# Probabilistic Tagging

- $W = w_1 \dots w_n$ - words in the corpus (observed)
- $T = t_1 \dots t_n$ - the corresponding tags (unknown)

# Probabilistic Tagging

- $W = w_1 \dots w_n$ - words in the corpus (observed)
- $T = t_1 \dots t_n$ - the corresponding tags (unknown)

### *Tagging: Probabilistic View (Generative Model)*

Find

$$\hat{T} = argmax_T P(T|W)$$

# Probabilistic Tagging

- $W = w_1 \ldots w_n$ - words in the corpus (observed)
- $T = t_1 \ldots t_n$ - the corresponding tags (unknown)

*Tagging: Probabilistic View (Generative Model)*

Find

$$\begin{aligned} \hat{T} &= argmax_T P(T|W) \\ &= argmax_T \frac{P(W|T)P(T)}{P(W)} \end{aligned}$$

# Probabilistic Tagging

- $W = w_1 \dots w_n$ - words in the corpus (observed)
- $T = t_1 \dots t_n$ - the corresponding tags (unknown)

*Tagging: Probabilistic View (Generative Model)*

Find

$$\begin{aligned}\hat{T} &= argmax_T P(T|W) \\ &= argmax_T \frac{P(W|T)P(T)}{P(W)} \\ &= argmax_T P(W|T)P(T)\end{aligned}$$

# Probabilistic Tagging

- $W = w_1 \ldots w_n$ - words in the corpus (observed)
- $T = t_1 \ldots t_n$ - the corresponding tags (unknown)

### Tagging: Probabilistic View (Generative Model)

Find

$$
\begin{aligned}
\hat{T} &= argmax_T P(T|W) \\
&= argmax_T \frac{P(W|T)P(T)}{P(W)} \\
&= argmax_T P(W|T)P(T) \\
&= argmax_T \prod_i P(w_i|w_1 \ldots w_{i-1}, t_1 \ldots t_i) P(t_i|t_1 \ldots t_{i-1})
\end{aligned}
$$

# *Further simplifications*

$$\hat{T} = argmax_T \prod_i P(w_i|w_1 \ldots w_{i-1}, t_1 \ldots t_i) P(t_i|t_1 \ldots t_{i-1})$$

- The probability of a word appearing depends only on its own POS tag

# Further simplifications

$$\hat{T} = argmax_T \prod_i P(w_i|w_1 \ldots w_{i-1}, t_1 \ldots t_i) P(t_i|t_1 \ldots t_{i-1})$$

- The probability of a word appearing depends only on its own POS tag
  $P(w_i|w_1 \ldots w_{i-1}, t_1 \ldots t_i) \approx P(w_i|t_i)$
- Bigram assumption: the probability of a tag appearing depends only on the previous tag

# Further simplifications

$$\hat{T} = argmax_T \prod_i P(w_i|w_1 \dots w_{i-1}, t_1 \dots t_i) P(t_i|t_1 \dots t_{i-1})$$

- The probability of a word appearing depends only on its own POS tag
  $P(w_i|w_1 \dots w_{i-1}, t_1 \dots t_i) \approx P(w_i|t_i)$
- Bigram assumption: the probability of a tag appearing depends only on the previous tag
  $P(t_i|t_1 \dots t_{i-1}) \approx P(t_i|t_{i-1})$

# Further simplifications

$$\hat{T} = argmax_T \prod_i P(w_i|w_1 \dots w_{i-1}, t_1 \dots t_i) P(t_i|t_1 \dots t_{i-1})$$

- The probability of a word appearing depends only on its own POS tag
  $P(w_i|w_1 \dots w_{i-1}, t_1 \dots t_i) \approx P(w_i|t_i)$
- Bigram assumption: the probability of a tag appearing depends only on the previous tag
  $P(t_i|t_1 \dots t_{i-1}) \approx P(t_i|t_{i-1})$
- Using these simplifications:
  $\hat{T} = argmax_T \prod_i P(w_i|t_i) P(t_i|t_{i-1})$

