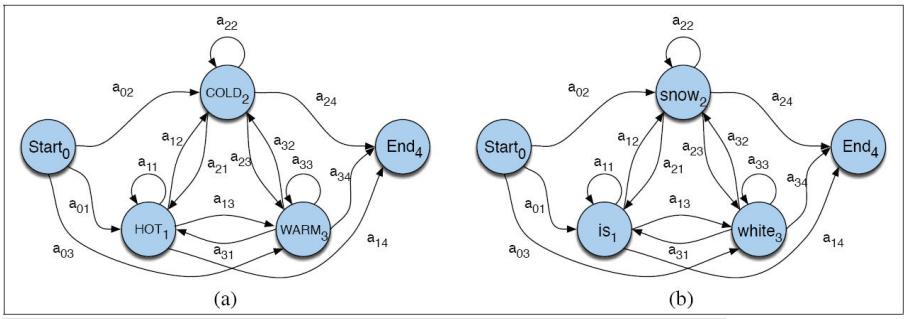
HMMs → MEMMs (→CRFs)

- Algorithms for HMMs
- MaxEnt classifier
- MEMMs

Markov chains

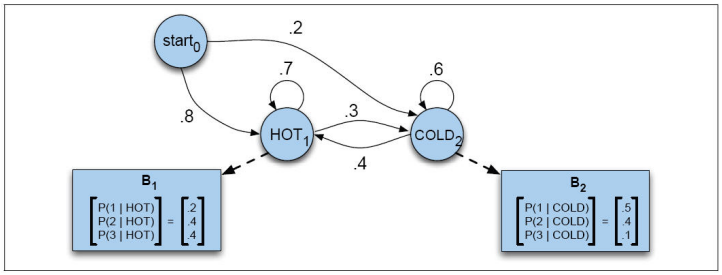


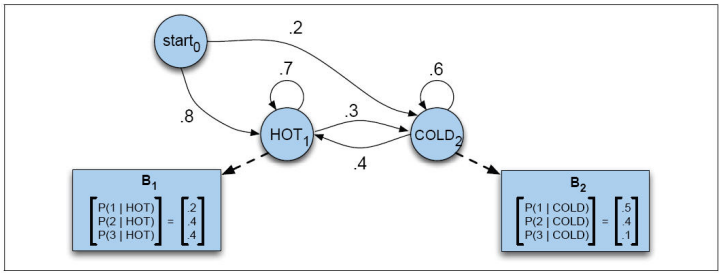
$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{01} a_{02} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
q_0, q_F	a special **start state** and **end (final) state** that are not associated with observations

Assign probability to an unambiguous sequence

Hidden Markov model

- Useful when we want to determine the probability of a sequence of events that is not directly observable, e.g. POS tagging **or temperature (given # of ice cream cones eaten)**





$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
q_0, q_F	a special **start state** and **end (final) state** that are not associated with observations, together with transition probabilities $a_{01} a_{02} \ldots a_{0n}$ out of the start state and $a_{1F} a_{2F} \ldots a_{nF}$ into the end state

