person
Adjectives	Properties, qualities. Usually modify nouns
Adverbs	Qualify the manner of verbs (“She ran downhill extremely quickly yesteray”)
Determiner	Mark the beginning of a noun phrase (“a dog”)
Pronouns	Refer to a noun phrase (he, she, it)
Prepositions	Indicate spatial/temporal relationships (on the table)
Conjunctions	Conjoin two phrases, clauses, sentences (and, or)

Open class	Nouns	fax, affluenza, subtweet, bitcoin, cronut, emoji, listicle, mocktail, selfie, skort
	Verbs	text, chillax, manspreading, photobomb, unfollow, google
	Adjectives	crunk, amazeballs, post-truth, woke
	Adverbs	hella, wicked
Closed class	Determiner	
	Pronouns	
	Prepositions	English has a new preposition, because internet [Garber 2013; Pullum 2014]
	Conjunctions	

OOV? Guess Noun

POS tagging

Labeling the tag that's correct for the context.



Fruit flies like a banana

Time flies like an arrow

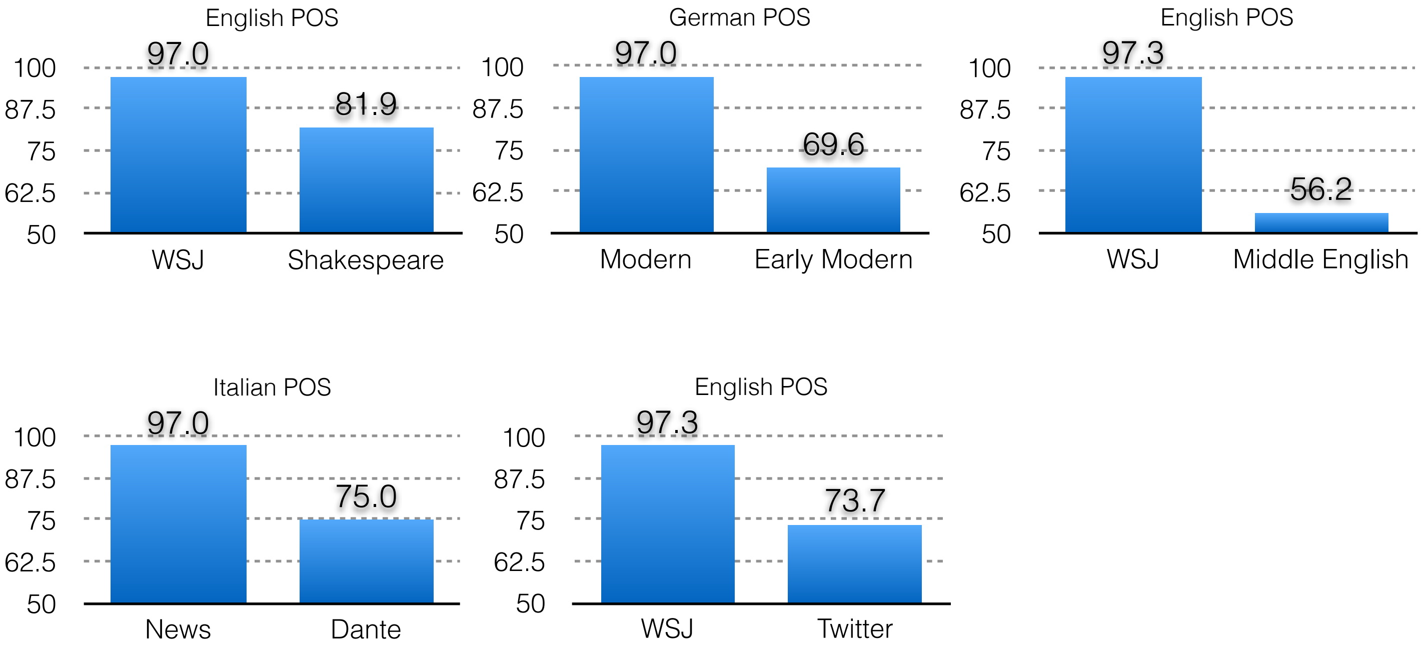
(Just tags in evidence within the Penn Treebank — more are possible!)

State of the art

- Baseline: Most frequent class = 92.34%
- Token accuracy: 98% (English news)
  [Bohnet et al. 2018]
    - Optimistic: includes punctuation, words with only one tag (deterministic tagging)
    - Substantial drop across domains (e.g., train on news, test on literature)
- Whole sentence accuracy: 55%

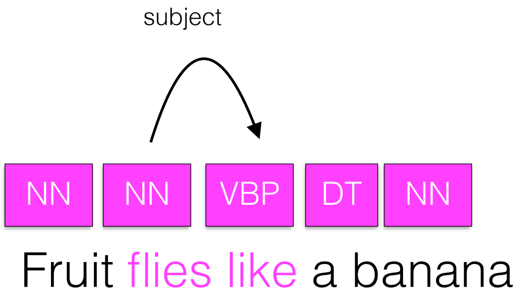
Manning 2011

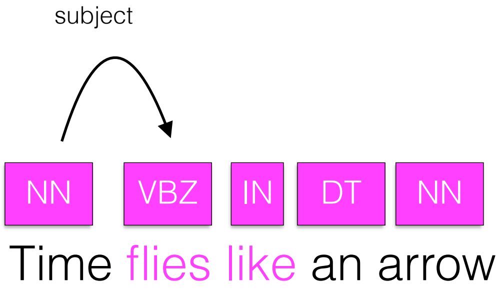
Domain difference



Why is part of speech tagging useful?

