CS388: Natural Language Processing

Lecture 4: HMMs, POS

Greg Durrett

TEXAS
The University of Texas at Austin

Parts of this lecture adapted from Dan Klein, UC Berkeley
and Vivek Srikumar, University of Utah

Administrivia

- Mini 1 due today at 11:59pm
  - Shuffling: online methods are sensitive to dataset order, shuffling helps!
- Project 1 out today
  - Viterbi algorithm, CRF NER system, extension
  - This class will cover what you need to get started on it, the next class will cover everything you need to complete it

Recall: Multiclass Classification

- Two views of multiclass classification:
  - Different features: $\text{argmax}_{y \in \mathcal{Y}} w^\top f(x, y)$
  - Different weights: $\text{argmax}_{y \in \mathcal{Y}} w_y^\top f(x)$
- "Different features" (most relevant for us in the next week):

$f(x, y =$ Health $) =$ [1, 1, 0, 0, 0, 0, 0, 0, 0]   I[contains *drug* & label = Health]

$f(x, y =$ Sports $) =$ [0, 0, 0, 1, 1, 0, 0, 0, 0]

  - Equivalent to having three weight vectors stapled together

Recall: Multiclass Classification

- Logistic regression: $P(y|x) = \frac{\exp\left(w^\top f(x, y)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w^\top f(x, y')\right)}$

Gradient of log likelihood:

$$f(x, y^*) - \mathbb{E}_y[f(x, y)] = f(x, y^*) - \sum_y [P_w(y|x) f(x, y)]$$

"towards gold feature value, away from expectation of feature value"

This Lecture

- Part-of-speech tagging
- Hidden Markov Models
- HMM parameter estimation
- Viterbi algorithm
- State-of-the-art in POS tagging

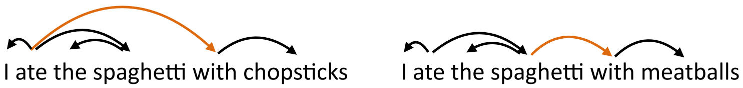
Where are we in the course?

- This lecture + next lecture: sequence modeling. Think about structured sequence representations of language
- Afterwards: neural networks. Revisit machine learning methods for the structures we've already seen (mostly classification)
- Then: trees: syntax and semantics. Back to thinking about structure

POS Tagging

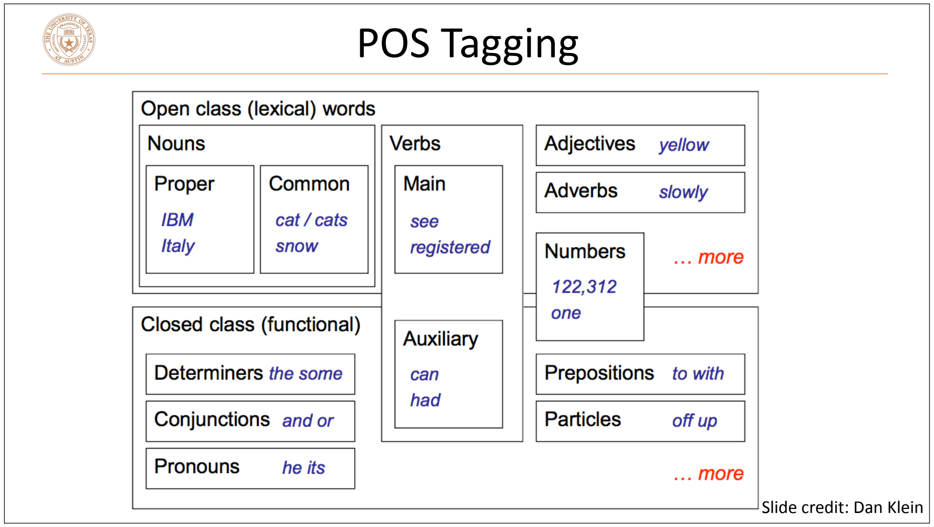
Linguistic Structures

- Language has hierarchical structure, can represent with trees


I ate the spaghetti with chopsticks
I ate the spaghetti with meatballs

- Understanding syntax fundamentally requires trees — the sentences have the same shallow analysis. But the first step we'll take towards understanding this is understanding **parts of speech**

NN   NNS   VBZ NNS
Teacher strikes idle kids



POS Tagging

Open class (lexical) words

Nouns

Proper *IBM Italy*

Common *cat / cats snow*

Verbs

Main *see registered*

Adjectives *yellow*

Adverbs *slowly*

Numbers *122,312 one*

… more

Closed class (functional)