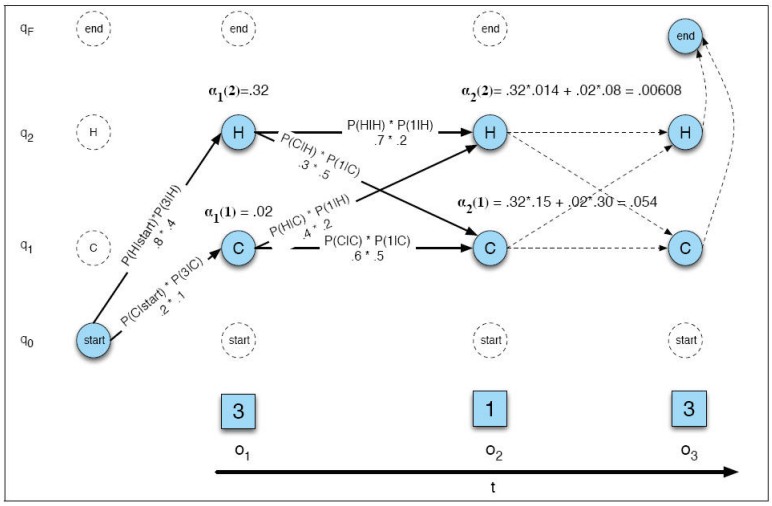
Forward algorithm

Viterbi

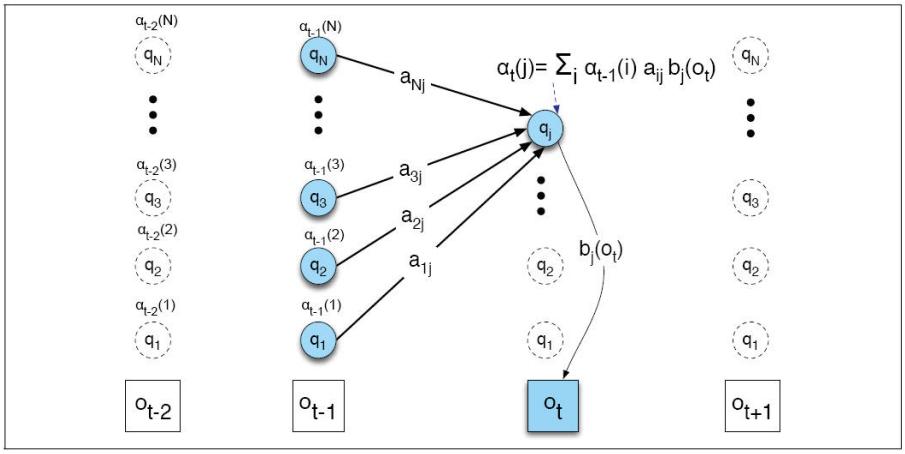
Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Forward-backward algorithm / EM

Forward algorithm



Different from forward pass of Viterbi



function FORWARD(*observations* of len T, *state-graph* of len N) **returns** *forward-prob*

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

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

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

return $forward[q_F, T]$

Forward algorithm

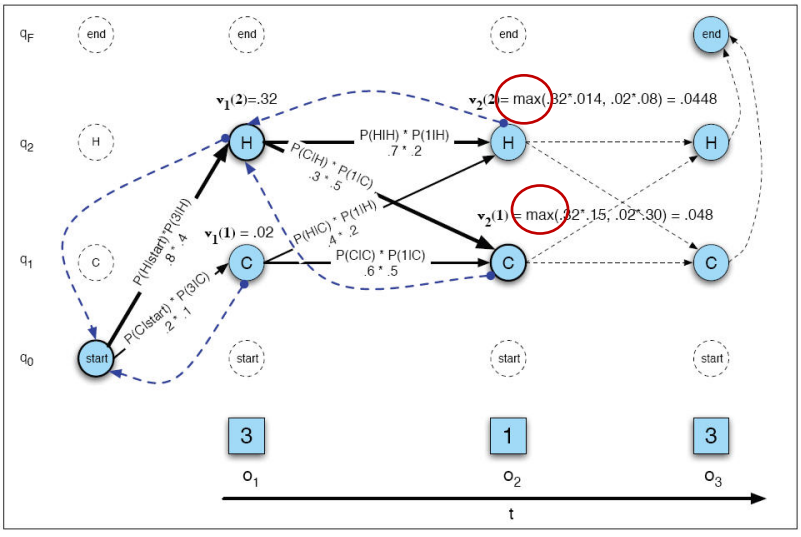
Viterbi

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Forward-backward algorithm / EM

