Statistical POS Tagging

Natalie Parde
UIC CS 421

Statistical POS Tagging

- Predicts POS tags based on the probabilities of those tags occurring
- Probabilities can be based on various sources of information
- Doing this requires a **training corpus**
  - No probabilities associated with words not in the corpus!

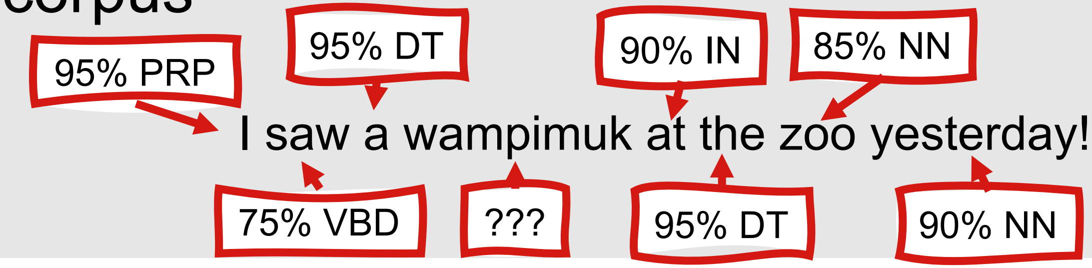
Simple Statistical POS Tagger

- Using a training corpus, determine the most frequent tag for each word
- Assign POS tags to new words based on those frequencies
- Assign NN to new words for which there is no information from the training corpus

I saw a wampimuk at the zoo yesterday!

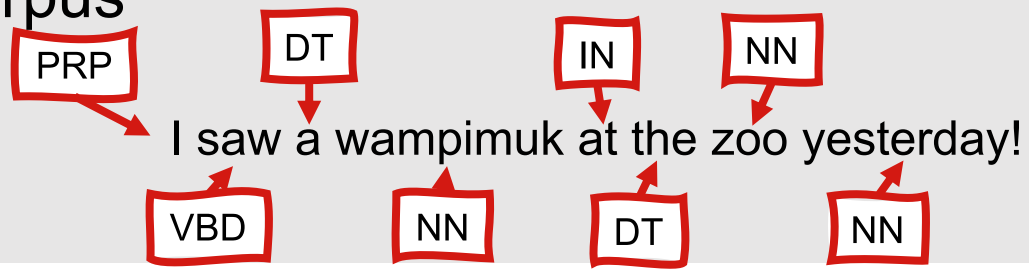
Simple Statistical POS Tagger

- Using a training corpus, determine the most frequent tag for each word
- Assign POS tags to new words based on those frequencies
- Assign NN to new words for which there is no information from the training corpus



Simple Statistical POS Tagger

- Using a training corpus, determine the most frequent tag for each word
- Assign POS tags to new words based on those frequencies
- Assign NN to new words for which there is no information from the training corpus



Simple Statistical POS Tagger

- This approach works reasonably well
  - Approximately 90% accuracy
- However, we can do much better!
- One way to improve upon our results is to use **HMMs**

HMM POS Tagger

- Selects the most likely tag sequence for a sequence of observed words, maximizing the following formula:
  - P(word | tag) * P(tag | previous *n* tags)
- More formally, letting $T = \{t_1, t_2, \ldots, t_n\}$ and $W = \{w_1, w_2, \ldots, w_n\}$, find the most probable sequence of tags *T* underlying the observed words *W*

What do we mean by “previous *n* tags”?

- For our example here, we’ll assume *n*=1 and create a bigram HMM tagger, meaning we’re only looking at a word/tag given the word/tag immediately preceding it

Bigram HMM Tagger

- To determine the tag t_i for a single word w_i:
  - $t_i = \underset{t_j \in \{t_0, t_1, \ldots, t_{t-1}\}}{\operatorname{argmax}} P(t_j | t_{i-1}) P(w_i | t_j)$
- This means we need to be able to compute two probabilities:
  - The probability that the tag is t_j given that the previous tag is t_{i-1}
    - $P(t_j | t_{i-1})$
  - The probability that the word is w_i given that the tag is t_j
    - $P(w_i | t_j)$
- We can compute both of these from corpora like the Penn Treebank or the Brown Corpus
- Then, we can find the most optimal sequence of tags using the Viterbi algorithm!