Determiners *the some*

Conjunctions *and or*

Pronouns *he its*

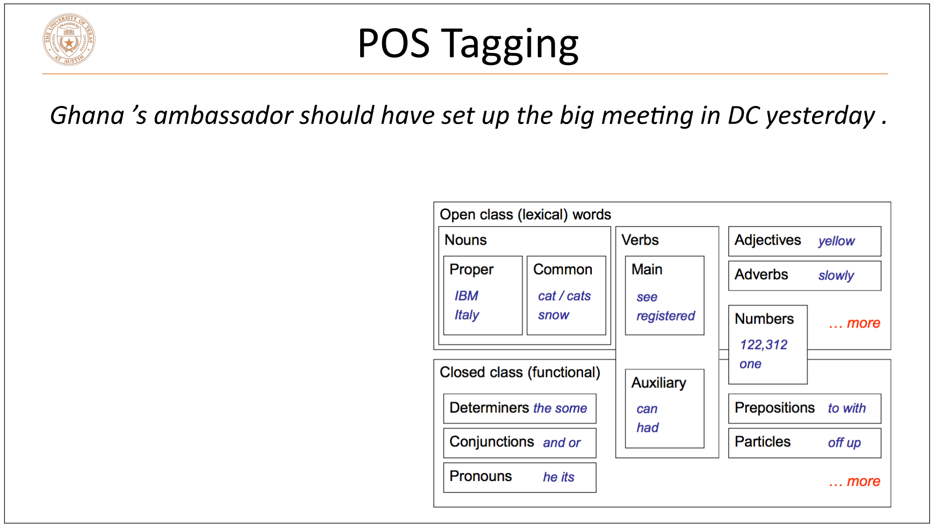
Auxiliary *can had*

Prepositions *to with*

Particles *off up*

… more

Slide credit: Dan Klein



POS Tagging

Ghana 's ambassador should have set up the big meeting in DC yesterday .

Open class (lexical) words

Nouns

Proper *IBM Italy*

Common *cat / cats snow*

Verbs

Main *see registered*

Adjectives *yellow*

Adverbs *slowly*

Numbers *122,312 one*

… more

Closed class (functional)

Determiners *the some*

Conjunctions *and or*

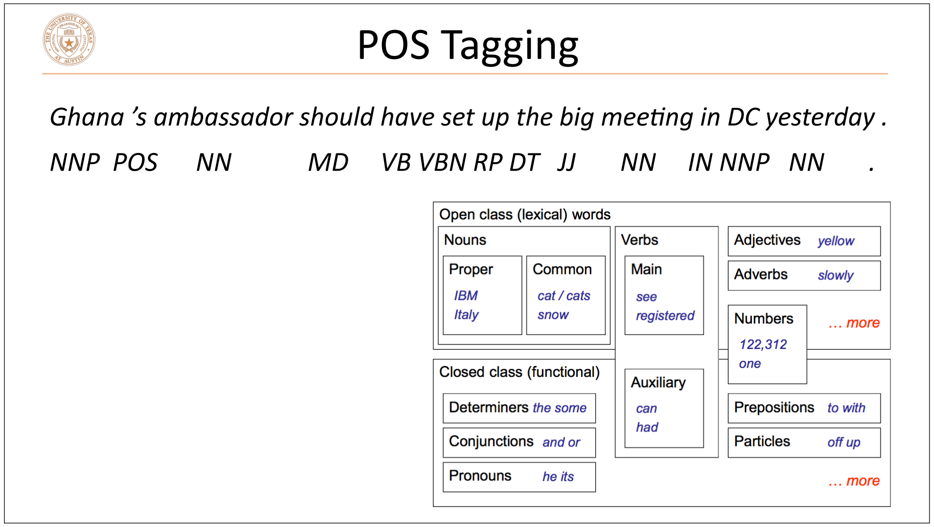
Pronouns *he its*

Auxiliary *can had*

Prepositions *to with*

Particles *off up*

… more



POS Tagging

Ghana 's ambassador should have set up the big meeting in DC yesterday .

NNP POS NN MD VB VBN RP DT JJ NN IN NNP NN .

Open class (lexical) words

Nouns

Proper *IBM Italy*

Common *cat / cats snow*

Verbs

Main *see registered*

Adjectives *yellow*

Adverbs *slowly*

Numbers *122,312 one*

… more

Closed class (functional)

Determiners *the some*

Conjunctions *and or*

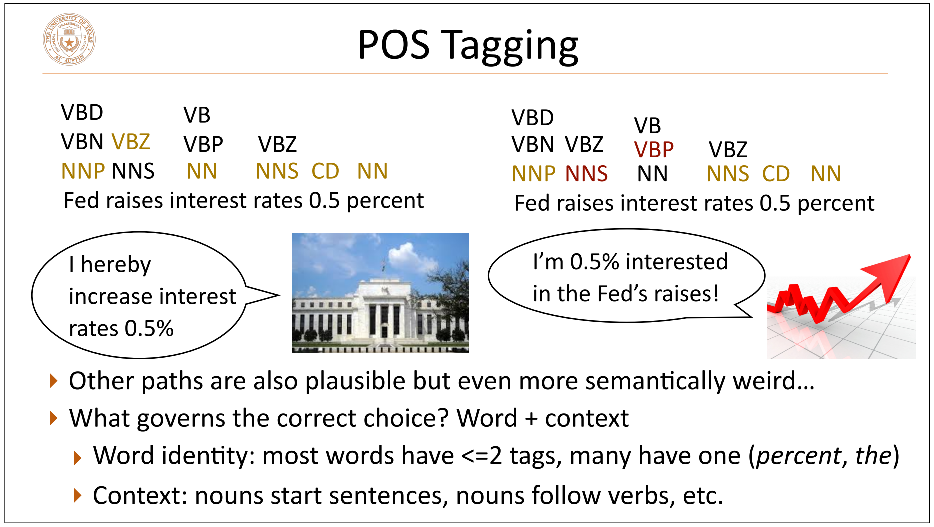
Pronouns *he its*

Auxiliary *can had*

Prepositions *to with*

Particles *off up*

… more



POS Tagging

VBD VB
VBN VBZ VBP VBZ
NNP NNS NN NNS CD NN
Fed raises interest rates 0.5 percent