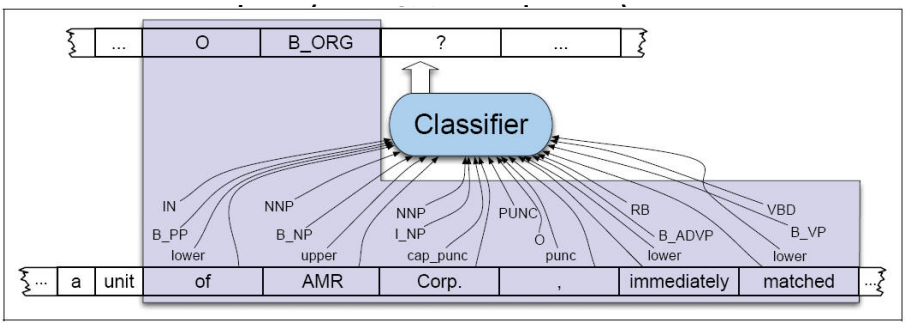
Not covering this

Viterbi



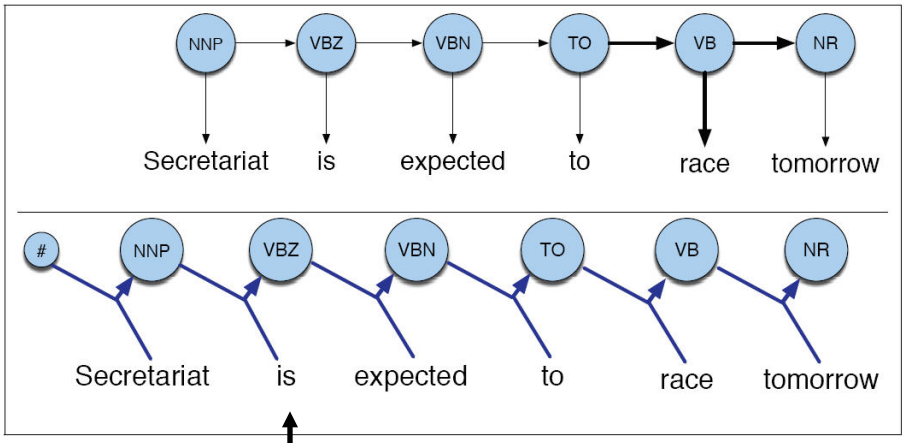
Feature extraction

- We'd like to be able to include lots of features as in classification-based



Figure, copyright J&M 2nd ed

Not possible with HMMs



Maximum entropy Markov model (MEMM)

Solution

- Combine classifier AND Viterbi
- Use MaxEnt classifier
  - Multinomial logistic regression

HMMs → MEMMs (→CRFs)

- Algorithms for HMMs
- MaxEnt classifier
- MEMMs

MaxEnt

- From the class of **exponential** or **log-linear** classifiers
  - Extracts features from the input
  - Combines them linearly
  - Uses the sum as an exponent

c class
x input
w weight
f feature
Z normalization factor

$$p(c|x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i\right)$$

MaxEnt

- A bit of a simplification
- Features *f* and weights *w* depend on the class (so f_i is really $f_i(c,x)$)
- Features are **indicator functions**, return 0 or 1

$$p(c|x) = \frac{\exp\left(\sum_{i=0}^{N} w_{ci} f_i\right)}{\sum_{c' \in C} \exp\left(\sum_{i=0}^{N} w_{c'i} f_i\right)} \leftarrow Z$$

(# of features; # of classes)

Features

- Should **race** be a VB or an NN?

Secretariat/NNP is/BEZ expected/VBN to/TO **race**/?? tomorrow/ADV

$$f_1(c, x) = \begin{cases} 1 & \text{if } word_i = \text{“race”} \ \& \ c = \text{NN} \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(c, x) = \begin{cases} 1 & \text{if } t_{i-1} = \text{TO} \ \& \ c = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

MaxEnt

- Provides a probability distribution over the classes
- For classification, choose highest
  - $argmax_{c \text{ in } C}$
- In sequence tagging, use the distribution
- See textbook for how to LEARN weights

HMMs → MEMMs (→CRFs)

- Algorithms for HMMs
- MaxEnt classifier
- MEMMs

HMM: find best possible tag sequence
highest probability given the word in the sentence