POS indicative of syntax





POS indicative of MWE

at least one adjective/noun or noun phrase | and definitely one noun

((A | N)+ | ((A | N)*(NP))(A | N)*)N

AN: linear function; lexical ambiguity; mobile phase
NN: regression coefficients; word sense; surface area
AAN: Gaussian random variable; lexical conceptual paradigm; aqueous mobile phase
ANN: cumulative distribution function; lexical ambiguity resolution; accessible surface area
NAN: mean squared error; domain independent set; silica based packing
NNN: class probability function; text analysis system; gradient elution chromatography
NPN: degrees of freedom; [*no example*]; energy of adsorption

Justeson and Katz 1995

POS is indicative of pronunciation

Noun	Verb
My conduct is great	I conduct myself well
She won the contest	I contest the ticket
He is my escort	He escorted me
That is an insult	Don’t insult me
Rebel without a cause	He likes to rebel
He is a suspect	I suspect him

Tagsets

- Penn Treebank
- Universal Dependencies
- Twitter POS

Verbs

tag	description	example
VB	base form	I want to like
VBD	past tense	I/we/he/she/you liked
VBG	present participle	He was liking it
VBN	past participle	I had liked it
VBP	present (non 3rd-sing)	I like it
VBZ	present (3rd-sing)	He likes it
MD	modal verbs	He can go

Nouns

	tag	description	example
non-proper	NN	non-proper, singular or mass	company
	NNS	non-proper, plural	companies
proper	NNP	proper, singular	Carolina
	NNPS	proper, plural	Carolinas

DT (Article)

- Articles (a, the, every, no)
- Indefinite determiners (another, any, some, each)
- That, these, this, those when preceding noun
- All, both when not preceding another determiner or possessive pronoun

Santorini 1990

```
65548 the/dt
26970 a/dt
 4405 an/dt
 3115 this/dt
 2117 some/dt
 2102 that/dt
 1274 all/dt
 1085 any/dt
  953 no/dt
  778 those/dt
```

JJ
(Adjectives)

- General adjectives
  - *happy person*
  - *new mail*
- Ordinal numbers
  - *fourth person*

```
2002 other/jj
1925 new/jj
1563 last/jj
1174 many/jj
1142 such/jj
1058 first/jj
 824 major/jj
 715 federal/jj
 698 next/jj
 644 financial/jj
```

Santorini 1990

RB (Adverb)

- Most words that end in -ly
- Degree words (quite, too, very)
- Negative markers: not, n't, never

```
4410 n't/rb
2071 also/rb
1858 not/rb
1109 now/rb
1070 only/rb
1027 as/rb
 961 even/rb
 839 so/rb
 810 about/rb
 804 still/rb
```

Santorini 1990

IN (preposition, subordinating conjunction)

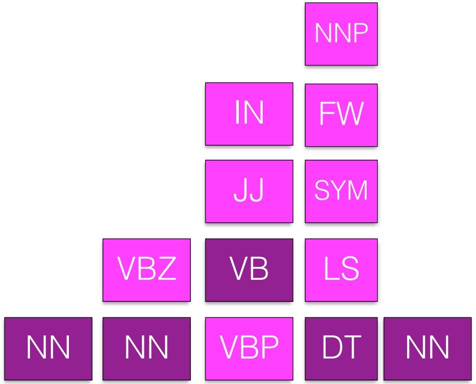
- All prepositions (except *to*) and subordinating conjunctions
  - He jumped on the table because he was excited

```
31111 of/in
22967 in/in
11425 for/in
 7181 on/in
 6684 that/in
 6399 at/in
 6229 by/in
 5940 from/in
 5874 with/in
 5239 as/in
```

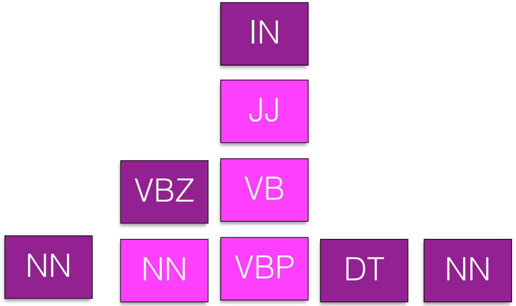
Santorini 1990

POS tagging

Labeling the tag that's correct for the context.



Fruit flies like a banana



Time flies like an arrow

(Just tags in evidence within the Penn Treebank — more are possible!)