VBD VB
VBN VBZ VBP VBZ
NNP NNS NN NNS CD NN
Fed raises interest rates 0.5 percent

I hereby increase interest rates 0.5%

I'm 0.5% interested in the Fed's raises!

- Other paths are also plausible but even more semantically weird…
- What governs the correct choice? Word + context
  - Word identity: most words have <=2 tags, many have one (*percent*, *the*)
  - Context: nouns start sentences, nouns follow verbs, etc.

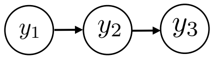
What is this good for?

- Text-to-speech: record, lead
- Preprocessing step for syntactic parsers or other tasks
- (Very) shallow information extraction

Hidden Markov Models

Hidden Markov Models

- Input $\mathbf{x} = (x_1, ..., x_n)$ Output $\mathbf{y} = (y_1, ..., y_n)$
- Model the sequence of tags ***y*** over words ***x*** as a Markov process
- Markov property: future is conditionally independent of the past given the present



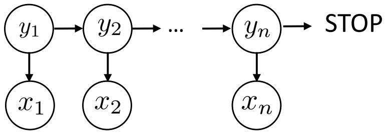
$$P(y_3|y_1, y_2) = P(y_3|y_2)$$

- If ***y*** are tags, this roughly corresponds to assuming that the next tag only depends on the current tag, not anything before

Hidden Markov Models

- Input $\mathbf{x} = (x_1, ..., x_n)$ Output $\mathbf{y} = (y_1, ..., y_n)$

y ∈ T = set of possible tags (including STOP);
x ∈ V = vocab of words

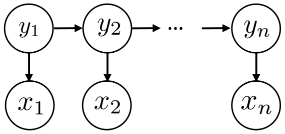


$$P(\mathbf{y}, \mathbf{x}) = \underbrace{P(y_1)}_{\text{Initial distribution}} \underbrace{\prod_{i=2}^{n} P(y_i|y_{i-1})}_{\text{Transition probabilities}} \underbrace{\prod_{i=1}^{n} P(x_i|y_i)}_{\text{Emission probabilities}}$$

- Observation (*x*) depends only on current state (*y*)

HMMs: Parameters

- Input $\mathbf{x} = (x_1, ..., x_n)$ Output $\mathbf{y} = (y_1, ..., y_n)$



$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^{n} P(y_i | y_{i-1}) \prod_{i=1}^{n} P(x_i | y_i)$$

- Initial distribution: |T| x 1 vector (distribution over initial states)
- Emission distribution: |T| x |V| matrix (distribution over words per tag)
- Transition distribution: |T| x |T| matrix (distribution over next tags per tag)

HMMs Example

Tags = {N , V, STOP} Vocabulary = {they, can, fish}

Initial

y_1	
N	1.0
V	0
STOP	0

Transition

y_{i-1} \ y_i	N	V	STOP
N	1/5	3/5	1/5
V	1/5	1/5	3/5

Emission

y_i \ x_i	they	can	fish
N	1	0	0
V	0	1/2	1/2

Transitions in POS Tagging

VBD		VB			
VBN	VBZ	VBP	VBZ		
NNP	NNS	NN	NNS	CD	NN
Fed	raises	interest	rates	0.5	percent

- $P(y_1 = \mathrm{NNP})$ likely because start of sentence
- $P(y_2 = \mathrm{VBZ} | y_1 = \mathrm{NNP})$ likely because verb often follows noun
- $P(y_3 = \mathrm{NN} | y_2 = \mathrm{VBZ})$: direct object can follow verb

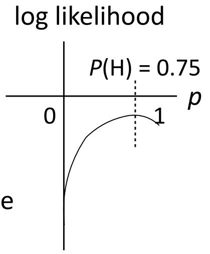
- How are these probabilities learned?

Learning HMMs

Maximum Likelihood Estimation

- Imagine a coin flip which is heads with probability *p*
- Observe (H, H, H, T) and maximize likelihood: $\prod_{j=1}^{m} P(y_j) = p^3(1-p)$
- Equivalent to maximizing *log* likelihood

$$\sum_{j=1}^{m} \log P(y_j) = 3 \log p + \log(1-p)$$



- Maximum likelihood parameters for binomial/ multinomial = read counts off of the data + normalize

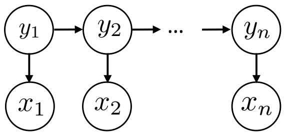
Training HMMs

- Transitions
  - Count up all pairs (y_i, y_{i+1}) in the training data
  - Count up occurrences of what tag *T* can transition to
  - Normalize to get a distribution for P(next tag | *T*)
  - Need to *smooth* this distribution, won't discuss here
- Emissions: similar scheme, but trickier smoothing!