HMM vs. MEMM

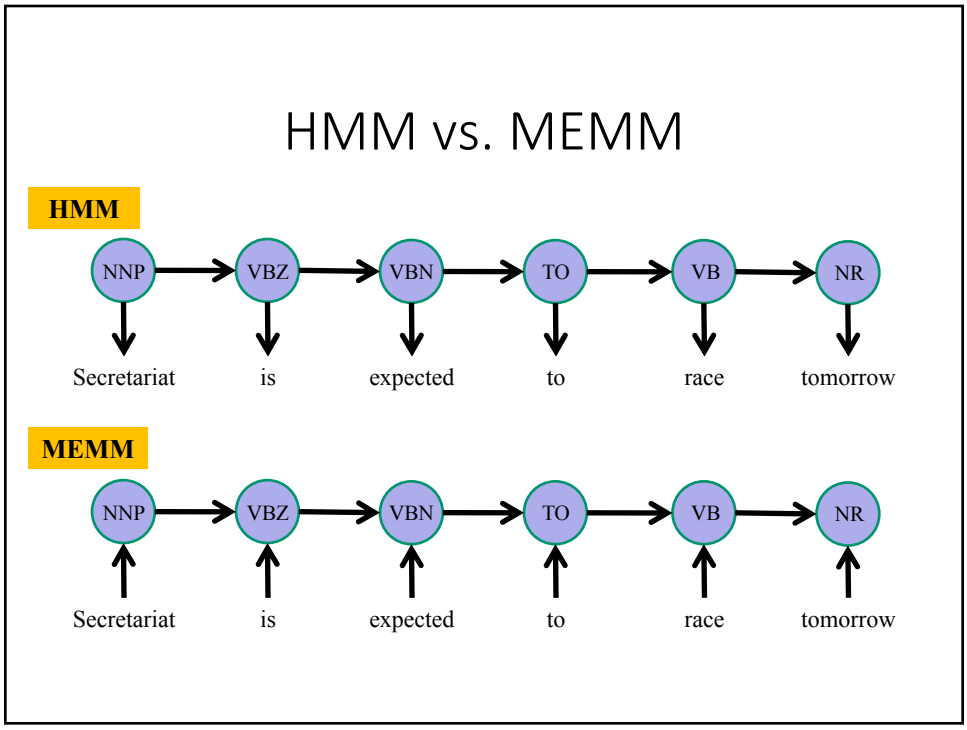
HMM

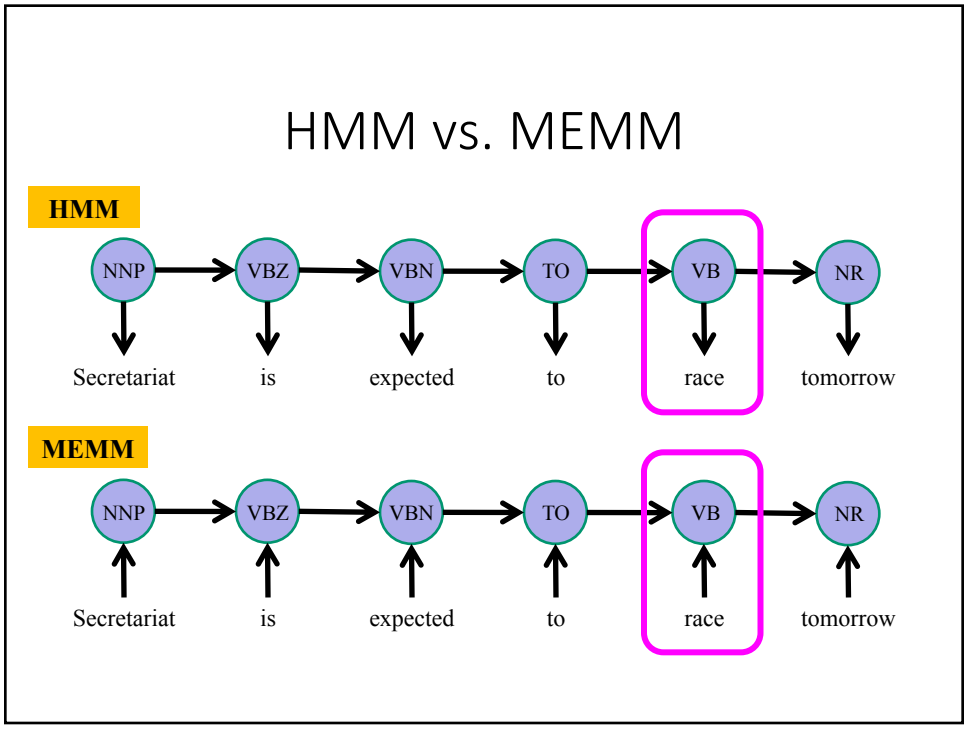
$$\hat{T} = \operatorname{argmax}_T P(T|W)$$
$$= \operatorname{argmax}_T P(W|T)\,P(T)$$
$$= \operatorname{argmax}_T \prod_i P(word_i|tag_i) \prod_i P(tag_i|tag_{i-1})$$

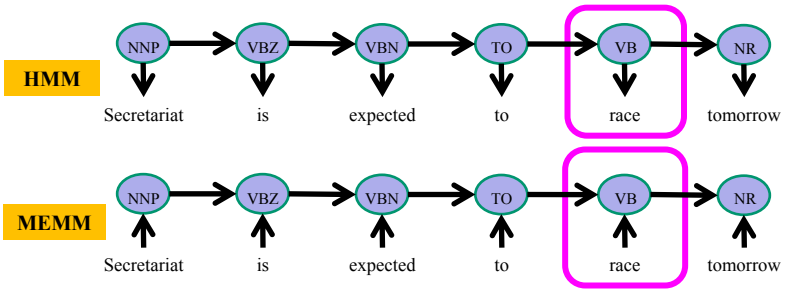
MEMM

$$\hat{T} = \operatorname{argmax}_T P(T|W)$$
$$= \operatorname{argmax}_T \prod_i P(tag_i|word_i, tag_{i-1})$$

