CRF model

$$\underbrace{P(\overset{\text{Tagging}}{y} \mid \overset{\text{Text}}{x})}\propto \exp(\underbrace{\theta^{\mathsf{T}} f(x, y)})$$

Params — Feature Vector

Tagging — Text

- The idea:
  - Global scoring function for output y
    G(y) = θ'f(x,y) $= \sum_{j=1}^{J} \theta_j f_j(x,y)$
  - Global features from sum of local features
  - Scoring function decomposes into local parts (therefore can use Viterbi)
    To predict y from x, θ
- Advantages vs. HMM
  - Can add arbitrary observation features!
    - Character ngrams
    - Capitalization
    - Word presence in dictionary...
  - Discriminative learning: we use the ***perceptron***
    Learn θ from gold (x,y)

Global $f(x,y) = \sum_t f(y_{t-1}, y_t, x_t)$

Score $G(y) = \theta' f(x,y)$

$$= \theta^T \left(\sum_t f(y_{t-1}, y_t, x_t) \right)$$

$$= \sum_t \underbrace{\theta^T f(y_{t-1}, y_t, x_t)}_{g(y_{t-1}, y_t)}$$

$$= \sum_t A(y_{t-1}, y_t) + B_t(y_t)$$

$$A(y_{t-1}, y_t) \downarrow \theta_{y_{t-1}, y_t} \qquad B_t(y_t) \downarrow \theta_{obs: y_t, x_t}$$

$$A(\text{Noun}, \text{Verb}) = \theta_{\text{Noun}, \text{Verb}}$$

Transition Feature

$$f_3(y_{t-1}, y_t, x_t) = \begin{cases} 1 & \text{if } y_{t-1} = V \text{ and } \\ & \quad y_t = V \\ 0 & \text{otherwise} \end{cases}$$

obs. Feat

$$f_{1500}(y_{t-1}, y_t, x_t) = \begin{cases} 1 & \text{if } y_t = V \text{ and } \\ & \quad x_t = \text{get} \\ 0 & \text{otherwise} \end{cases}$$

CRF is from local features

$x =$	finna	get	good
$y =$ y_0 = [START]	*V*	*V*	*A*

$f(y_{t-1}, y_t, x_{t-2} .. x_{t+2})$

S, V   V, finna

local $f(y_{t-1}, y_t, x_t) =$ [0 0 1 0 0 . . . . . 1 . . .]

$f(S, V, \text{finna})$

binary feats

V, V

$f(V, V, \text{get}) =$ [1 . . . 1]) +

) +

V, V   V, get

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

$G(y) = \theta' f(x,y)$

Count of V=>V transitions in y

$$= \sum_t f(y_{t-1}, y_t, x_t)$$

	finna	get	good
gold y =	V	V	A

	Transition features							Observation features																
θ	-0.6	-1.0	1.1	0.5	0.0	0.8	0.5	-1.3	-1.6	0.0	0.6	0.0	-0.2	-0.2	0.8	-1.0	0.1	-1.9	1.1	1.2	-0.1	-1.0	-0.1	-0.1
$f(x,y)$	1	0	2	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0	0	0	0

$$f_{trans:V,A}(x,y) = \sum_{t=2}^{N} 1\{y_{t-1} = V, y_t = A\}$$

$$f_{obs:V,\text{finna}}(x,y) = \sum_{t=1}^{N} 1\{y_t = V, x_t = \text{finna}\}$$

Goodness(y) = $\theta^{\mathsf{T}} f(x,y)$

Structured/multiclass Perceptron

- Goal: learn model parameters θ from labeled data
- For ~30 iterations
  - For each (x,y) in dataset
    - PREDICT

finna get good

$y^* =$ N V A

$$y^* = \arg\max_{y'} \theta^{\mathsf{T}} f(x, y')$$

G(y)

    - IF y=y*, do nothing
    - ELSE update weights

gold

$$\theta := \theta + r[f(x, y) - f(x, y^*)]$$

learning rate constant
e.g. r=1 r=0.05

Features for TRUE label

Features for PREDICTED label

Update rule

learning rate e.g. r=1

Features for TRUE label

Features for PREDICTED label

$$\theta := \theta + r[f(x, y) - f(x, y^*)]$$

For each feature j in true y but not predicted y*:

$$\theta_j := \theta_j + (r) f_j(x, y)$$

For each feature j not in true y, but in predicted y*:

$$\theta_j := \theta_j - (r) f_j(x, y)$$

	finna	get	good
gold y =	V	V	A
pred y^* =	N	V	A

error

f(x, y)	f(x, y*)	f(x,y) - f(x, y*)
V,V: 1		V,V: +1
	N,V: 1	N,V: -1
V,A: 1	V,A: 1	
	N,finna: 1	N,finna: -1
V,finna: 1		V,finna: +1
V,get: 1	V,get: 1	
A,good: 1	A,good: 1	

$$\theta_j := \theta_j^{(old)} + r(f_j(x,y) - f_j(x,y^*$$

<u>Perceptron update rule:</u> $\theta := \theta + r[f(x,y) - f(x,y^*)]$

.5