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

- Given two possible sequences of tags for the following sentence, what is the best way to tag the word "race"?
- Brown Corpus tagset:
  - Contains a specific tag for the infinitive use of "to"
  - Labels "tomorrow" as NR (adverbial noun) rather than NN (singular common noun)

Example: Bigram HMM Tagger

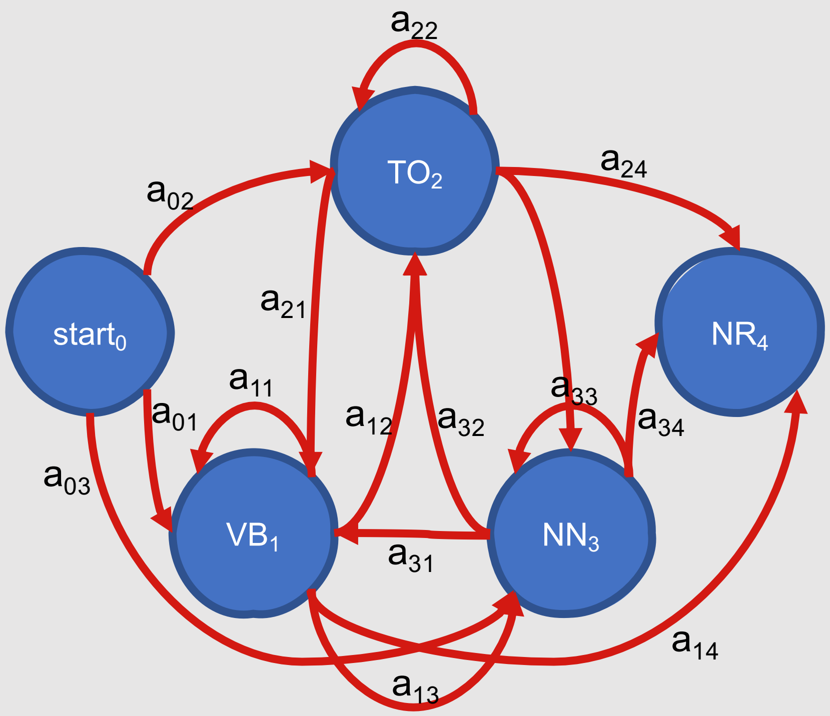
Example: Bigram HMM Tagger

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

- Since we're creating a bigram HMM tagger and focusing on the word "race," we only need to be concerned with the subsequence "to race tomorrow"

We can thus create the following Markov chain:

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

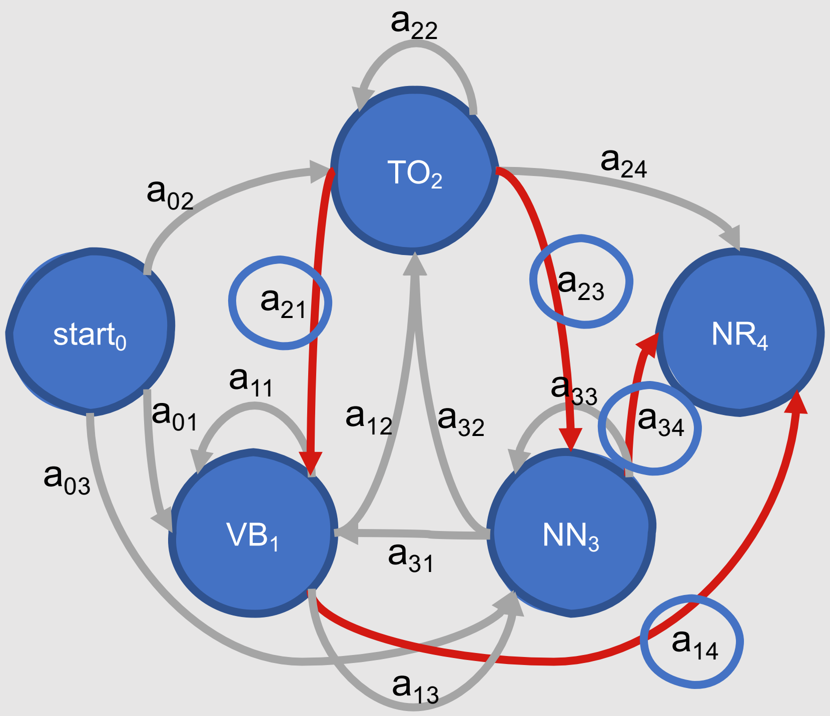


Example: Bigram HMM Tagger

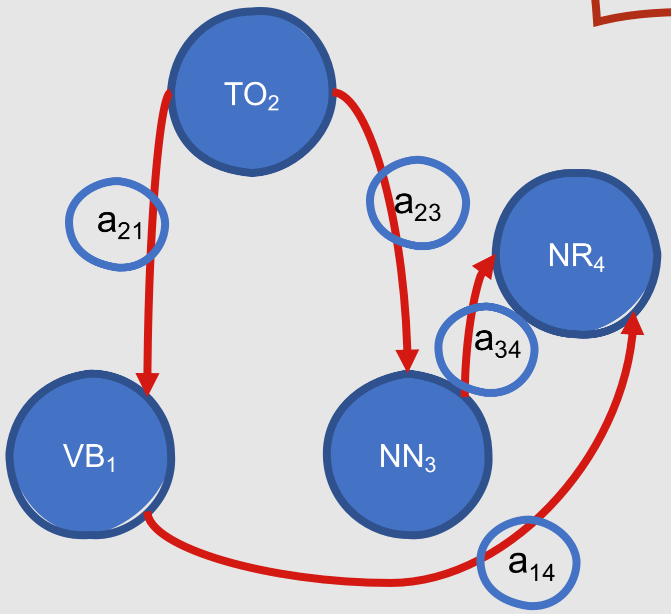
Example: Bigram HMM Tagger

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

The specific transition probabilities we are interested in are:



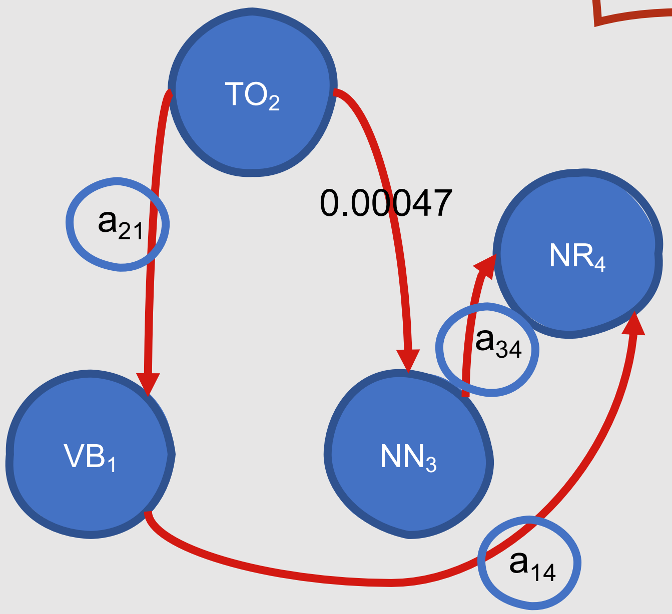
Secretariat	is	expected	to	race	tomorrow
NNP	VBZ	VBN	TO	VB	NR
NNP	VBZ	VBN	TO	NN	NR



- We can compute the transition probabilities for a_{21}, a_{23}, a_{34}, and a_{14} using frequency counts from the Brown Corpus
- $P(t_i|t_{i-1}) = \frac{c(t_{i-1}t_i)}{c(t_{i-1})}$

Example: Bigram HMM Tagger

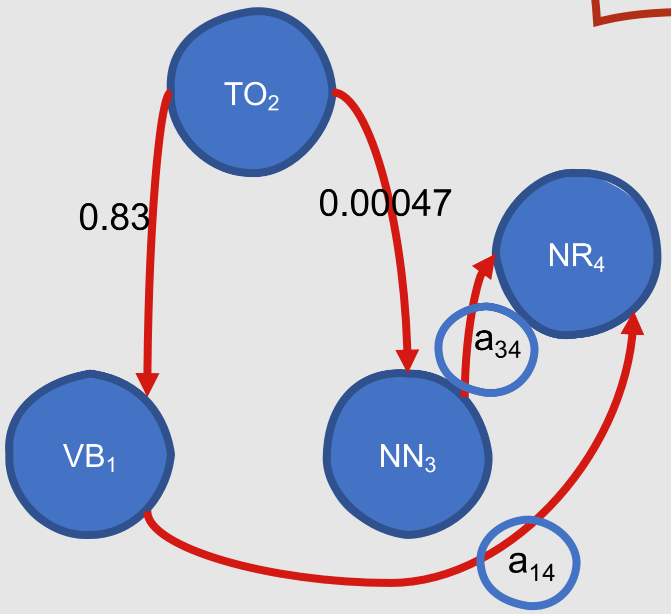
Secretariat	is	expected	to	race	tomorrow
NNP	VBZ	VBN	TO	VB	NR
NNP	VBZ	VBN	TO	NN	NR



- We can compute the transition probabilities for a_{21}, a_{23}, a_{34}, and a_{14} using frequency counts from the Brown Corpus
- $P(t_i|t_{i-1}) = \frac{c(t_{i-1}t_i)}{c(t_{i-1})}$
- So, P(NN|TO) = C(TO NN) / C(TO) = 0.00047

Example: Bigram HMM Tagger

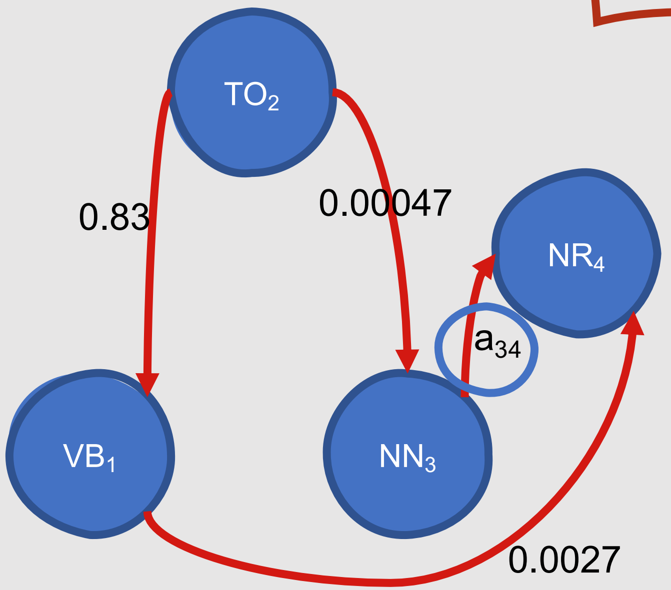
Secretariat	is	expected	to	race	tomorrow
NNP	VBZ	VBN	TO	VB	NR
NNP	VBZ	VBN	TO	NN	NR



- We can compute the transition probabilities for a_{21}, a_{23}, a_{34}, and a_{14} using frequency counts from the Brown Corpus
- $P(t_i|t_{i-1}) = \frac{c(t_{i-1}t_i)}{c(t_{i-1})}$
- So, P(NN|TO) = C(TO NN) / C(TO) = 0.00047
- Likewise, P(VB|TO) = C(TO VB) / C(TO) = 0.83

Example: Bigram HMM Tagger

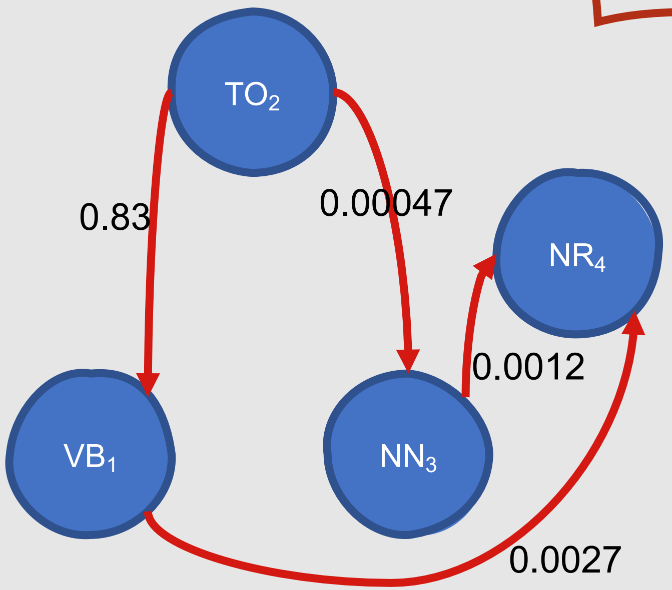
Secretariat	is	expected	to	race	tomorrow
NNP	VBZ	VBN	TO	VB	NR
NNP	VBZ	VBN	TO	NN	NR



- We can compute the transition probabilities for a_{21}, a_{23}, a_{34}, and a_{14} using frequency counts from the Brown Corpus
- $P(t_i|t_{i-1}) = \frac{c(t_{i-1}t_i)}{c(t_{i-1})}$
- So, P(NN|TO) = C(TO NN) / C(TO) = 0.00047
- Likewise, P(VB|TO) = C(TO VB) / C(TO) = 0.83
- P(NR|VB) = C(VB NR) / C(VB) = 0.0027

Example: Bigram HMM Tagger

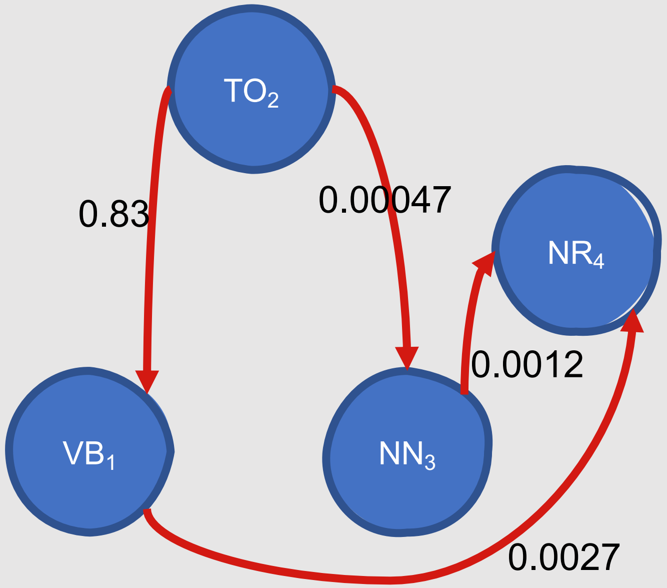
Secretariat	is	expected	to	race	tomorrow
NNP	VBZ	VBN	TO	VB	NR
NNP	VBZ	VBN	TO	NN	NR



- We can compute the transition probabilities for a_{21}, a_{23}, a_{34}, and a_{14} using frequency counts from the Brown Corpus
- $P(t_i|t_{i-1}) = \frac{c(t_{i-1}t_i)}{c(t_{i-1})}$
- So, P(NN|TO) = C(TO NN) / C(TO) = 0.00047
- Likewise, P(VB|TO) = C(TO VB) / C(TO) = 0.83
- P(NR|VB) = C(VB NR) / C(VB) = 0.0027
- Finally, P(NR|NN) = C(NN NR) / C(NN) = 0.0012

Example: Bigram HMM Tagger

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

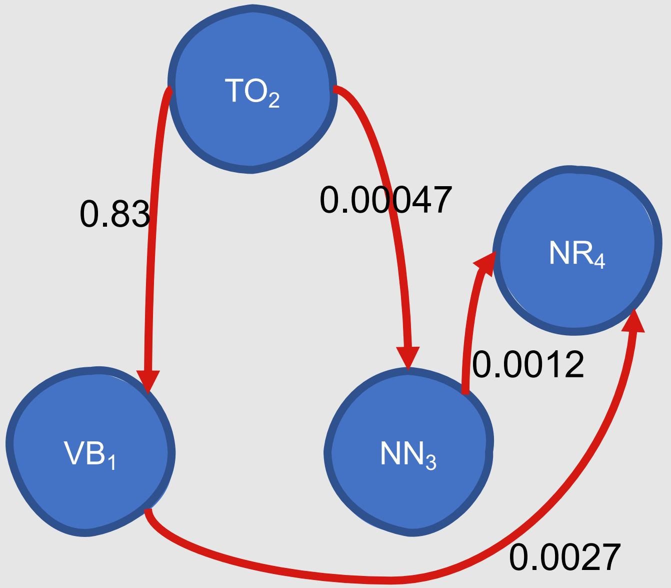


	race
VB	
NN	

- We have our transition probabilities …what now?
- Observation likelihoods!
- We can also compute these using frequency counts from the Brown Corpus
- $P(w_i|t_i) = \frac{c(w_i,t_i)}{c(t_i)}$
- Since we're trying to decide the best tag for "race," we need to compute both P(race|VB) and P(race|NN)

Example: Bigram HMM Tagger

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

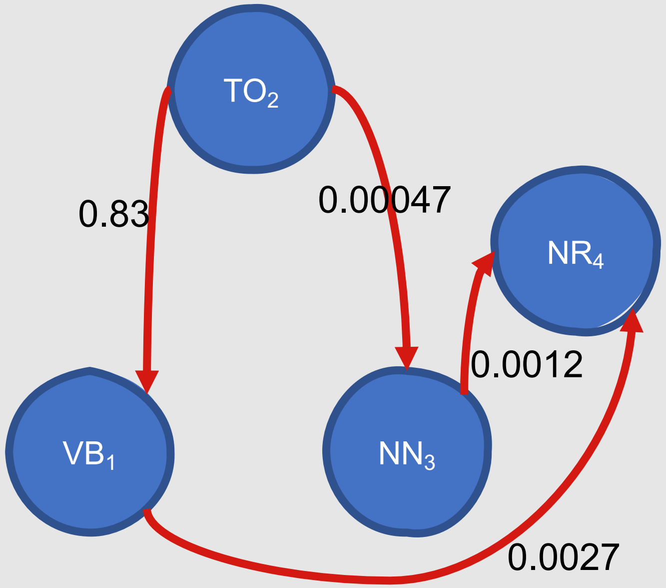


	race
VB	0.00012
NN	

- We have our transition probabilities …what now?
- Observation likelihoods!
- We can also compute these using frequency counts from the Brown Corpus
- $P(w_i|t_i) = \frac{c(w_i,t_i)}{c(t_i)}$
- Since we're trying to decide the best tag for "race," we need to compute both P(race|VB) and P(race|NN)
- P(race|VB) = C(race, VB) / C(VB) = 0.00012

Example: Bigram HMM Tagger

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

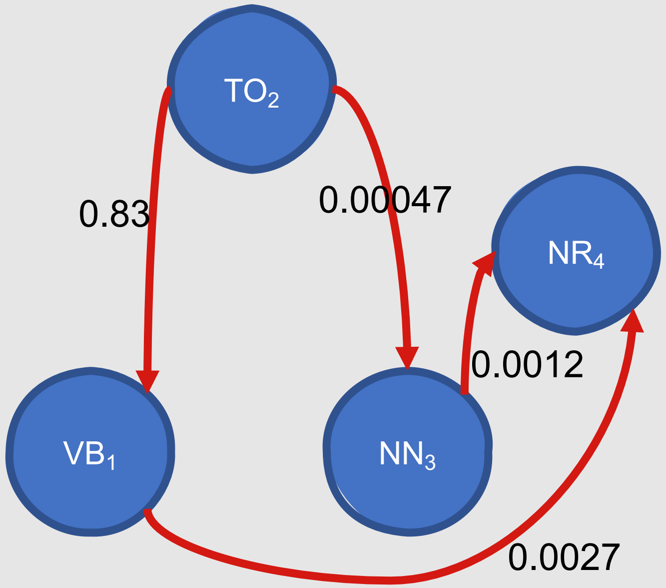


	race
VB	0.00012
NN	0.00057

- We have our transition probabilities …what now?
- Observation likelihoods!
- We can also compute these using frequency counts from the Brown Corpus
- $P(w_i|t_i) = \frac{c(w_i,t_i)}{c(t_i)}$
- Since we're trying to decide the best tag for "race," we need to compute both P(race|VB) and P(race|NN)
- P(race|VB) = C(race, VB) / C(VB) = 0.00012
- P(race|NN) = C(race, NN) / C(NN) = 0.00057

Example: Bigram HMM Tagger

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

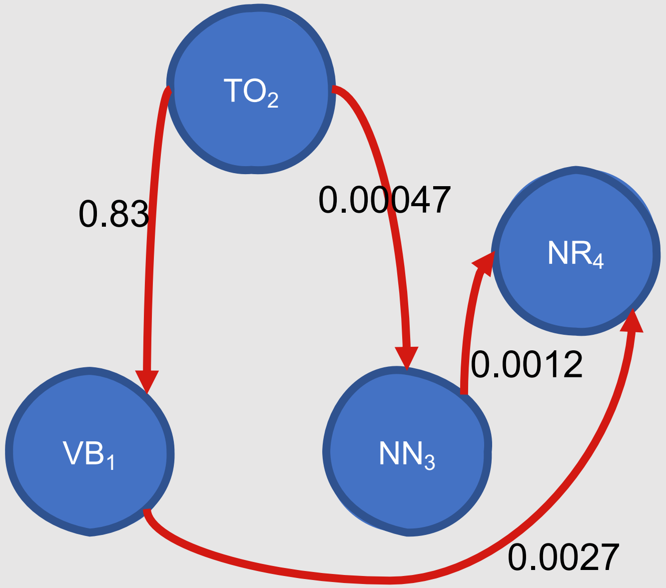


	race
VB	0.00012
NN	0.00057

- Now, to decide how to tag “race,” we can consider our two possible sequences:
    - to (TO) race (VB) tomorrow (NR)
    - to (TO) race (NN) tomorrow (NR)
- We will select the tag that maximizes the probability:
    - $P(t_i|TO)P(NR|t_i)P(race|t_i)$
- We determine that:
    - P(VB|TO)P(NR|VB)P(race|VB) = 0.83 * 0.0027 * 0.00012 = 0.00000027
    - P(NN|TO)P(NR|NN)P(race|NN) = 0.00047 * 0.0012 * 0.00057 = 0.00000000032

Example: Bigram HMM Tagger

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

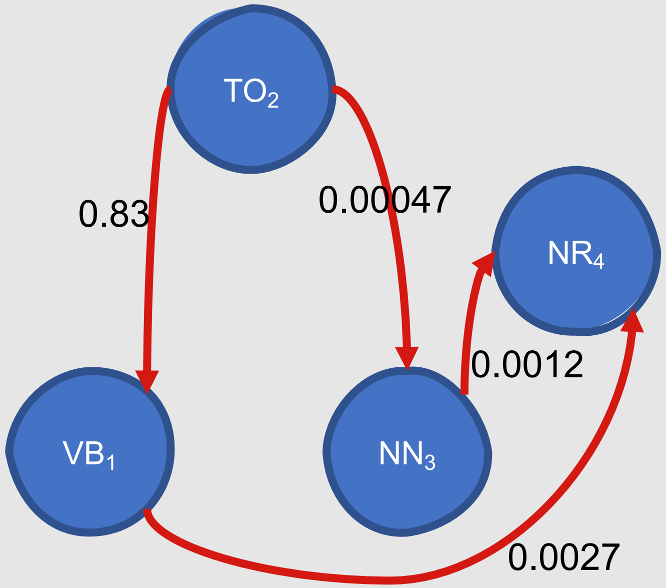


	race
VB	0.00012
NN	0.00057

- Now, to decide how to tag "race," we can consider our two possible sequences:
  - to (TO) race (VB) tomorrow (NR)
  - to (TO) race (NN) tomorrow (NR)
- We will select the tag that maximizes the probability:
  - $P(t_i|TO)P(NR|t_i)P(race|t_i)$
- We determine that:
  - **P(VB|TO)P(NR|VB)P(race|VB) = 0.83 * 0.0027 * 0.00012 = 0.00000027**
    - Optimal sequence!
  - P(NN|TO)P(NR|NN)P(race|NN) = 0.00047 * 0.0012 * 0.00057 = 0.00000000032

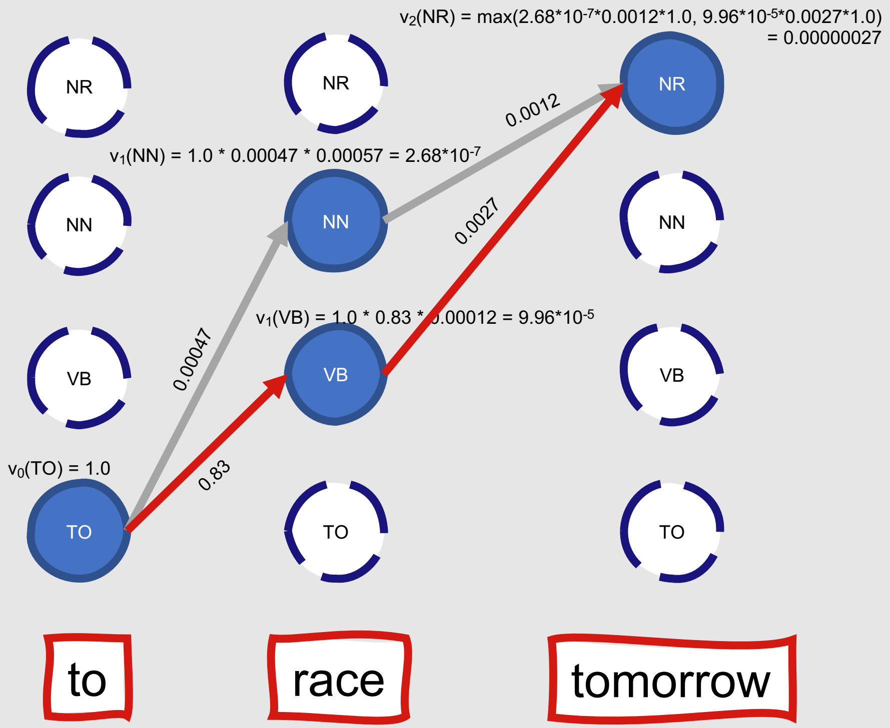
Example: Bigram HMM Tagger

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



	race
VB	0.00012
NN	0.00057

- Visualized in a Viterbi trellis, this would look like:



Example: Bigram HMM Tagger

What if we used greater values of *n*?

- For example, a trigram HMM tagger instead of a bigram HMM tagger?
- Generally, more context → more accurate predictions
- However, greater values of *n* also require more computational work …you need to determine whether the trade-off is worth it