CS 378 Lecture 11

Viterbi, Beam search, POS

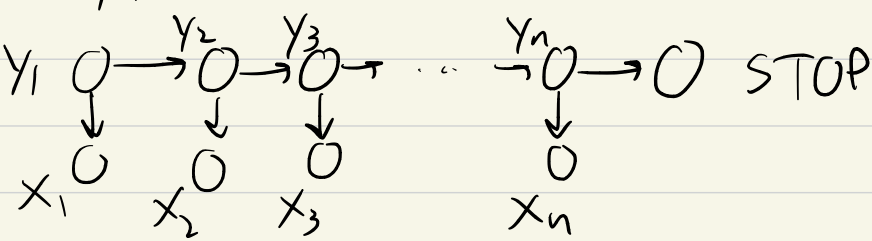
<u>Announcements</u>

- Sam Bowman talk
- A3
- Midterm topics, old exams
- FP

<u>Recap</u>

HMMs $\mathcal{T}$ tags, $\mathcal{V}$ words

$$P(\bar{y}, \bar{x}) = P(y_1)\, P(x_1 \mid y_1)\, P(y_2 \mid y_1)\, P(x_2 \mid y_2) \cdots$$



generative model

Parameters:

$P(y_1)$ initial distribution
$|\mathcal{T}|$-len vector

$P(y_i | y_{i-1})$ transitions, $|\mathcal{T}| \times |\mathcal{T}|$ mat

$P(x_i | y_i)$ emissions, $|\mathcal{T}| \times |\mathcal{V}|$ mat

Inference:

$$\underset{\bar{y}}{\text{argmax}}\ P(\bar{y} | \bar{x}) \longrightarrow \underset{\bar{y}}{\text{argmax}}\ \log P(\bar{y}, \bar{x})$$

"what are the most likely tags for sent $\bar{x}$?"

Use Viterbi algorithm to compute

Example: log probabilities

$$S = \begin{array}{cc} N & -1 \\ V & -1 \end{array} \qquad T = \begin{array}{cccc} & N & V & STOP \\ N & -2 & -1 & -1 \\ V & -1 & -1 & -2 \end{array}$$

$$E = \begin{array}{cccc} & \text{they} & \text{can} & \text{fish} \\ N & -1 & -3 & -1 \\ V & -3 & -1 & -1 \end{array}$$

they can fish: what is the most likely tag sequence?

NNN   8 choices

NNV $\Rightarrow$ each has a score (sum of log probs)

NVN

⋮

$$\begin{array}{ccccccc} -1 & & -1 & & -1 & & -2 \\ & N & & V & & V & & STOP \\ -1 & \text{they} & -1 & \text{can} & -1 & \text{fish} \end{array}$$

7 terms

score $= -8$

N V N STOP $\Rightarrow$ score -7
t c f

<u>Viterbi</u> Dynamic programming

Define $v_i(\tilde{y})$ $n \times |\mathcal{T}|$ matrix
n is sent length

log prob of the best tag sequence ending in $\tilde{y}$ at timestep i

N	$-1 \; -1 = \boxed{-2}$		
V	$-1 \; -3 = \boxed{-4}$		
	they	can	fish

V

Initial: $v_1(\tilde{y}) = \log P(x_1 | \tilde{y}) + \overset{\text{initial dist}}{\log P(\tilde{y})}$

Recurrent: compute v_i using v_{i-1}

$$v_i(\tilde{y}) = \log P(x_i | \tilde{y}) + \max_{\tilde{y}_{prev}} \left[\log P(\tilde{y} | \tilde{y}_{prev}) + v_{i-1}(\tilde{y}_{prev}) \right]$$

<u>Viterbi alg</u>

for $i = 1 \dots n$

  for $\tilde{y} \in \mathcal{T}$

    Compute $v_i(\tilde{y})$ as above

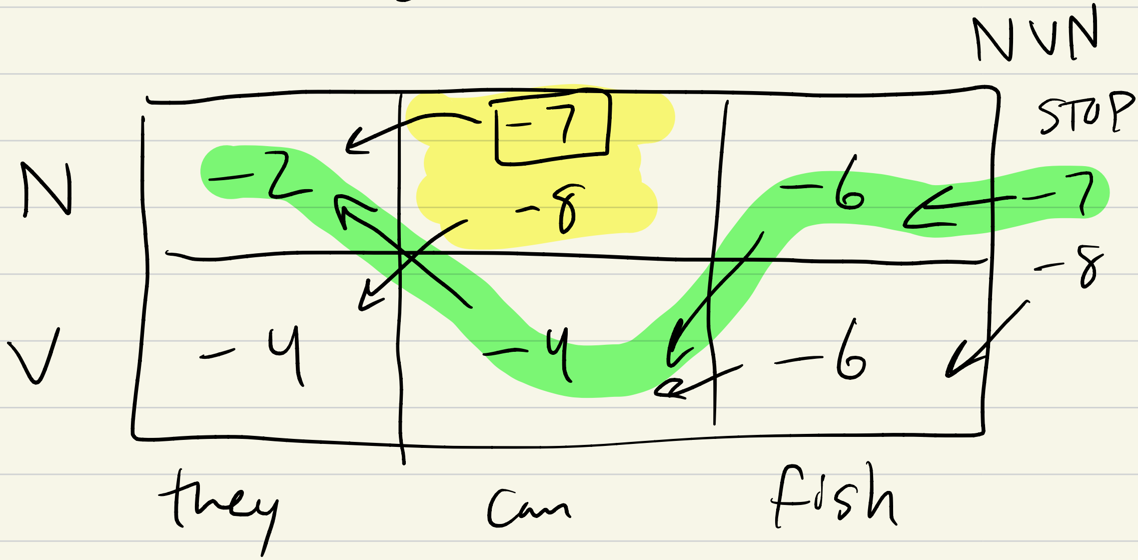
Compute $v_{n+1}(STOP)$ = do the recurrent one more time

$$v_{n+1}(STOP) = \max_{\bar{y}} \log P(\bar{y}, \bar{x})$$

Track "backpointers" to reconstruct the sequence

$$S = \begin{array}{cc} N & -1 \\ V & -1 \end{array} \qquad T = \begin{array}{c|ccc} & N & V & STOP \\ \hline N & -2 & -1 & -1 \\ V & -1 & -1 & -2 \end{array}$$

$$E = \begin{array}{c|ccc} & \text{they} & \text{can} & \text{fish} \\ \hline N & -1 & -3 & -1 \\ V & -3 & -1 & -1 \end{array}$$



$V_2(N) = \cancel{\log P(\text{can} \mid N)} \to -3$

$$+ \max \begin{cases} N: \log P(N \mid N) + v_1(N) & -2 - 2 = -4 \\ V: \log P(N \mid V) + v_1(V) & -1 - 4 = -5 \end{cases}$$

$= -7$

Beam search

Viterbi runtime: $O(|\mathcal{T}|^2 \cdot n)$

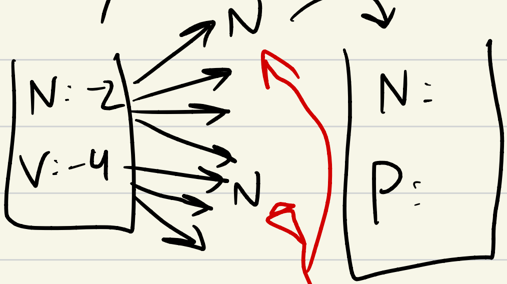
	they	can	fish
N	-2		
V	-4		
P	-20		
CC	-20		
•	-20		
\$	-20		
⋮	⋮		

most transitions are bad

Beam search: only track top K scoring states in each column they $k \cdot |T|$ possibilities

add to beam

max over duplicates



everything else is kicked out

keep only higher N

$$O(n|T|^2) \Rightarrow O(n|T|k \log k)$$

(really $\log k$ from data structure)

$k=1$: greedy

k ranges from 2 to 100