POS Tagging

HMM POS Tagging

- Penn Treebank English POS tagging (see homework): 44 tags
- Baseline: assign each word its most frequent tag: ~90% accuracy
- Trigram HMM (model pairs of tags): ~95% accuracy / 55% on words not seen in train
- TnT tagger (Brants 1998, tuned HMM): 96.2% acc / 86.0% on unks
- CRF tagger (Toutanova + Manning 2000): 96.9% / 87.0%
- State-of-the-art (BiLSTM-CRFs, BERT): 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% of errors
- 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?”

CRFs and NER

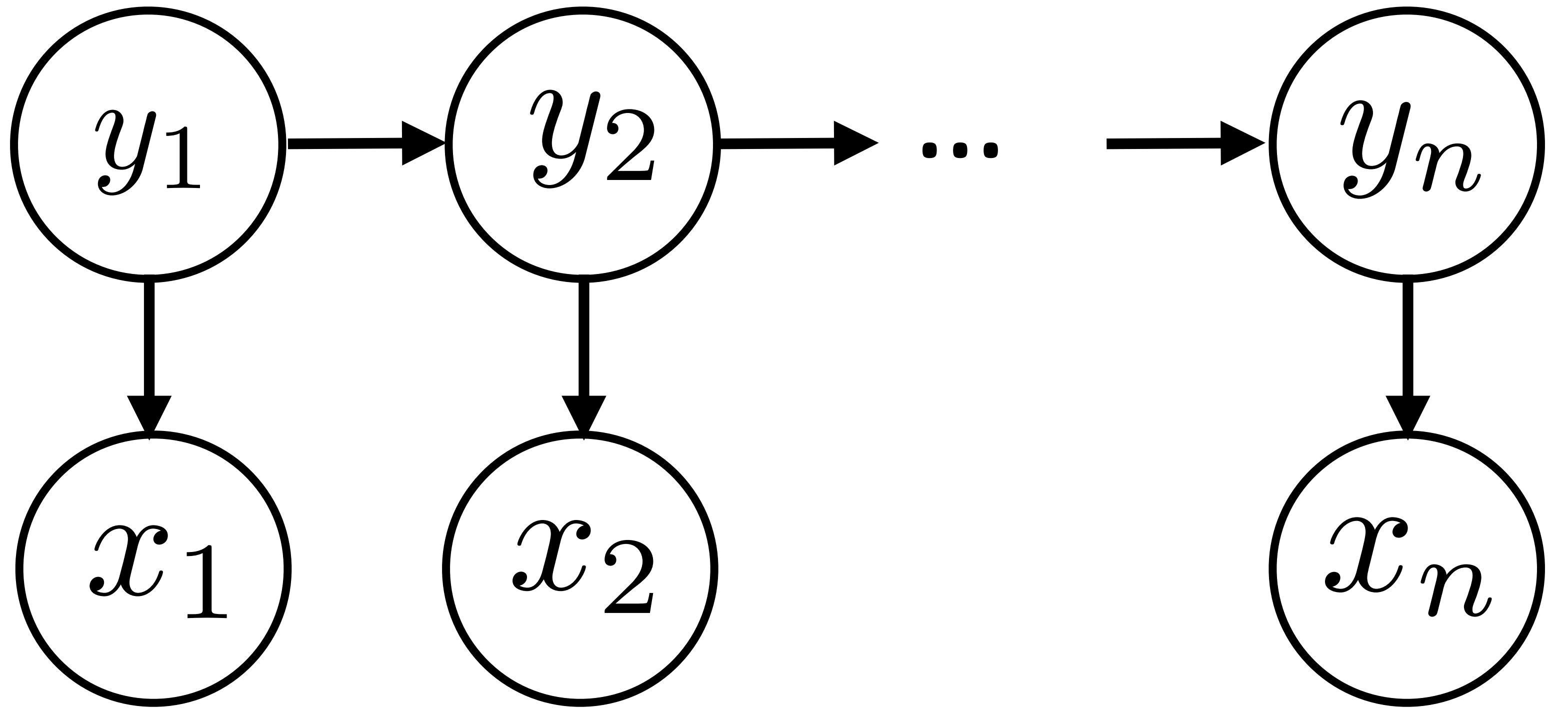
Named Entity Recognition

B-PER	I-PER	O	O	O	B-LOC	O	O	O	B-ORG	O	O
Barack	*Obama*	*will*	*travel*	*to*	*Hangzhou*	*today*	*for*	*the*	*G20*	*meeting*	*.*
PERSON					LOC				ORG		

- Frame as a sequence problem with a BIO tagset: begin, inside, outside
- Why might an HMM not do so well here?
  - Lots of O's, so tags aren't as informative about context
  - Want to use context features (*to Hangzhou* => *Hangzhou* is a LOC)
- Conditional random fields (CRFs) can help solve these problems

HMMs

- Big advantage: transitions, scoring pairs of adjacent y's



- Big downside: not able to incorporate useful word context information
- Solution: switch from generative to discriminative model (conditional random fields) so we can condition on the *entire input*.
- Conditional random fields: logistic regression + features on pairs of y's

Tagging with Logistic Regression

- Logistic regression over each tag individually:

"different features" approach to features for a single tag

$$P(y_i = y|\mathbf{x}, i) = \frac{\exp(\mathbf{w}^\top \mathbf{f}(y, i, \mathbf{x}))}{\sum_{y' \in \mathcal{Y}} \exp(\mathbf{w}^\top \mathbf{f}(y', i, \mathbf{x}))}$$

- Over all tags:

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \prod_{i=1}^{n} P(y_i = \tilde{y}_i|\mathbf{x}, i) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(\tilde{y}_i, i, \mathbf{x})\right)$$

- Score of a prediction: sum of weights dot features over each individual predicted tag (this is a simple CRF but not the general form)

- Set *Z* equal to the product of denominators; we'll discuss this in a few slides

Example

B-PER I-PER O O

Barack Obama will travel

feats = $\mathbf{f}_e$(B-PER, i=1, $\mathbf{x}$) + $\mathbf{f}_e$(I-PER, i=2, $\mathbf{x}$) + $\mathbf{f}_e$(O, i=3, $\mathbf{x}$) + $\mathbf{f}_e$(O, i=4, $\mathbf{x}$)

[CurrWord=*Obama* & label=I-PER, PrevWord=*Barack* & label=I-PER, CurrWordIsCapitalized & label=I-PER, ...]

B-PER B-PER O O

Barack Obama will travel

feats = $\mathbf{f}_e$(B-PER, i=1, $\mathbf{x}$) + $\mathbf{f}_e$(B-PER, i=2, $\mathbf{x}$) + $\mathbf{f}_e$(O, i=3, $\mathbf{x}$) + $\mathbf{f}_e$(O, i=4, $\mathbf{x}$)

Adding Structure

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(\tilde{y}_i, i, \mathbf{x})\right)$$

- We want to be able to learn that some tags don't follow other tags — want to have features on tag *pairs*

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_e(\tilde{y}_i, i, \mathbf{x}) + \sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_t(\tilde{y}_i, \tilde{y}_{i+1}, i, \mathbf{x})\right)$$

- Score: sum of weights dot $\mathbf{f}_e$ features over each predicted tag ("emissions") plus sum of weights dot $\mathbf{f}_t$ features over tag pairs ("transitions")
- This is a sequential CRF

Example

B-PER I-PER O O

Barack Obama will travel

feats = $\mathbf{f}_e$(B-PER, i=1, $\mathbf{x}$) + $\mathbf{f}_e$(I-PER, i=2, $\mathbf{x}$) + $\mathbf{f}_e$(O, i=3, $\mathbf{x}$) + $\mathbf{f}_e$(O, i=4, $\mathbf{x}$)
+ $\mathbf{f}_t$(B-PER, I-PER, i=1, $\mathbf{x}$) + $\mathbf{f}_t$(I-PER, O, i=2, $\mathbf{x}$) + $\mathbf{f}_t$(O, O, i=3, $\mathbf{x}$)

B-PER B-PER O O

Barack Obama will travel

feats = $\mathbf{f}_e$(B-PER, i=1, $\mathbf{x}$) + $\mathbf{f}_e$(B-PER, i=2, $\mathbf{x}$) + $\mathbf{f}_e$(O, i=3, $\mathbf{x}$) + $\mathbf{f}_e$(O, i=4, $\mathbf{x}$)
+ $\mathbf{f}_t$(B-PER, B-PER, i=1, $\mathbf{x}$) + $\mathbf{f}_t$(B-PER, O, i=2, $\mathbf{x}$) + $\mathbf{f}_t$(O, O, i=3, $\mathbf{x}$)

- *Obama* can start a new named entity (emission feats look okay), but we're not likely to have two PER entities in a row (transition feats)

Features for NER

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_e(\tilde{y}_i, i, \mathbf{x}) + \sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_t(\tilde{y}_i, \tilde{y}_{i+1}, i, \mathbf{x})\right)$$

O B-LOC

Barack Obama will travel to Hangzhou today for the G20 meeting .

Transitions: $\mathbf{f}_t(\text{O}, \text{B-LOC}, i = 5, \mathbf{x})$ = Indicator[O — B-LOC]

Emissions: $\mathbf{f}_e(\text{B-LOC}, i = 6, \mathbf{x})$ = Indicator[B-LOC & Curr word = *Hangzhou*]

Indicator[B-LOC & Prev word = *to*]

- We couldn't use a "previous word" feature in the HMM at all!

Conditional Random Fields

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_e(\tilde{y}_i, i, \mathbf{x}) + \sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_t(\tilde{y}_i, \tilde{y}_{i+1}, i, \mathbf{x})\right)$$

normalizer *Z*: must make this a probability distribution over all possible seqs

$$Z = \sum_{\mathbf{y}' \in \mathcal{Y}^n} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_e(y'_i, i, \mathbf{x}) + \sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_t(y'_i, y'_{i+1}, i, \mathbf{x})\right)$$

Inference and Learning

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_e(\tilde{y}_i, i, \mathbf{x}) + \sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_t(\tilde{y}_i, \tilde{y}_{i+1}, i, \mathbf{x})\right)$$

- Inference: Can use the Viterbi algorithm to find the highest scoring path. Replace HMM log probs with "scores" from weights dot features

$$\log P(x_i|y_i) \rightarrow \mathbf{w}^\top \mathbf{f}_e(y_i, i, \mathbf{x})$$

$$\log P(y_i|y_{i-1}) \rightarrow \mathbf{w}^\top \mathbf{f}_t(y_{i-1}, y_i, i, \mathbf{x})$$

(initial distribution is removed)

- Learning: requires running *forward-backward* (like Viterbi but with summing instead of maxing over *y*'s) to compute *Z*, then doing some tricky math to compute gradients [outside scope of the course/not on midterm]

Takeaways

- CRFs provide a way to build structured feature-based models: logistic regression over structured objects like sequences

- Inference and learning can still be done efficiently but require dynamic programming

- CRFs don't have to be linear models; can use scores derived from neural networks ("neural CRFs")