# *Computing the probability values*

### *Tag Transition probabilities $p(t_i|t_{i-1})$*

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

$$P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = 0.49$$

# Computing the probability values

### Tag Transition probabilities $p(t_i|t_{i-1})$

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

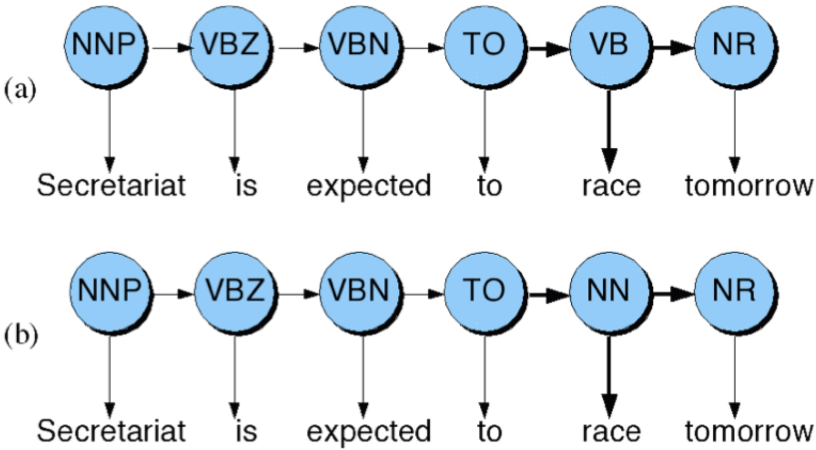
$$P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = 0.49$$

### Word Likelihood probabilities $p(w_i|t_i)$

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = 0.47$$

# *Disambiguating "race"*



# *Disambiguating "race"*

*Difference in probability due to*

- $P(VB|TO)$ vs. $P(NN|TO)$
- $P(race|VB)$ vs. $P(race|NN)$
- $P(NR|VB)$ vs. $P(NR|NN)$

# Disambiguating "race"

## Difference in probability due to

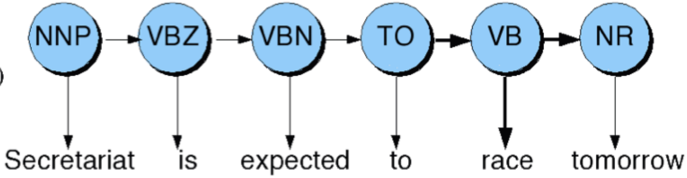
- $P(VB|TO)$ vs. $P(NN|TO)$
- $P(race|VB)$ vs. $P(race|NN)$
- $P(NR|VB)$ vs. $P(NR|NN)$

## After computing the probabilities

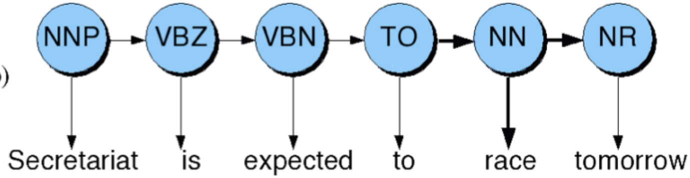
- $P(NN|TO)P(NR|NN)P(race|NN) = 0.0047 \times 0.0012 \times 0.00057 = 0.00000000032$
- $P(VB|TO)P(NR|VB)P(race|VB) = 0.83 \times 0.0027 \times 0.00012 = 0.00000027$

# What is this model?

(a)

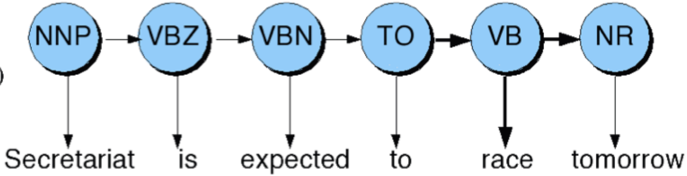



(b)

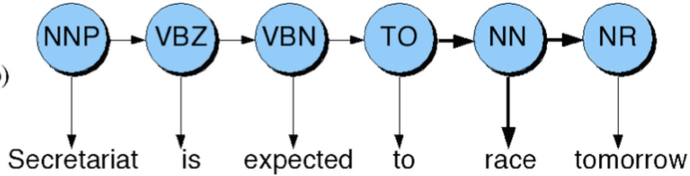


# What is this model?

(a)




(b)




*This is a Hidden Markov Model*

# Hidden Markov Models

- Tag Transition probabilities $p(t_i|t_{i-1})$
- Word Likelihood probabilities (emissions) $p(w_i|t_i)$

# Hidden Markov Models

- Tag Transition probabilities $p(t_i|t_{i-1})$
- Word Likelihood probabilities (emissions) $p(w_i|t_i)$
- What we have described with these probabilities is a hidden markov model.
- Let us quickly introduce the Markov Chain, or observable Markov Model.

# *Markov Chain = First-order Markov Model*

### *Weather example*

- Three types of weather: *sunny, rainy, foggy*

# *Markov Chain = First-order Markov Model*

*Weather example*

- Three types of weather: *sunny, rainy, foggy*
- $q_n$: variable denoting the weather on the $n^{th}$ day

# Markov Chain = First-order Markov Model

### *Weather example*

- Three types of weather: *sunny, rainy, foggy*
- $q_n$: variable denoting the weather on the $n^{th}$ day
- We want to find the following conditional probabilities:

$$P(q_n|q_{n-1}, q_{n-2}, \ldots, q_1)$$

# Markov Chain = First-order Markov Model

## *Weather example*

- Three types of weather: *sunny, rainy, foggy*
- $q_n$: variable denoting the weather on the $n^{th}$ day
- We want to find the following conditional probabilities:

$$P(q_n|q_{n-1}, q_{n-2}, \ldots, q_1)$$

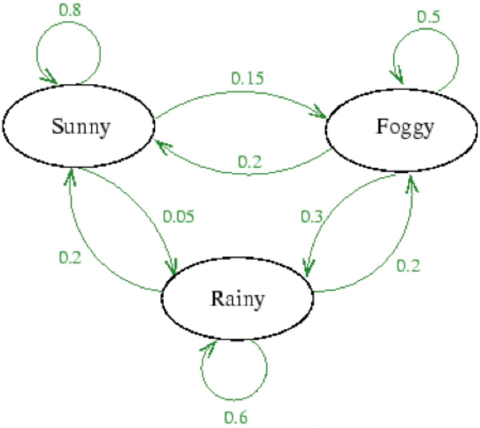
## *First-order Markov Assumption*

$$P(q_n|q_{n-1}, q_{n-2}, \ldots, q_1) = P(q_n|q_{n-1})$$

# Markov Chain Transition Table

**Table 1:** Probabilities $p(q_{n+1}|q_n)$ of tomorrow's weather based on today's weather

| | Tomorrow's weather | | |
|---|---|---|---|
| Today's weather |  |  |  |
|  | 0.8 | 0.05 | 0.15 |
|  | 0.2 | 0.6 | 0.2 |
|  | 0.2 | 0.3 | 0.5 |



# *Using Markov Chain*

- Given that today the weather is sunny, what is the probability that tomorrow is sunny and day after is rainy?

# *Using Markov Chain*

- Given that today the weather is sunny, what is the probability that tomorrow is sunny and day after is rainy?

$$P(q_2 = sunny, q_3 = rainy | q_1 = sunny)$$

# Using Markov Chain

- Given that today the weather is sunny, what is the probability that tomorrow is sunny and day after is rainy?

$$P(q_2 = sunny, q_3 = rainy | q_1 = sunny)$$

$$= P(q_3 = rainy | q_2 = sunny, q_1 = sunny) \times P(q_2 = sunny | q_1 = sunny)$$

# Using Markov Chain

- Given that today the weather is sunny, what is the probability that tomorrow is sunny and day after is rainy?

$P(q_2 = sunny, q_3 = rainy|q_1 = sunny)$

$$\begin{aligned} &= P(q_3 = rainy|q_2 = sunny, q_1 = sunny) \times P(q_2 = sunny|q_1 = sunny) \\ &= P(q_3 = rainy|q_2 = sunny) \times P(q_2 = sunny|q_1 = sunny) \\ &= 0.05 \times 0.8 \\ &= 0.04 \end{aligned}$$

# Hidden Markov Model

- For Markov chains, the output symbols are the same as the states
  *'sunny' weather is both observable and state*

# *Hidden Markov Model*

- For Markov chains, the output symbols are the same as the states
  *'sunny' weather is both observable and state*
- But in POS tagging

# Hidden Markov Model

- For Markov chains, the output symbols are the same as the states
  *'sunny' weather is both observable and state*
- But in POS tagging
  *The output symbols are words*

# *Hidden Markov Model*

- For Markov chains, the output symbols are the same as the states
  *'sunny' weather is both observable and state*
- But in POS tagging
  *The output symbols are words*
  *But the hidden states are POS tags*
- A Hidden Markov Model is an extension of a Markov chain in which the output symbols are not the same as the states
- We don't know which state we are in

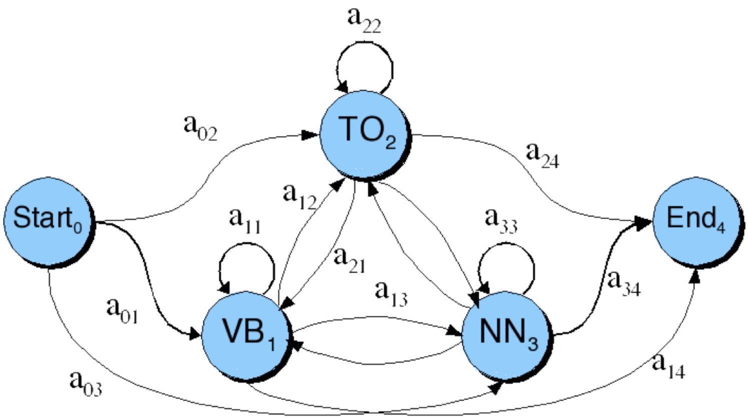
# Hidden Markov Models (HMMs)

### Elements of an HMM model

- A set of states (here: the tags)
- An output alphabet (here: words)
- Initial state (here: beginning of sentence)
- State transition probabilities (here $p(t_n|t_{n-1})$)
- Symbol emission probabilities (here $p(w_i|t_i)$)

# *Graphical Representation*

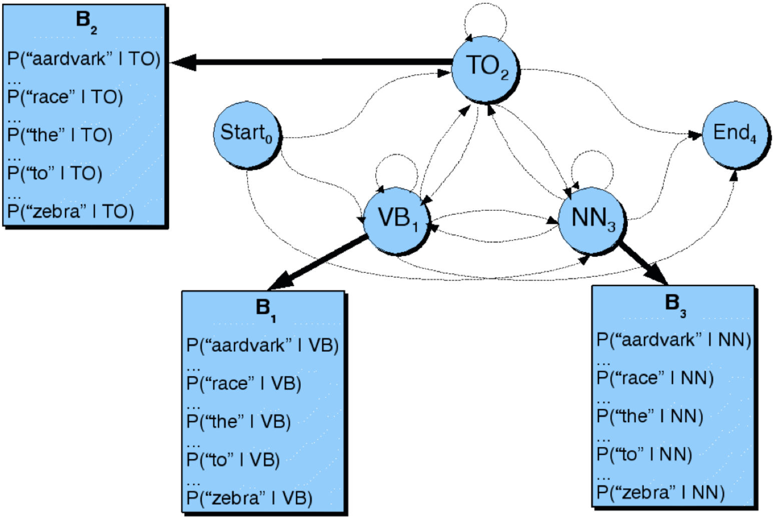
When tagging a sentence, we are walking through the state graph:



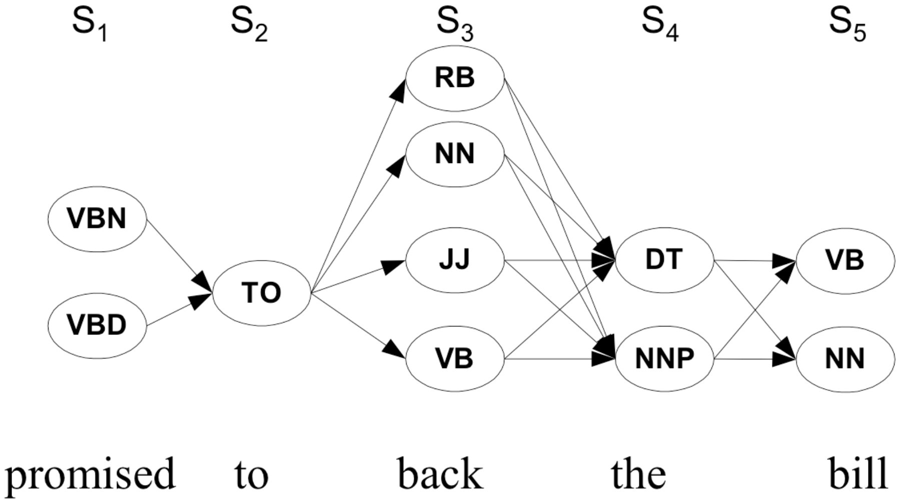

Edges are labeled with the state transition probabilities: $p(t_n|t_{n-1})$

# Graphical Representation

At each state we emit a word: $P(w_n|t_n)$



# *Walking through the states: best path*



# Walking through the states: best path