Inference: Viterbi Algorithm

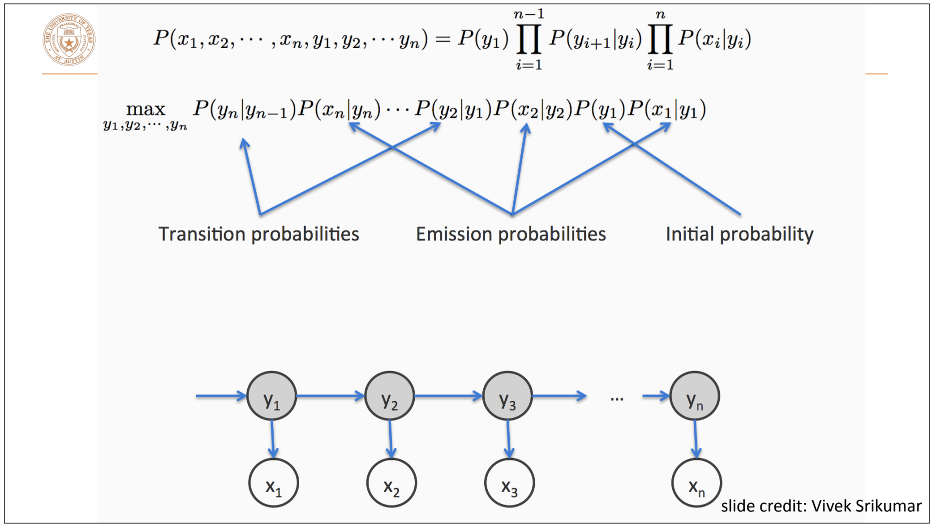
Inference in HMMs

- Input $\mathbf{x} = (x_1, ..., x_n)$ Output $\mathbf{y} = (y_1, ..., y_n)$



$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^{n} \boxed{P(y_i|y_{i-1})} \prod_{i=1}^{n} P(x_i|y_i)$$

- Inference problem: $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \text{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{\cancel{P(\mathbf{x})}}$
- Exponentially many possible ***y*** here!
- Solution: dynamic programming (possible because of Markov structure!)



$$P(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^{n} P(x_i|y_i)$$

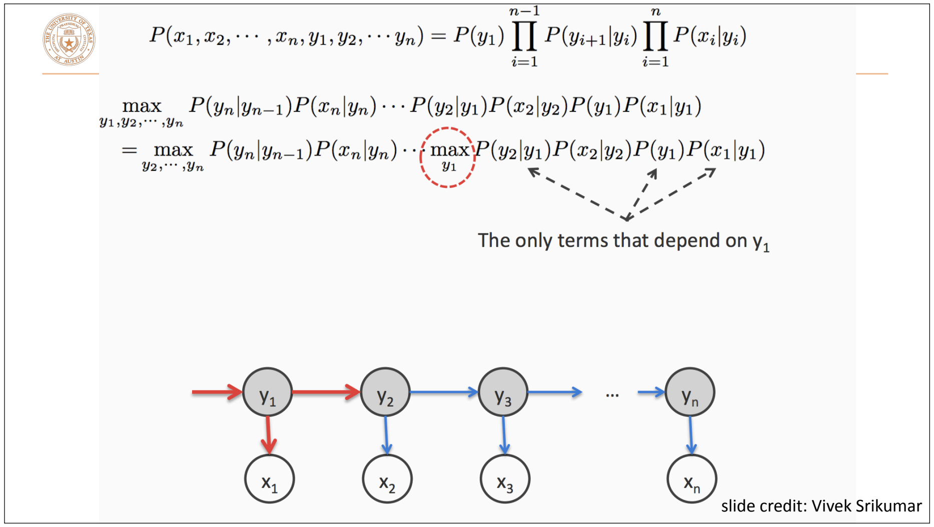
$$\max_{y_1, y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots P(y_2|y_1) P(x_2|y_2) P(y_1) P(x_1|y_1)$$

Transition probabilities

Emission probabilities

Initial probability

slide credit: Vivek Srikumar



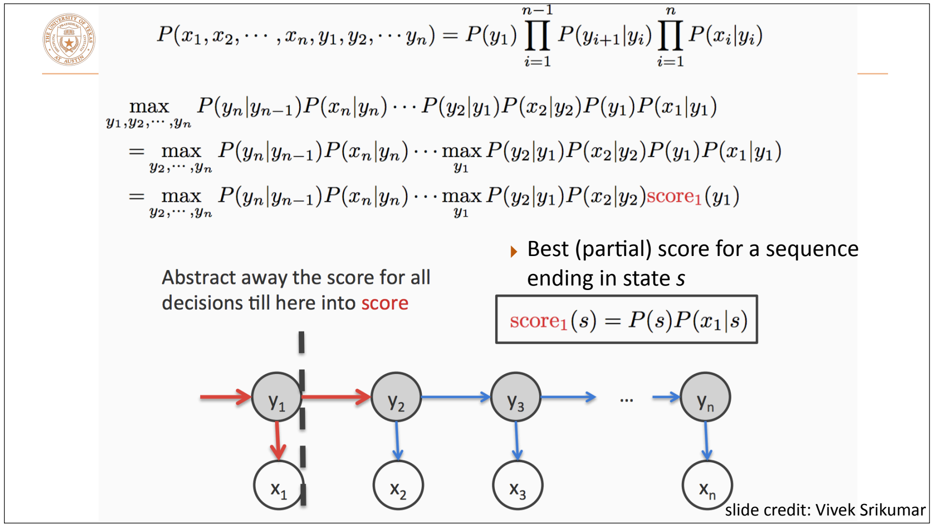
$$P(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^{n} P(x_i|y_i)$$

$$\max_{y_1, y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots P(y_2|y_1) P(x_2|y_2) P(y_1) P(x_1|y_1)$$

$$= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1) P(x_2|y_2) P(y_1) P(x_1|y_1)$$

The only terms that depend on y_1

slide credit: Vivek Srikumar



$$P(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^{n} P(x_i|y_i)$$

$$\max_{y_1, y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots P(y_2|y_1) P(x_2|y_2) P(y_1) P(x_1|y_1)$$

$$= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1) P(x_2|y_2) P(y_1) P(x_1|y_1)$$

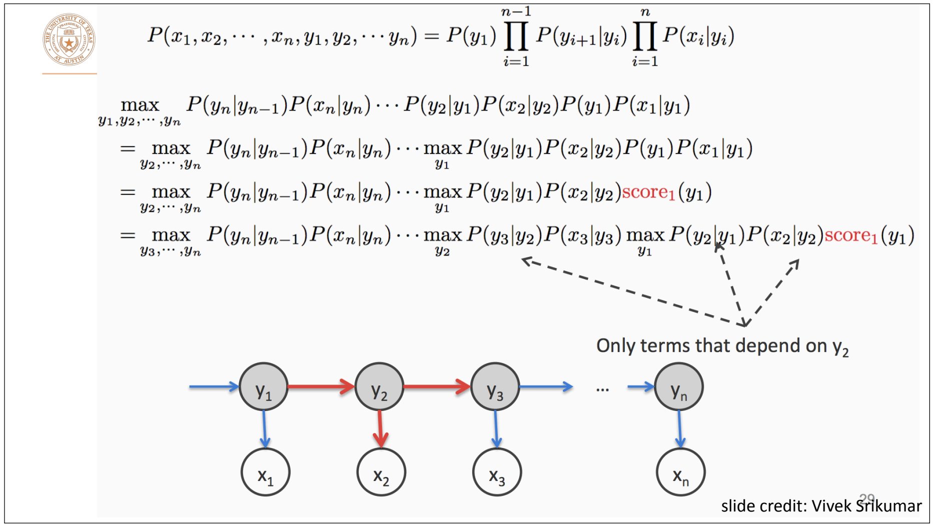
$$= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1) P(x_2|y_2) \text{score}_1(y_1)$$

Abstract away the score for all decisions till here into score

- Best (partial) score for a sequence ending in state *s*

$$\boxed{\text{score}_1(s) = P(s) P(x_1|s)}$$

slide credit: Vivek Srikumar



$$P(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^{n} P(x_i|y_i)$$

$$\max_{y_1, y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots P(y_2|y_1) P(x_2|y_2) P(y_1) P(x_1|y_1)$$

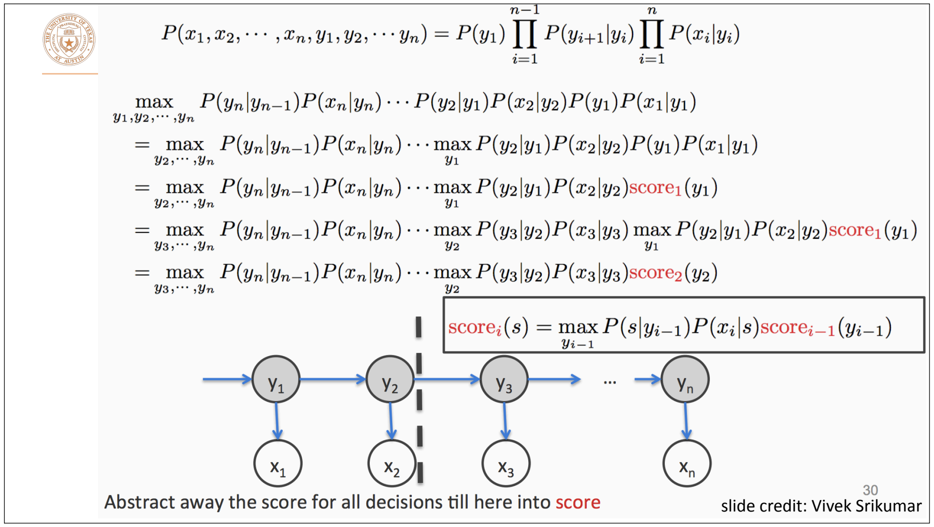
$$= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1) P(x_2|y_2) P(y_1) P(x_1|y_1)$$

$$= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1) P(x_2|y_2) \text{score}_1(y_1)$$

$$= \max_{y_3, \cdots, y_n} P(y_n|y_{n-1}) P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2) P(x_3|y_3) \max_{y_1} P(y_2|y_1) P(x_2|y_2) \text{score}_1(y_1)$$

Only terms that depend on y_2

slide credit: Vivek Srikumar



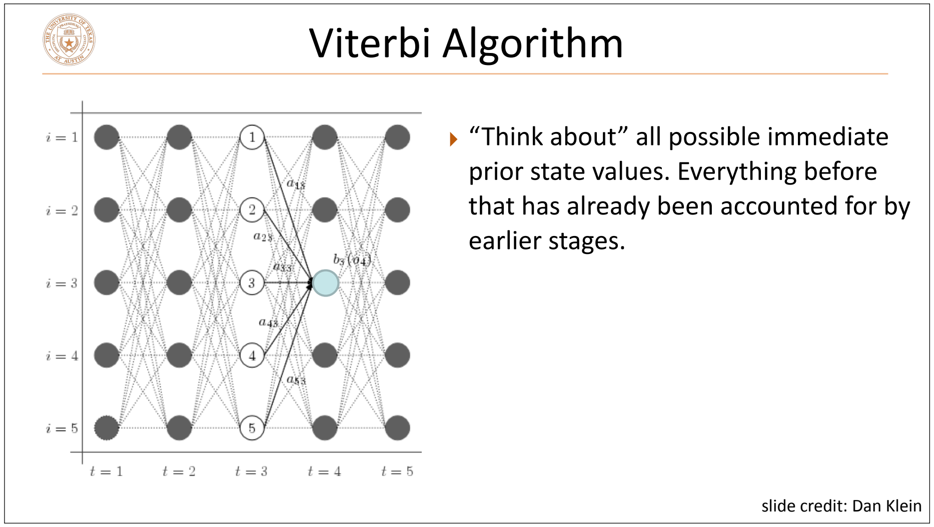
$$P(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^{n} P(x_i|y_i)$$

$$\begin{aligned} &\max_{y_1, y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3) \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3)\text{score}_2(y_2) \end{aligned}$$

$$\boxed{\text{score}_i(s) = \max_{y_{i-1}} P(s|y_{i-1})P(x_i|s)\text{score}_{i-1}(y_{i-1})}$$

Abstract away the score for all decisions till here into score

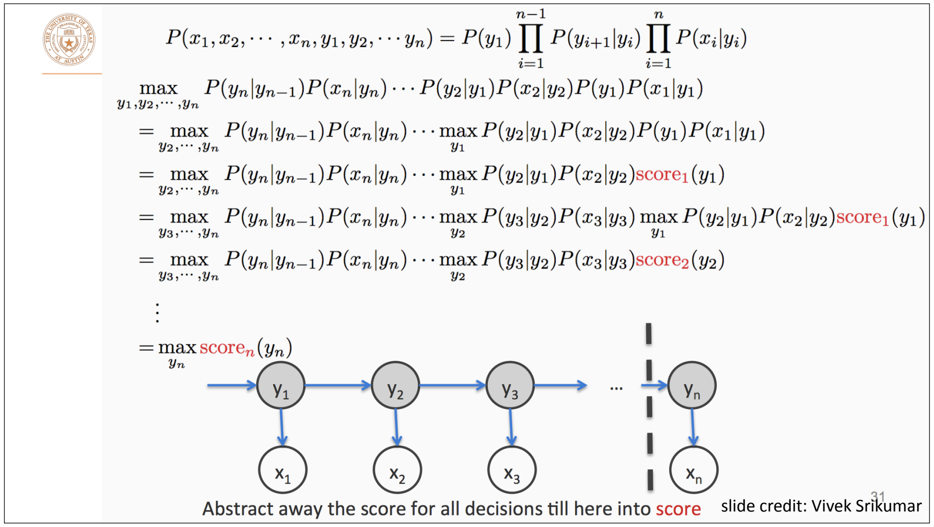
slide credit: Vivek Srikumar



Viterbi Algorithm

- "Think about" all possible immediate prior state values. Everything before that has already been accounted for by earlier stages.

slide credit: Dan Klein

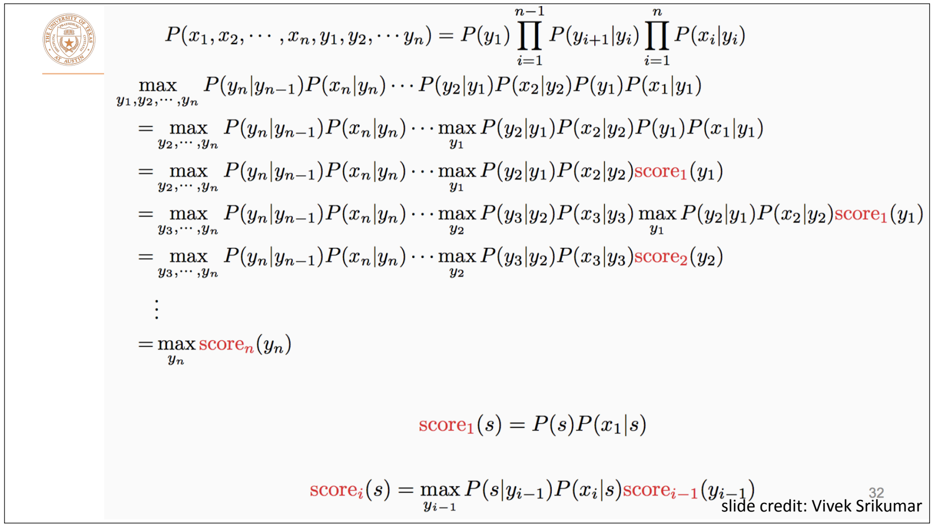


$$P(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^{n} P(x_i|y_i)$$

$$\begin{aligned} &\max_{y_1, y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3) \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3)\text{score}_2(y_2) \\ &\vdots \\ &= \max_{y_n} \text{score}_n(y_n) \end{aligned}$$

Abstract away the score for all decisions till here into score

slide credit: Vivek Srikumar



$$P(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^{n} P(x_i|y_i)$$

$$\begin{aligned} &\max_{y_1, y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3) \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \cdots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3)\text{score}_2(y_2) \\ &\vdots \\ &= \max_{y_n} \text{score}_n(y_n) \end{aligned}$$

$$\text{score}_1(s) = P(s)P(x_1|s)$$

$$\text{score}_i(s) = \max_{y_{i-1}} P(s|y_{i-1})P(x_i|s)\text{score}_{i-1}(y_{i-1})$$

slide credit: Vivek Srikumar

1. Initial: For each state s, calculate

$$\text{score}_1(s) = P(s)P(x_1|s) = \pi_s B_{x_1,s}$$

2. Recurrence: For i = 2 to n, for every state s, calculate

$$\text{score}_i(s) = \max_{y_{i-1}} P(s|y_{i-1})P(x_i|s)\text{score}_{i-1}(y_{i-1})$$
$$= \max_{y_{i-1}} A_{y_{i-1},s} B_{s,x_i} \text{score}_{i-1}(y_{i-1})$$

3. Final state: calculate

$$\max_{\mathbf{y}} P(\mathbf{y}, \mathbf{x}|\pi, A, B) = \max_{s} \text{score}_n(s)$$

π: Initial probabilities
A: Transitions
B: Emissions

This only calculates the max. To get final answer (*argmax*),

- keep track of which state corresponds to the max at each step
- build the answer using these back pointers

slide credit: Vivek Srikumar

POS Taggers

HMM POS Tagging

- Baseline: assign each word its most frequent tag: ~90% accuracy
- Trigram HMM: ~95% accuracy / 55% on unknown words

Slide credit: Dan Klein

Trigram Taggers

NNP VBZ NN NNS CD NN
Fed raises interest rates 0.5 percent

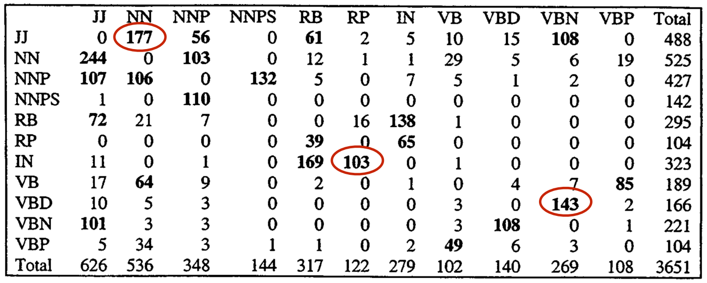
- Trigram model: y_1 = (<S>, NNP), y_2 = (NNP, VBZ), …
- P((VBZ, NN) | (NNP, VBZ)) — more context! Noun-verb-noun S-V-O
- Tradeoff between model capacity and data size — trigrams are a "sweet spot" for POS tagging

HMM POS Tagging

- Baseline: assign each word its most frequent tag: ~90% accuracy
- Trigram HMM: ~95% accuracy / 55% on unknown words
- TnT tagger (Brants 1998, tuned HMM): 96.2% accuracy / 86.0% on unks
- State-of-the-art (BiLSTM-CRFs): 97.5% / 89%+

Slide credit: Dan Klein

Errors



	JJ	NN	NNP	NNPS	RB	RP	IN	VB	VBD	VBN	VBP	Total
JJ	0	**177**	**56**	0	**61**	2	5	10	15	**108**	0	488
NN	**244**	0	**103**	0	12	1	1	29	5	6	19	525
NNP	**107**	**106**	0	**132**	5	0	7	5	1	2	0	427
NNPS	1	0	**110**	0	0	0	0	0	0	0	0	142
RB	**72**	21	7	0	0	16	**138**	1	0	0	0	295
RP	0	0	0	0	**39**	0	**65**	0	0	0	0	104
IN	11	0	1	0	**169**	**103**	0	1	0	0	0	323
VB	17	**64**	9	0	2	0	1	0	4	7	**85**	189
VBD	10	5	3	0	0	0	0	3	0	**143**	2	166
VBN	**101**	3	3	0	0	0	0	3	**108**	0	1	221
VBP	5	34	3	1	1	0	2	**49**	6	3	0	104
Total	626	536	348	144	317	122	279	102	140	269	108	3651

JJ/NN NN
official knowledge

VBD RP/IN DT NN
made up the story

RB VBD/VBN NNS
recently sold shares

(NN NN: tax cut, art gallery, …)

Slide credit: Dan Klein / Toutanova + Manning (2000)

Remaining Errors

- Lexicon gap (word not seen with that tag in training) 4.5%
- Unknown word: 4.5%
- Could get right: 16% (many of these involve parsing!)
- Difficult linguistics: 20%

VBD / VBP? (past or present?)

They set up absurd situations, detached from reality

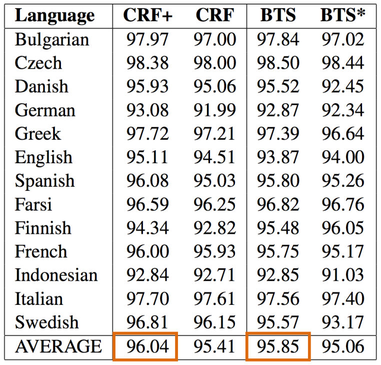
- Underspecified / unclear, gold standard inconsistent / wrong: **58%**

adjective or verbal participle? JJ / VBN?

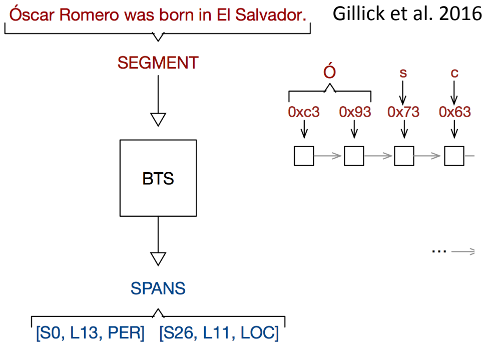
a $ 10 million fourth-quarter charge against discontinued operations

Manning 2011 "Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?"

Other Languages

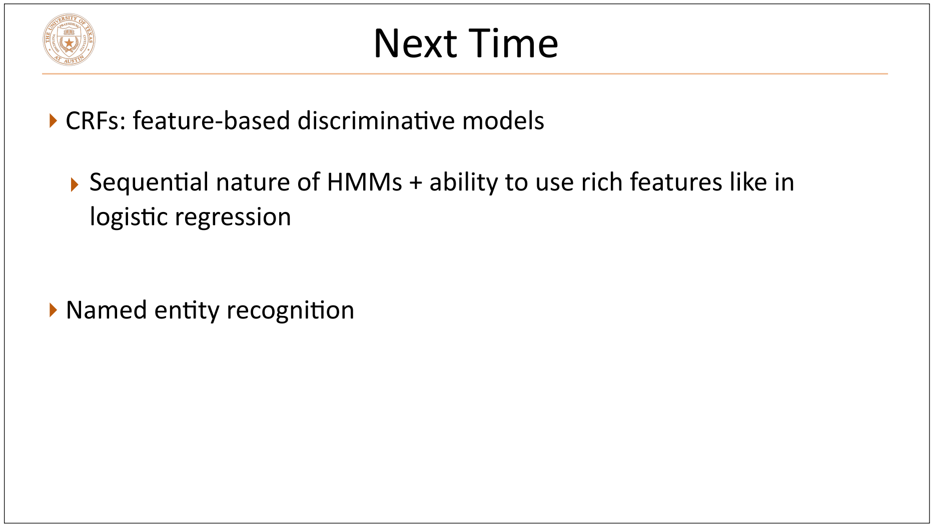


Language	CRF+	CRF	BTS	BTS*
Bulgarian	97.97	97.00	97.84	97.02
Czech	98.38	98.00	98.50	98.44
Danish	95.93	95.06	95.52	92.45
German	93.08	91.99	92.87	92.34
Greek	97.72	97.21	97.39	96.64
English	95.11	94.51	93.87	94.00
Spanish	96.08	95.03	95.80	95.26
Farsi	96.59	96.25	96.82	96.76
Finnish	94.34	92.82	95.48	96.05
French	96.00	95.93	95.75	95.17
Indonesian	92.84	92.71	92.85	91.03
Italian	97.70	97.61	97.56	97.40
Swedish	96.81	96.15	95.57	93.17
AVERAGE	96.04	95.41	95.85	95.06



Gillick et al. 2016

- Universal POS tagset (~12 tags), cross-lingual model works as well as tuned CRF using external resources



Next Time

- CRFs: feature-based discriminative models
  - Sequential nature of HMMs + ability to use rich features like in logistic regression
- Named entity recognition