Go through all possible tag sequences w/o exponential time

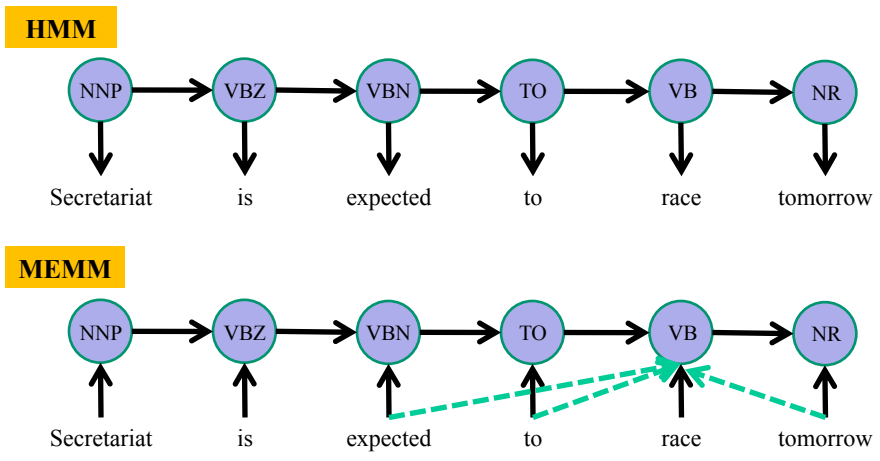

HMM vs. MEMM
HMM
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow
MEMM
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow


HMM vs. MEMM
HMM
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow
MEMM
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow



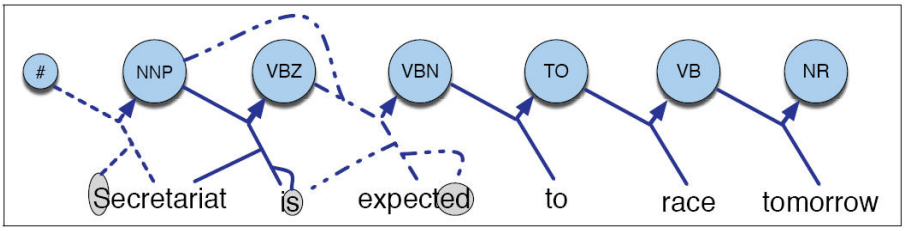
HMM	MEMM
"Generative" models ➔ joint probability **p(words, tags)** ➔"generate" input (in addition to tags) ➔ but we need to predict tags, not words!	"Discriminative" or "Conditional" models ➔ conditional probability **p(tags | words)** ➔"condition" on input ➔ Focusing only on predicting tags
Probability of each slice = emission * transition = p(word_i | tag_i) * p(tag_i | tag_i-1) = ➔ Cannot incorporate long distance features	Probability of each slice = p(tag_i | tag_i-1, word_i) or p(tag_i | tag_i-1, all words) ➔ Can incorporate long distance features