Sequence labeling

$$x = \{x_1, \ldots, x_n\}$$

$$y = \{y_1, \ldots, y_n\}$$

- For a set of inputs x with n sequential time steps, one corresponding label y_i for each x_i

Named entity recognition

B-PERS	I-PERS	O	O	B-ORG
Natalie	Johnson	works	for	UCB

3 or 4-class:

- person
- location
- organization
- (misc)

7-class:

- person
- location
- organization
- time
- money
- percent
- date

POS tagging training data

- Wall Street Journal (~1M tokens, 45 tags, English)
- Universal Dependencies (universal dependency treebanks for many languages; common POS tags for all)
  https://github.com/UniversalDependencies

Majority class

- Pick the label each word is seen most often with in the training data

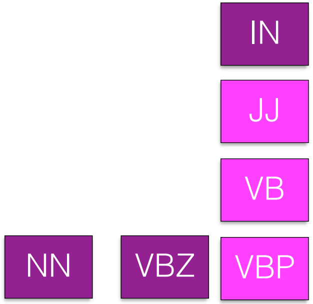
fruit	flies	like	a	banana
NN 12	VBZ 7	VB 74	FW 8	NN 3
	NNS 1	VBP 31	SYM 13	
		JJ 28	LS 2	
		IN 533	JJ 2	
			IN 1	
			DT 25820	
			NNP 2	

Sequences

- Models that make independent predictions for elements in a sequence can reason over expressive representations of the input x (including correlations among inputs at different time steps x_i and x_j).

- But they don't capture another important source of information: correlations in the labels y.

Sequences



Time flies like an arrow

Sequences

Most common tag bigrams in
Penn Treebank training

DT	NN	41909
NNP	NNP	37696
NN	IN	35458
IN	DT	35006
JJ	NN	29699
DT	JJ	19166
NN	NN	17484
NN	,	16352
IN	NNP	15940
NN	.	15548
JJ	NNS	15297
NNS	IN	15146
TO	VB	13797
NNP	,	13683
IN	NN	11565

Sequences

x	time	flies	like	an	arrow
y	NN	VBZ	IN	DT	NN

$$P(y = \text{NN VBZ IN DT NN} \mid x = \text{time flies like an arrow})$$

Generative vs. Discriminative models

- Generative models specify a joint distribution over the labels and the data. With this you could generate new data

$$P(x, y) = P(y)\, P(x \mid y)$$

- Discriminative models specify the conditional distribution of the label y given the data x. These models focus on how to discriminate between the classes

$$P(y \mid x)$$

Generative Model

$$P(x, y) = P(x \mid y)\, P(y)$$

x	time	flies	like	an	arrow
y	NN	VBZ	IN	DT	NN

$$\max_y P(x \mid y) P(y)$$

How do we parameterize these probabilities when x and y are sequences?

Estimating the seq. Prob.

$$\begin{aligned} P(y_i, \ldots, y_n) = {} & P(y_1) \\ & \times P(y_2 \mid y_1) \\ & \times P(y_3 \mid y_1, y_2) \\ & \ldots \\ & \times P(y_n \mid y_1, \ldots, y_{n-1}) \end{aligned}$$

- Remember: a Markov assumption is an approximation to this exact decomposition (the chain rule of probability)

Hidden Markov Model

$$\max_y P(x \mid y) P(y)$$

Prior probability of label sequence

$$P(y) = P(y_1, \ldots, y_n)$$

$$P(y_1, \ldots, y_n) \approx \prod_{i=1}^{n} P(y_i \mid y_{i-1})$$

- We'll make a first-order Markov assumption and calculate the joint probability as the product the individual factors conditioned only on the previous tag.

Hidden Markov Model

First-order HMM

$$P(y_1,\dots,y_n) \approx \prod_{i=1}^{n} P(y_i \mid y_{i-1})$$

Second-order HMM

$$P(y_1,\dots,y_n) \approx \prod_{i=1}^{n} P(y_i \mid y_{i-2}, y_{i-1})$$

Third-order HMM

$$P(y_1,\dots,y_n) \approx \prod_{i=1}^{n} P(y_i \mid y_{i-3}, y_{i-2}, y_{i-1})$$

Hidden Markov Model

$$\boxed{\max_y P(x \mid y)P(y)}$$

$$P(x \mid y) = P(x_1, \ldots, x_n \mid y_1, \ldots, y_n)$$

$$P(x_1, \ldots, x_n \mid y_1, \ldots, y_n) \approx \prod_{i=1}^{N} P(x_i \mid y_i)$$

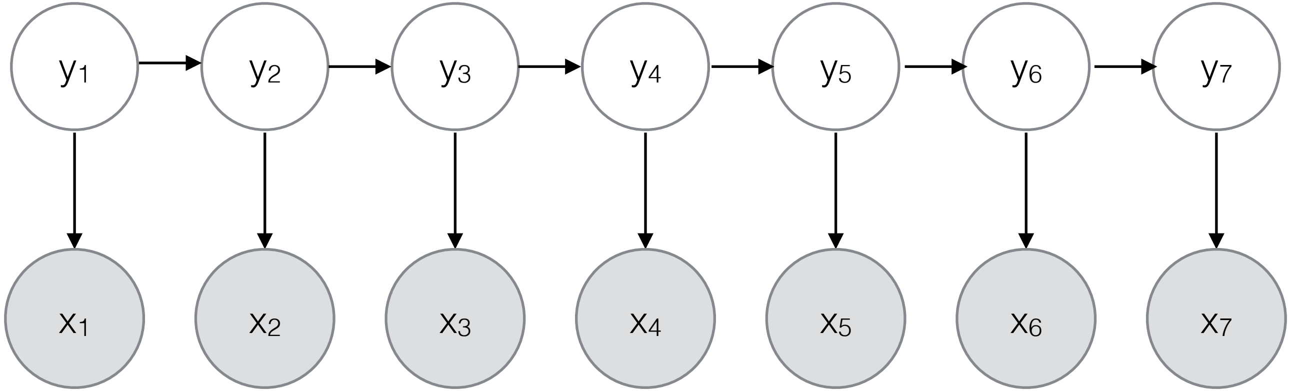
- Here again we'll make a strong assumption: the probability of the word we see at a given time step is only dependent on its own label, no matter the Markov order used for P(y).

HMM

$$P(x_1,..,x_n,y_1,..,y_n) \approx \prod_{i=1}^{n} P(y_i|\,y_{i-1}) \prod_{i=1}^{n} P(x_i|\,y_i)$$

HMM

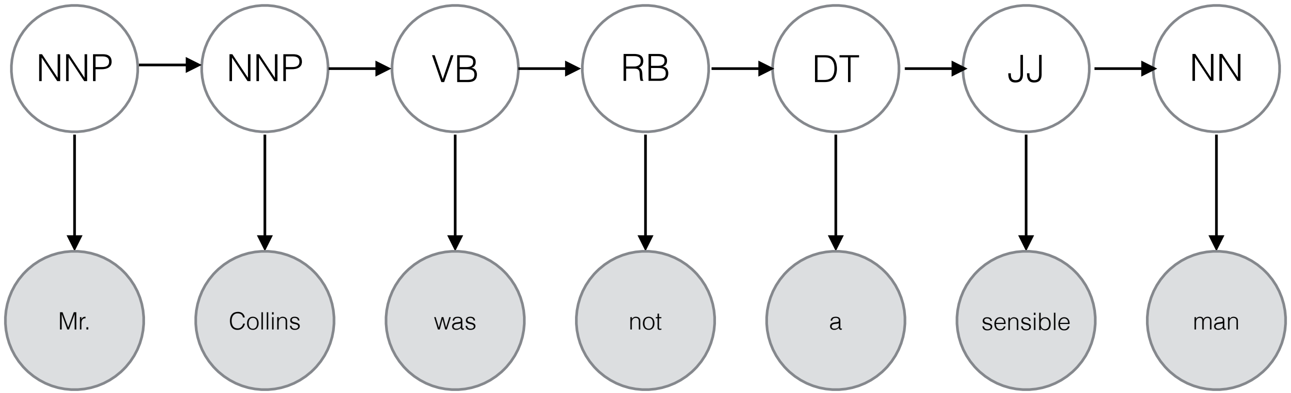
$P(y_3 \mid y_2)$



$P(x_3 \mid y_3)$

HMM

$$P(VB \mid NNP)$$



$$P(was \mid VB)$$

Parameter estimation

$$P(y_t \mid y_{t-1}) \qquad \frac{c(y_1, y_2)}{c(y_1)}$$

MLE for both is just counting
and normalizing

$$P(x_t \mid y_t) \qquad \frac{c(x, y)}{c(y)}$$

Transition probabilities

	NNP	**MD**	**VB**	**JJ**	**NN**	**RB**	**DT**
$<s>$	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

Figure 10.5 The A transition probabilities $P(t_i|t_{i-1})$ computed from the WSJ corpus without smoothing. Rows are labeled with the conditioning event; thus $P(VB|MD)$ is 0.7968.

Emission probabilities

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0.000097	0
NN	0	0.000200	0.000223	0.000006	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Figure 10.6 Observation likelihoods *B* computed from the WSJ corpus without smoothing.

Decoding

- **Decoding:** Finding the optimal path for a sequence using transition and emission parameters.

- **Greedy decoding**: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit	flies	like	a	banana
NN	VB	IN	DT	NN

Decoding

DT	NN	VBD	IN	DT	NN	???
The	horse	raced	past	the	barn	fell

Decoding

DT	NN	VBD	IN	DT	NN	???
The	horse	raced	past	the	barn	fell
DT	NN	VBN	IN	DT	NN	VBD

Information later on in the sentence can influence the best tags earlier on.

All paths

END							
DT							
NNP							
VB							
NN							
MD							
START							
	^	Janet	will	back	the	bill	$

Ideally, what we want is to calculate the joint probability of each path and pick the one with the highest probability. But for N time steps and K labels, number of possible paths = K^N

5 word sentence with 45 Penn Treebank tags

45^5 = 184,528,125 different paths

45^{20} = 1.16e33 different paths

Viterbi algorithm

- Basic idea: if an optimal path through a sequence uses label L at time T, then it must have used an optimal path to get to label L at time T

- We can discard all non-optimal paths up to label L at time T

END							
DT							
NNP							
VB							
NN							
MD							
START							
	^	Janet	will	back	the	bill	$

- At each time step t ending in label K, we find the max probability of any path that led to that state

END		
DT		v_1(DT)
NNP		v_1(NNP)
VB		v_1(VB)
NN		v_1(NN)
MD		v_1(MD)
START		

Janet

What's the HMM probability of ending in Janet = NNP?

$$P(y_t \mid y_{t-1})P(x_t \mid y_t)$$

$$P(\text{NNP} \mid \text{START})P(Janet \mid \text{NNP})$$

END		
DT		v_1(DT)
NNP		v_1(NNP)
VB		v_1(VB)
NN		v_1(NN)
MD		v_1(MD)
START		
		Janet

Best path through time step 1 ending in tag y (trivially - best path for all is just START)

$$v_1(y) = \max_{u \in \mathcal{Y}} \left[P(y_t = y \mid y_{t-1} = u) P(x_t \mid y_t = y) \right]$$

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			
		Janet	will

What's the max HMM probability of ending in will = MD?

First, what's the HMM probability of a single path ending in will = MD?

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			
		Janet	will

$$P(y_1 \mid START)P(x_1 \mid y_1) \times P(y_2 = \mathrm{MD} \mid y_1)P(x_2 \mid y_2 = \mathrm{MD})$$

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			
		Janet	will

Best path through time step 2 ending in tag MD

$$P(\mathrm{DT} \mid \mathrm{START}) \times P(Janet \mid \mathrm{DT}) \times P(y_t = \mathrm{MD} \mid P(y_{t-1} = \mathrm{DT}) \times P(will \mid y_t = \mathrm{MD})$$

$$P(\mathrm{NNP} \mid \mathrm{START}) \times P(Janet \mid \mathrm{NNP}) \times P(y_t = \mathrm{MD} \mid P(y_{t-1} = \mathrm{NNP}) \times P(will \mid y_t = \mathrm{MD})$$

$$P(\mathrm{VB} \mid \mathrm{START}) \times P(Janet \mid \mathrm{VB}) \times P(y_t = \mathrm{MD} \mid P(y_{t-1} = \mathrm{VB}) \times P(will \mid y_t = \mathrm{MD})$$

$$P(\mathrm{NN} \mid \mathrm{START}) \times P(Janet \mid \mathrm{NN}) \times P(y_t = \mathrm{MD} \mid P(y_{t-1} = \mathrm{NN}) \times P(will \mid y_t = \mathrm{MD})$$

$$P(\mathrm{MD} \mid \mathrm{START}) \times P(Janet \mid \mathrm{MD}) \times P(y_t = \mathrm{MD} \mid P(y_{t-1} = \mathrm{MD}) \times P(will \mid y_t = \mathrm{MD})$$

Let's say the best path ending y_2=MD includes y_1=NNP, with probability 0.0090.

Under our first-order Markov assumption, *if* y_2=MD is in the best path for the complete sequence, y_1=NNP must be as well. That means we can forget every other path ending in y_2=MD that does not have y_1=NNP.

END			
DT		v_1(DT)	v_2(DT)
NNP		v_1(NNP)	v_2(NNP)
VB		v_1(VB)	v_2(VB)
NN		v_1(NN)	v_2(NN)
MD		v_1(MD)	v_2(MD)
START			
		Janet	will

0.0003	$P(\text{DT} \mid \text{START}) \times P(\mathit{Janet} \mid \text{DT}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{DT}) \times P(\mathit{will} \mid y_t = \text{MD})$
0.0090	$P(\text{NNP} \mid \text{START}) \times P(\mathit{Janet} \mid \text{NNP}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{NNP}) \times P(\mathit{will} \mid y_t = \text{MD})$
0.0001	$P(\text{VB} \mid \text{START}) \times P(\mathit{Janet} \mid \text{VB}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{VB}) \times P(\mathit{will} \mid y_t = \text{MD})$
0.0045	$P(\text{NN} \mid \text{START}) \times P(\mathit{Janet} \mid \text{NN}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{NN}) \times P(\mathit{will} \mid y_t = \text{MD})$
0.0002	$P(\text{MD} \mid \text{START}) \times P(\mathit{Janet} \mid \text{MD}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{MD}) \times P(\mathit{will} \mid y_t = \text{MD})$

None of the grey out paths could *possibly* be in the final optimal path, so we can forget them going forward.

To calculate this full probability, notice that we can reuse information we've already computed.

$$\underbrace{P(\text{DT} \mid \text{START}) \times P(Janet \mid \text{DT})}_{v_1(DT)} \times P(y_t = MD \mid P(y_{t-1} = DT) \times P(will \mid y_t = MD)$$

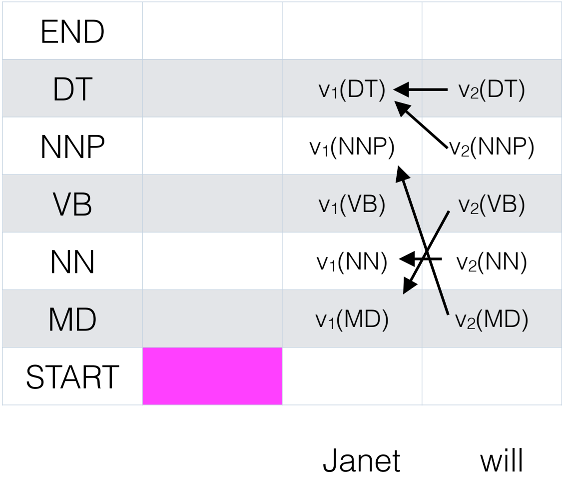
$$\underbrace{P(\text{NNP} \mid \text{START}) \times P(Janet \mid \text{NNP})}_{v_1(NNP)} \times P(y_t = MD \mid P(y_{t-1} = NNP) \times P(will \mid y_t = MD)$$

$$\underbrace{P(\text{VB} \mid \text{START}) \times P(Janet \mid \text{VB})}_{v_1(VB)} \times P(y_t = MD \mid P(y_{t-1} = VB) \times P(will \mid y_t = MD)$$

$\ldots$

END			
DT		v_1(DT)	v_2(DT)
NNP		v_1(NNP)	v_2(NNP)
VB		v_1(VB)	v_2(VB)
NN		v_1(NN)	v_2(NN)
MD		v_1(MD)	v_2(MD)
START			
		Janet	will

$$v_t(y) = \max_{u \in \mathcal{Y}} \left[v_{t-1}(u) \times P(y_t = y \mid y_{t-1} = u) P(x_t \mid y_t = y) \right]$$



Every y at time step t may have a different u at time step t-1 that leads to its max.

Once we've determined that u for each y, we can forget all of the other values of u for that each y, since we know they cannot be on the optimal path for the entire sequence.

$$v_t(y) = \max_{u \in \mathcal{Y}} \left[v_{t-1}(u) \times P(y_t = y \mid y_{t-1} = u) P(x_t \mid y_t = y) \right]$$

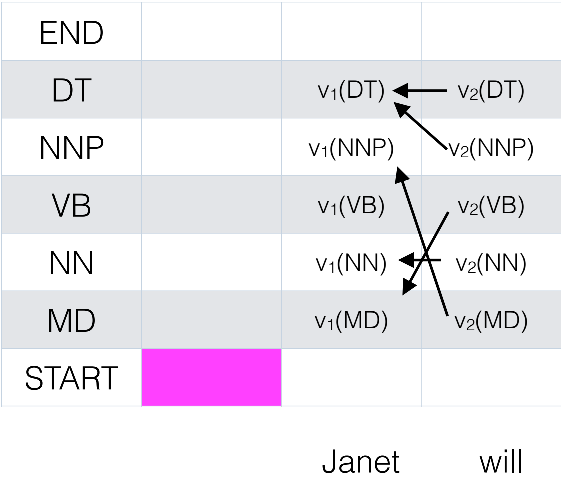
END				
DT		$v_1(DT)$	$v_2(DT)$	$v_3(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$	$v_3(NNP)$
VB		$v_1(VB)$	$v_2(VB)$	$v_3(VB)$
NN		$v_1(NN)$	$v_2(NN)$	$v_3(NN)$
MD		$v_1(MD)$	$v_2(MD)$	$v_3(MD)$
START				
		Janet	will	back

25 paths ending in back = VB

$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = DT)P(x_2 = will \mid y_2 = DT)P(y_2 = DT \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = DT)P(x_2 = will \mid y_2 = DT)P(y_2 = DT \mid y_1 = NNP)P(x_1 = Janet \mid y_1 = NNP)P(y_1 = NNP \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = DT)P(x_2 = will \mid y_2 = DT)P(y_2 = DT \mid y_1 = VB)P(x_1 = Janet \mid y_1 = VB)P(y_1 = VB \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = DT)P(x_2 = will \mid y_2 = DT)P(y_2 = DT \mid y_1 = NN)P(x_1 = Janet \mid y_1 = NN)P(y_1 = NN \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = DT)P(x_2 = will \mid y_2 = DT)P(y_2 = DT \mid y_1 = MD)P(x_1 = Janet \mid y_1 = MD)P(y_1 = MD \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NNP)P(x_2 = will \mid y_2 = NNP)P(y_2 = NNP \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NNP)P(x_2 = will \mid y_2 = NNP)P(y_2 = NNP \mid y_1 = NNP)P(x_1 = Janet \mid y_1 = NNP)P(y_1 = NNP \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NNP)P(x_2 = will \mid y_2 = NNP)P(y_2 = NNP \mid y_1 = VB)P(x_1 = Janet \mid y_1 = VB)P(y_1 = VB \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NNP)P(x_2 = will \mid y_2 = NNP)P(y_2 = NNP \mid y_1 = NN)P(x_1 = Janet \mid y_1 = NN)P(y_1 = NN \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NNP)P(x_2 = will \mid y_2 = NNP)P(y_2 = NNP \mid y_1 = MD)P(x_1 = Janet \mid y_1 = MD)P(y_1 = MD \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = VB)P(x_2 = will \mid y_2 = VB)P(y_2 = VB \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = VB)P(x_2 = will \mid y_2 = VB)P(y_2 = VB \mid y_1 = NNP)P(x_1 = Janet \mid y_1 = NNP)P(y_1 = NNP \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = VB)P(x_2 = will \mid y_2 = VB)P(y_2 = VB \mid y_1 = VB)P(x_1 = Janet \mid y_1 = VB)P(y_1 = VB \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = VB)P(x_2 = will \mid y_2 = VB)P(y_2 = VB \mid y_1 = NN)P(x_1 = Janet \mid y_1 = NN)P(y_1 = NN \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = VB)P(x_2 = will \mid y_2 = VB)P(y_2 = VB \mid y_1 = MD)P(x_1 = Janet \mid y_1 = MD)P(y_1 = MD \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NN)P(x_2 = will \mid y_2 = NN)P(y_2 = NN \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NN)P(x_2 = will \mid y_2 = NN)P(y_2 = NN \mid y_1 = NNP)P(x_1 = Janet \mid y_1 = NNP)P(y_1 = NNP \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NN)P(x_2 = will \mid y_2 = NN)P(y_2 = NN \mid y_1 = VB)P(x_1 = Janet \mid y_1 = VB)P(y_1 = VB \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NN)P(x_2 = will \mid y_2 = NN)P(y_2 = NN \mid y_1 = NN)P(x_1 = Janet \mid y_1 = NN)P(y_1 = NN \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NN)P(x_2 = will \mid y_2 = NN)P(y_2 = NN \mid y_1 = MD)P(x_1 = Janet \mid y_1 = MD)P(y_1 = MD \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = MD)P(x_2 = will \mid y_2 = MD)P(y_2 = MD \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = MD)P(x_2 = will \mid y_2 = MD)P(y_2 = MD \mid y_1 = NNP)P(x_1 = Janet \mid y_1 = NNP)P(y_1 = NNP \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = MD)P(x_2 = will \mid y_2 = MD)P(y_2 = MD \mid y_1 = VB)P(x_1 = Janet \mid y_1 = VB)P(y_1 = VB \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = MD)P(x_2 = will \mid y_2 = MD)P(y_2 = MD \mid y_1 = NN)P(x_1 = Janet \mid y_1 = NN)P(y_1 = NN \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = MD)P(x_2 = will \mid y_2 = MD)P(y_2 = MD \mid y_1 = MD)P(x_1 = Janet \mid y_1 = MD)P(y_1 = MD \mid START)$$

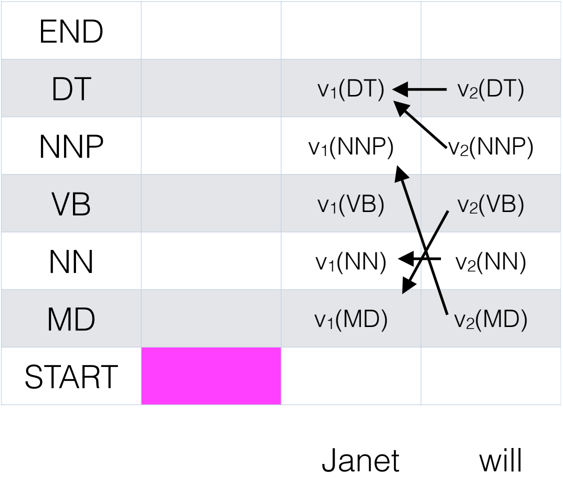
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = DT)P(x_2 = will \mid y_2 = DT)P(y_2 = DT \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NNP)P(x_2 = will \mid y_2 = NNP)P(y_2 = NNP \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = MD)P(x_2 = will \mid y_2 = MD)P(y_2 = MD \mid y_1 = NNP)P(x_1 = Janet \mid y_1 = NNP)P(y_1 = NNP \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NN)P(x_2 = will \mid y_2 = NN)P(y_2 = NN \mid y_1 = NN)P(x_1 = Janet \mid y_1 = NN)P(y_1 = NN \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = VB)P(x_2 = will \mid y_2 = VB)P(y_2 = VB \mid y_1 = MD)P(x_1 = Janet \mid y_1 = MD)P(y_1 = MD \mid START)$$

In calculating the best path ending in x_3=back and y_3=VB, we can forget every other path that we've already determined to be suboptimal.



$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = DT)P(x_2 = will \mid y_2 = DT)P(y_2 = DT \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NNP)P(x_2 = will \mid y_2 = NNP)P(y_2 = NNP \mid y_1 = DT)P(x_1 = Janet \mid y_1 = DT)P(y_1 = DT \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = MD)P(x_2 = will \mid y_2 = MD)P(y_2 = MD \mid y_1 = NNP)P(x_1 = Janet \mid y_1 = NNP)P(y_1 = NNP \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = NN)P(x_2 = will \mid y_2 = NN)P(y_2 = NN \mid y_1 = NN)P(x_1 = Janet \mid y_1 = NN)P(y_1 = NN \mid START)$$
$$P(x_3 = back \mid y_3 = VB)P(y_3 = VB \mid y_2 = VB)P(x_2 = will \mid y_2 = VB)P(y_2 = VB \mid y_1 = MD)P(x_1 = Janet \mid y_1 = MD)P(y_1 = MD \mid START)$$

In calculating the best path ending in x_3=back and y_3=VB, we can forget every other path that we've already determined to be suboptimal.



END				
DT		v_1(DT)	v_2(DT)	v_3(DT)
NNP		v_1(NNP)	v_2(NNP)	v_3(NNP)
VB		v_1(VB)	v_2(VB)	v_3(VB)
NN		v_1(NN)	v_2(NN)	v_3(NN)
MD		v_1(MD)	v_2(MD)	v_3(MD)
START				

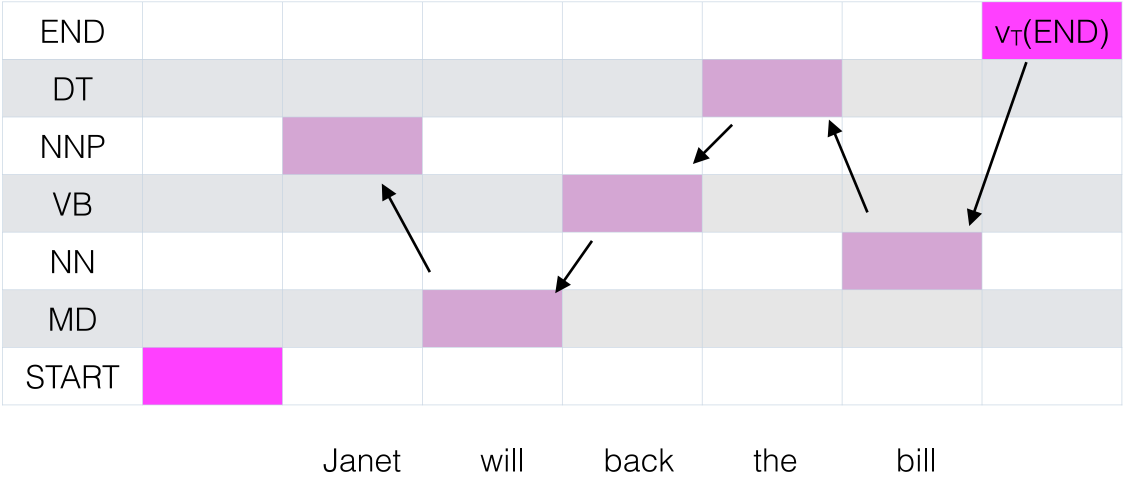
Janet will back

So for every label at every time step, we only need to keep track of which label at the previous time step *t*-1 led to the highest joint probability at that time step *t*.

END						
DT		$v_1(DT)$	$v_2(DT)$	$v_3(DT)$	$v_4(DT)$	$v_5(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$	$v_3(NNP)$	$v_4(NNP)$	$v_5(NNP)$
VB		$v_1(VB)$	$v_2(VB)$	$v_3(VB)$	$v_4(MD)$	$v_5(MD)$
NN		$v_1(NN)$	$v_2(NN)$	$v_3(NN)$	$v_4(NN)$	$v_5(NN)$
MD		$v_1(MD)$	$v_2(MD)$	$v_3(MD)$	$v_4(MD)$	$v_5(MD)$
START						
		Janet	will	back	the	bill

END							v_T(END)
DT		v_1(DT)	v_2(DT)	v_3(DT)	v_4(DT)	v_5(DT)	
NNP		v_1(NNP)	v_2(NNP)	v_3(NNP)	v_4(NNP)	v_5(NNP)	
VB		v_1(VB)	v_2(VB)	v_3(VB)	v_4(MD)	v_5(MD)	
NN		v_1(NN)	v_2(NN)	v_3(NN)	v_4(NN)	v_5(NN)	
MD		v_1(MD)	v_2(MD)	v_3(MD)	v_4(MD)	v_5(MD)	
START							
		Janet	will	back	the	bill	

v_T(END) encodes the best path through the entire sequence



For each timestep t + label, keep track of the max element from t-1 to reconstruct best path

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*

create a path probability matrix *viterbi[N+2,T]*
for each state s **from** 1 **to** N **do** ; initialization step
$viterbi[s,1] \leftarrow a_{0,s} * b_s(o_1)$
$backpointer[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ; recursion step
for each state s **from** 1 **to** N **do**

$$viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$$

$$backpointer[s,t] \leftarrow \operatorname*{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s}$$

$$viterbi[q_F,T] \leftarrow \max_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$$; termination step

$$backpointer[q_F,T] \leftarrow \operatorname*{argmax}_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$$; termination step

return the backtrace path by following backpointers to states back in time from $backpointer[q_F,T]$

Figure 10.8 Viterbi algorithm for finding optimal sequence of tags. Given an observation sequence and an HMM $\lambda = (A,B)$, the algorithm returns the state path through the HMM that assigns maximum likelihood to the observation sequence. Note that states 0 and q_F are non-emitting.

Logistics

- Exam1 is being graded and reviewed.
- No homework this week
    - Homework 4 will be released towards end of the week.
- AP1 is due this Sunday March 3.
- Quiz 4 will be out this Friday afternoon (Due Monday night).
- Next time: Neural Sequence Models