HMM vs. MEMM

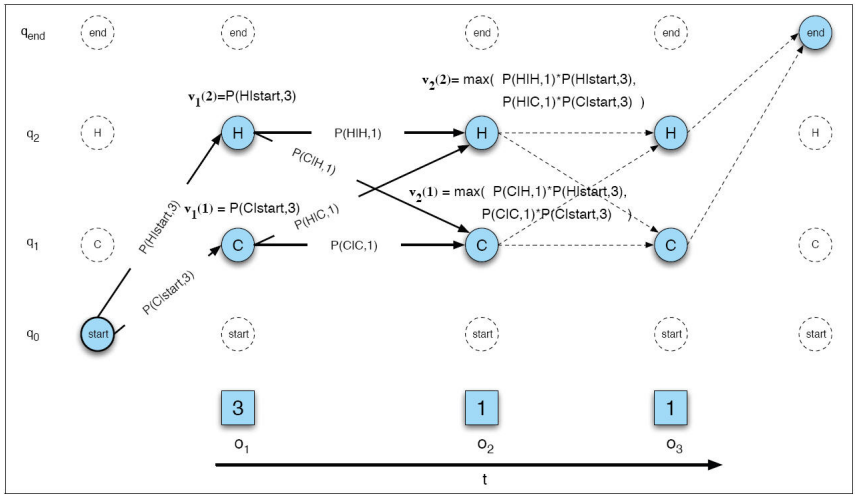


MEMM's

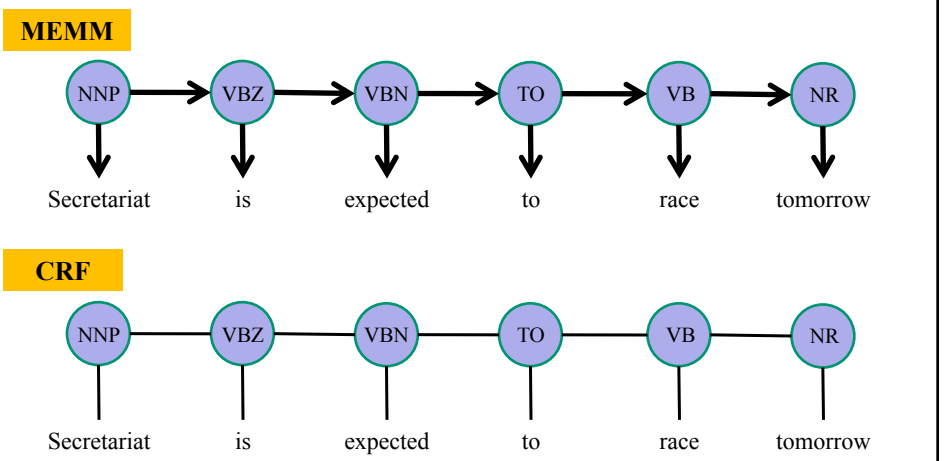
- Can condition on **many** features of the input

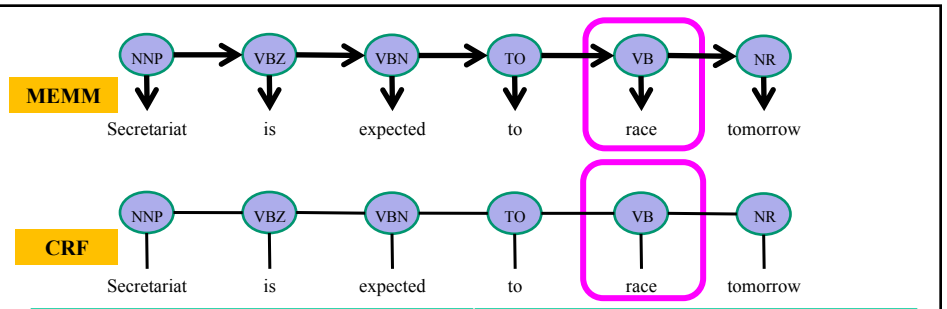


Inference (Viterbi)



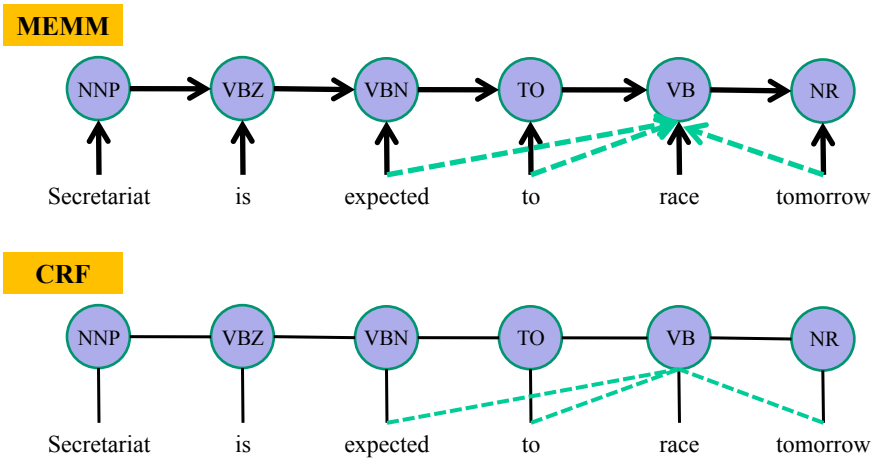
MEMM vs. CRF





MEMM	CRF
Directed graphical model	Undirected graphical model
"Discriminative" or "Conditional" models ➔ conditional probability **p(tags | words)**	
Probability is defined for each slice = P (tag_i | tag_i-1, word_i) or p (tag_i | tag_i-1, all words)	Instead of probability, ***potential (energy function)*** is defined for each slice = ϕ (tag_i, tag_i-1) * ϕ (tag_i, word_i) or ϕ(tag_i, tag_i-1, all words) * ϕ (tag_i, all words)
➔ Can incorporate long distance features	

MEMM vs. CRF


MEMM
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow

CRF
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow