HIDDEN MARKOV MODELS

FORMAL DEFINITION OF A FSA

- A finite automaton has
  - a finite set of N states *q0,q1,q2…q5*
  - a finite input alphabet of symbols *w,o,m,a,n*
  - a start state *q0*
  - a transition matrix or function

	w	o	m	a	n
q0	q1	-	q3	-	-
q1	-	q2	-	-	-
q2	-	-	q3	-	-
q3	-	-	-	q4	-
q4	-	-	-	-	q5
q5	-	-	-	-	-

FORMAL DEFINITION OF A HMM

- A Hidden Markov Model has
  - a finite set of N states *q0,q1,q2…q5*
  - a finite list of observations *[i, want, to, race]*
  - a start state q_0 and end state q_F with no observations.
- a transition ***probability*** matrix (A)
  - a_{ij} is the probability of transitioning from state i to j

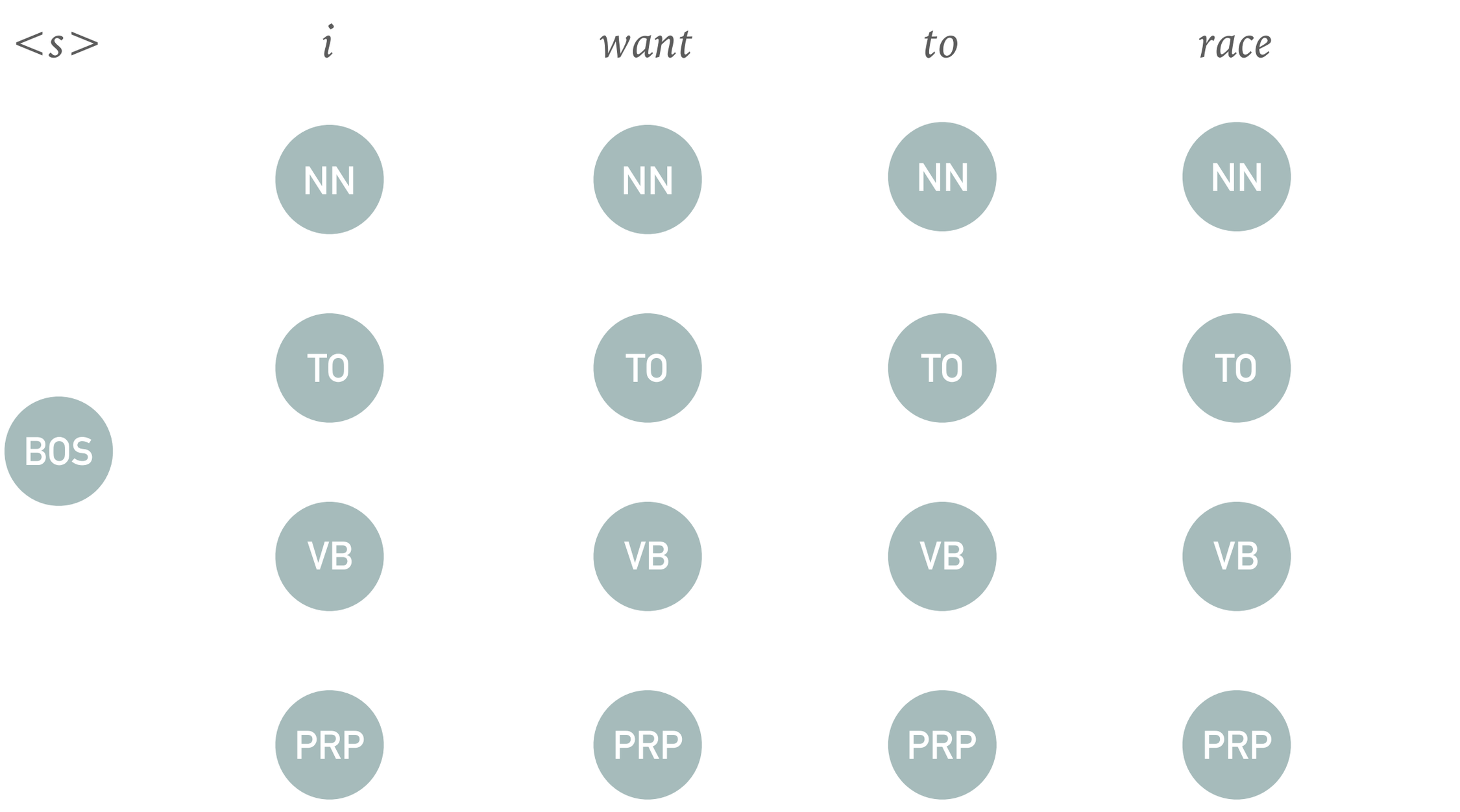
A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

FORMAL DEFINITION OF A HMM

- A Hidden Markov Model has
  - a finite set of N states *q0,q1,q2…q5*
  - a finite list of observations *[i, want, to, race]*
  - a start state q_0 and end state q_F with no observations.
- a transition ***probability*** matrix (A)
  - a_{ij} is the probability of transitioning from state i to j
- a matrix of observation likelihoods (B)

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

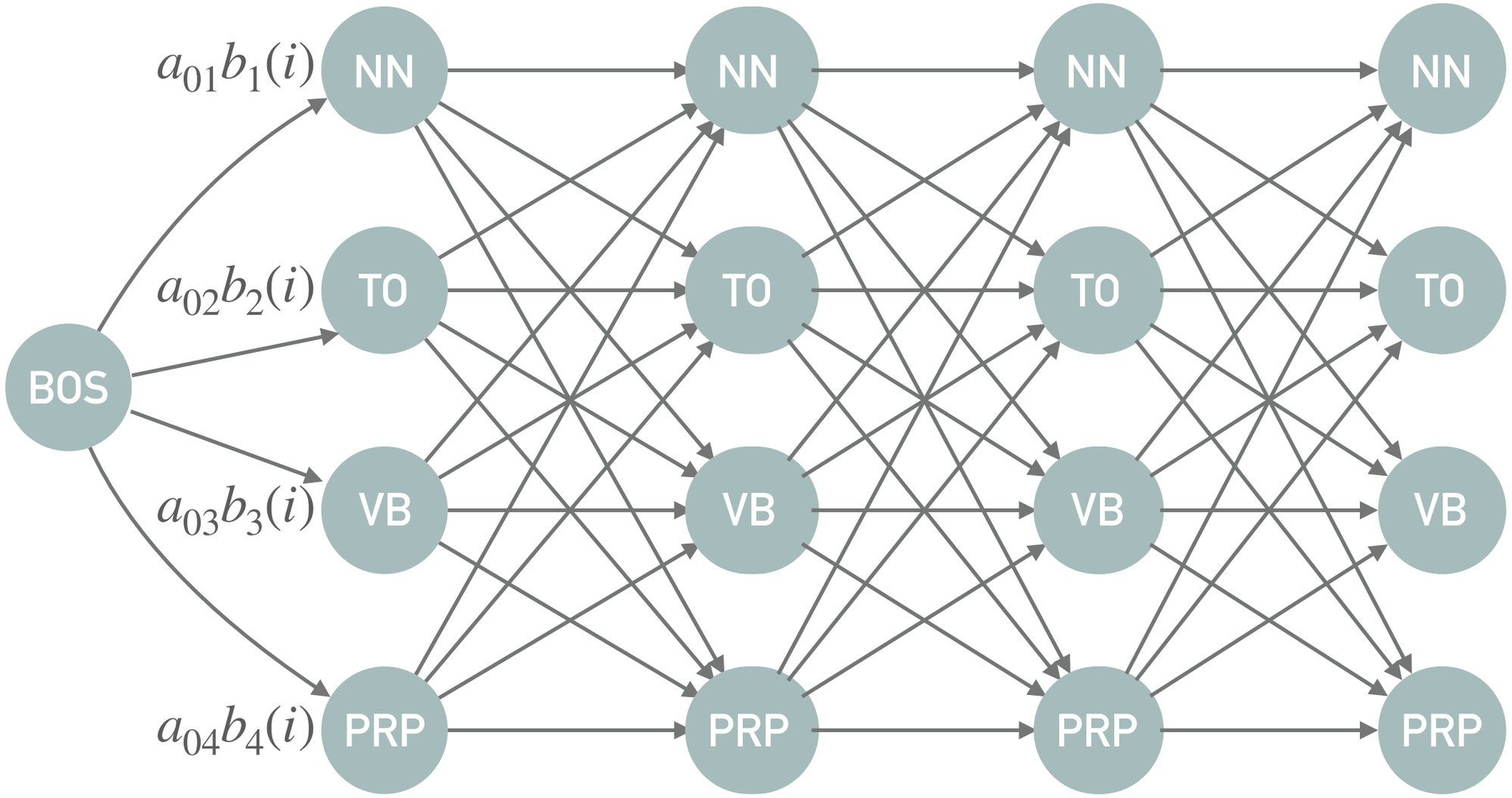
TRELLIS


<s>
i
want
to
race
NN
NN
NN
NN
TO
TO
TO
TO
BOS
VB
VB
VB
VB
PRP
PRP
PRP
PRP

TRELLIS

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

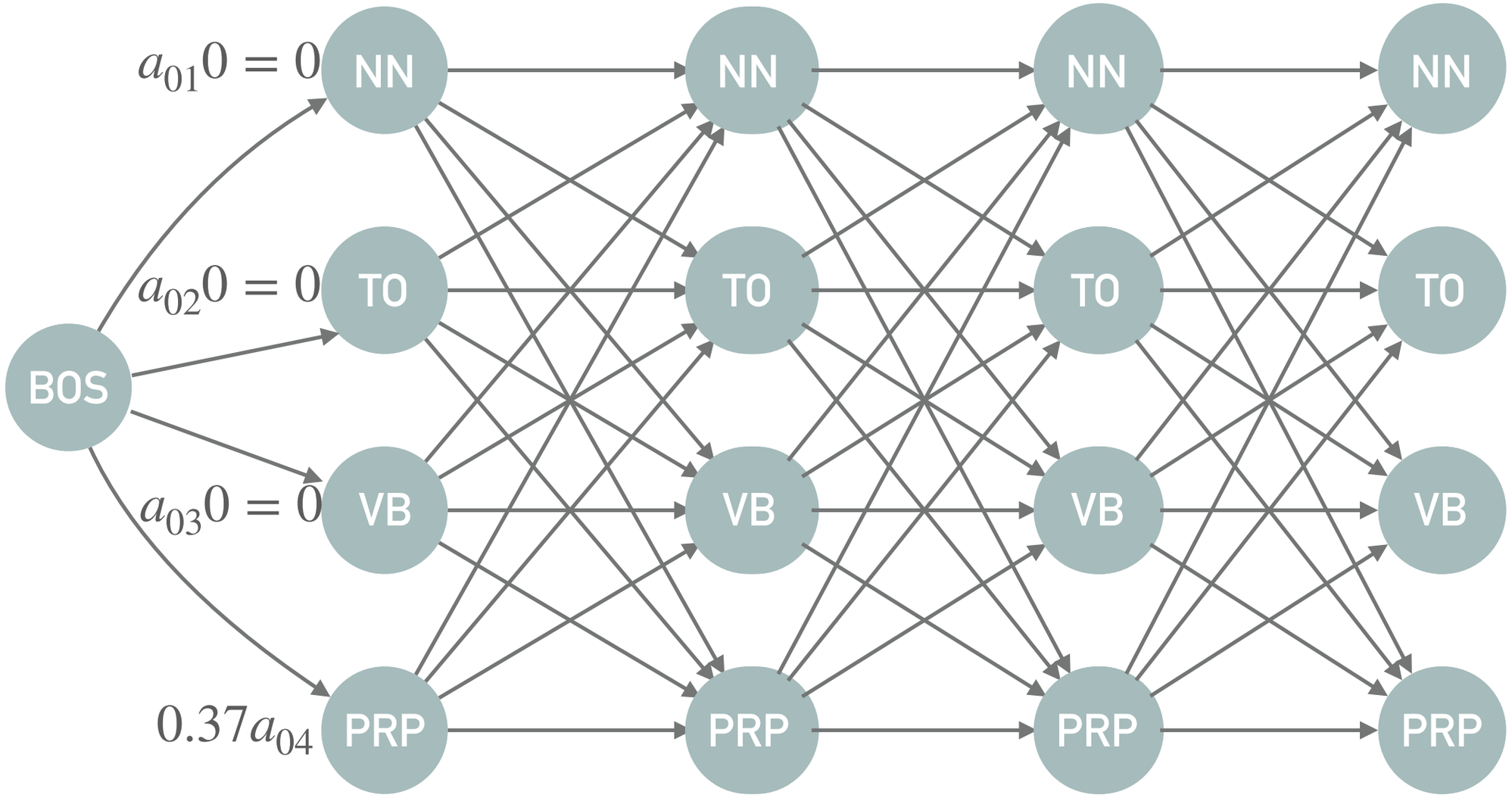
<s> *i* *want* *to* *race*



TRELLIS

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

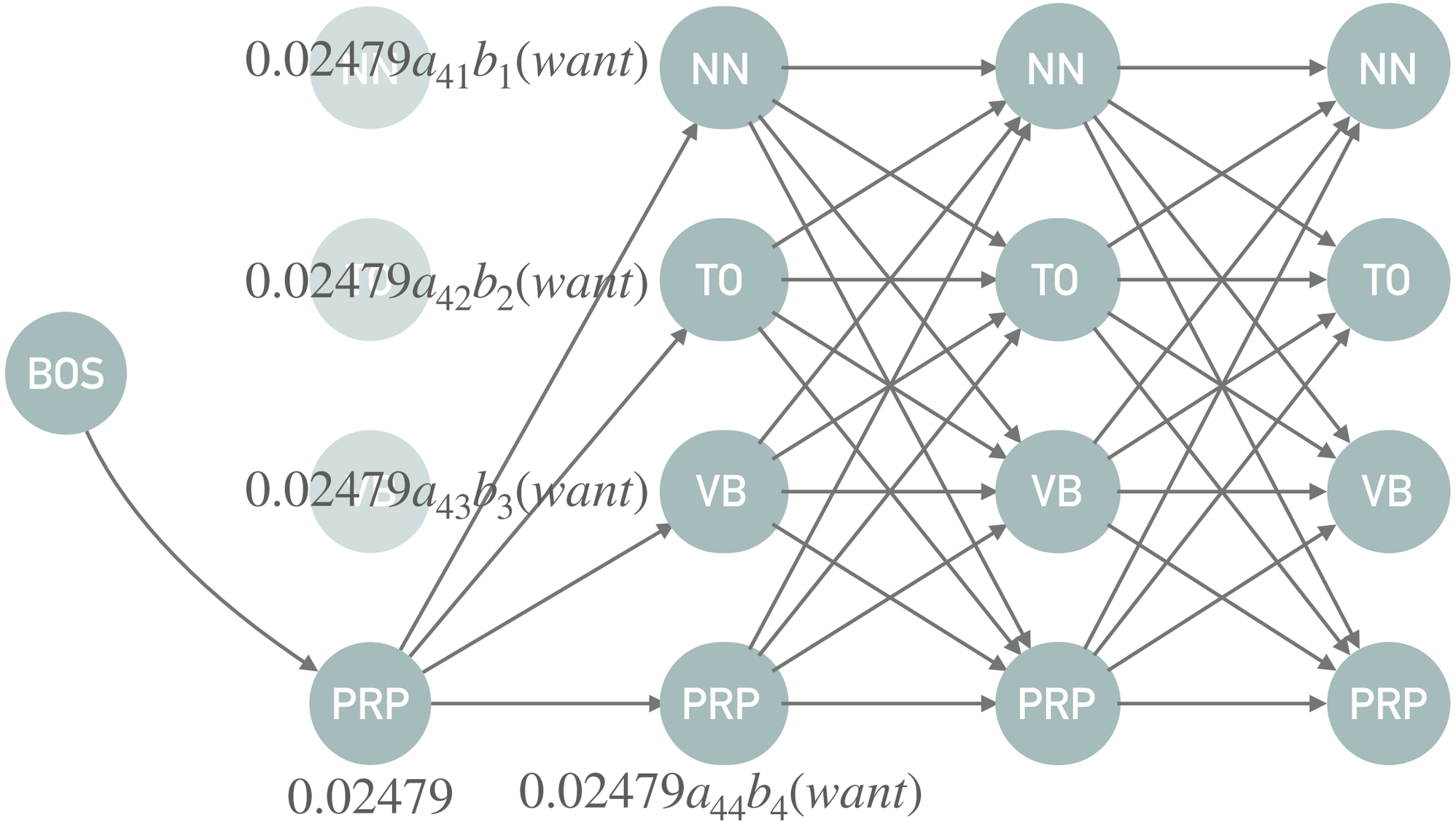
<s> *i* *want* *to* *race*



A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

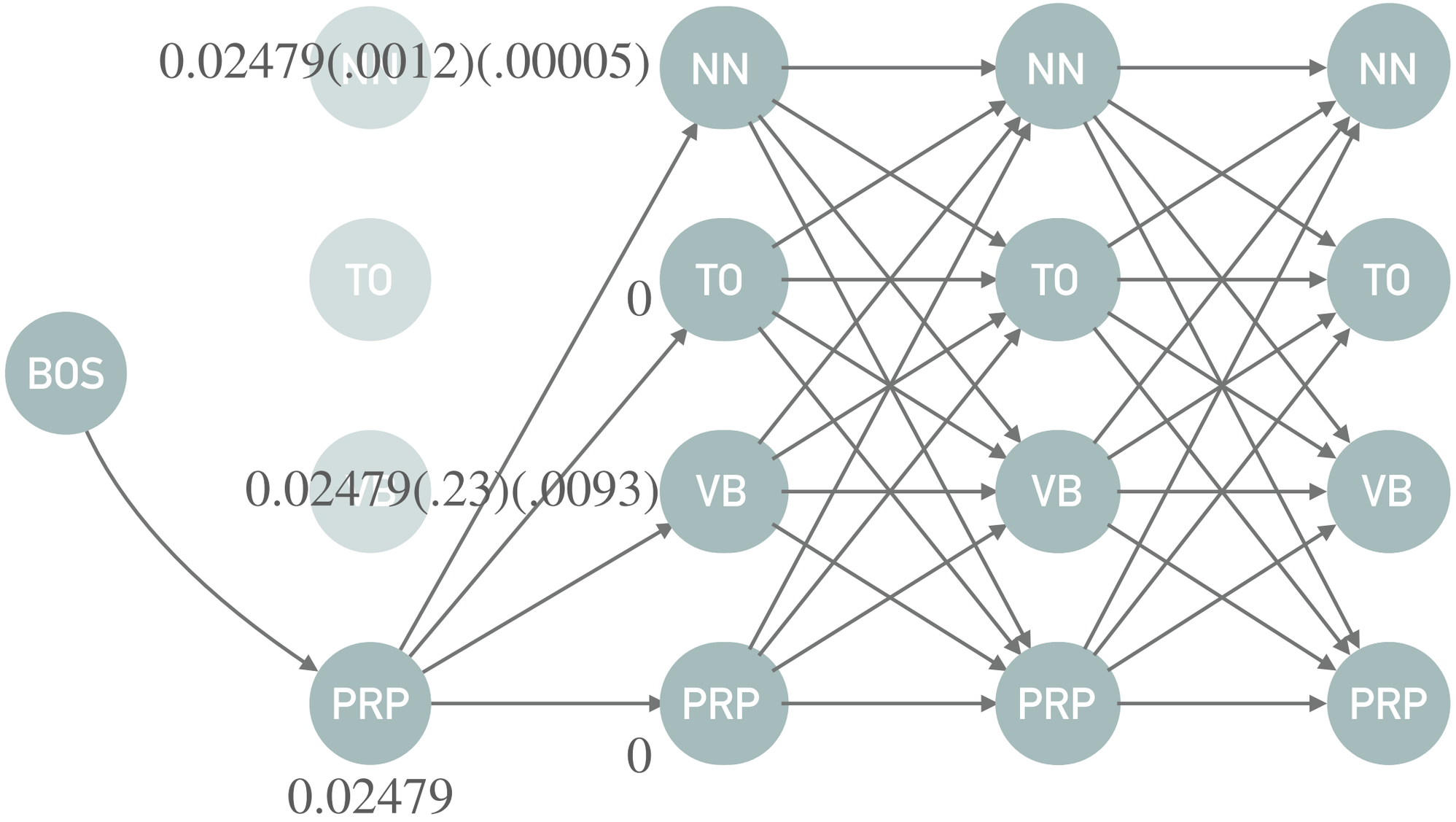
$<s>$ i $want$ to $race$



A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

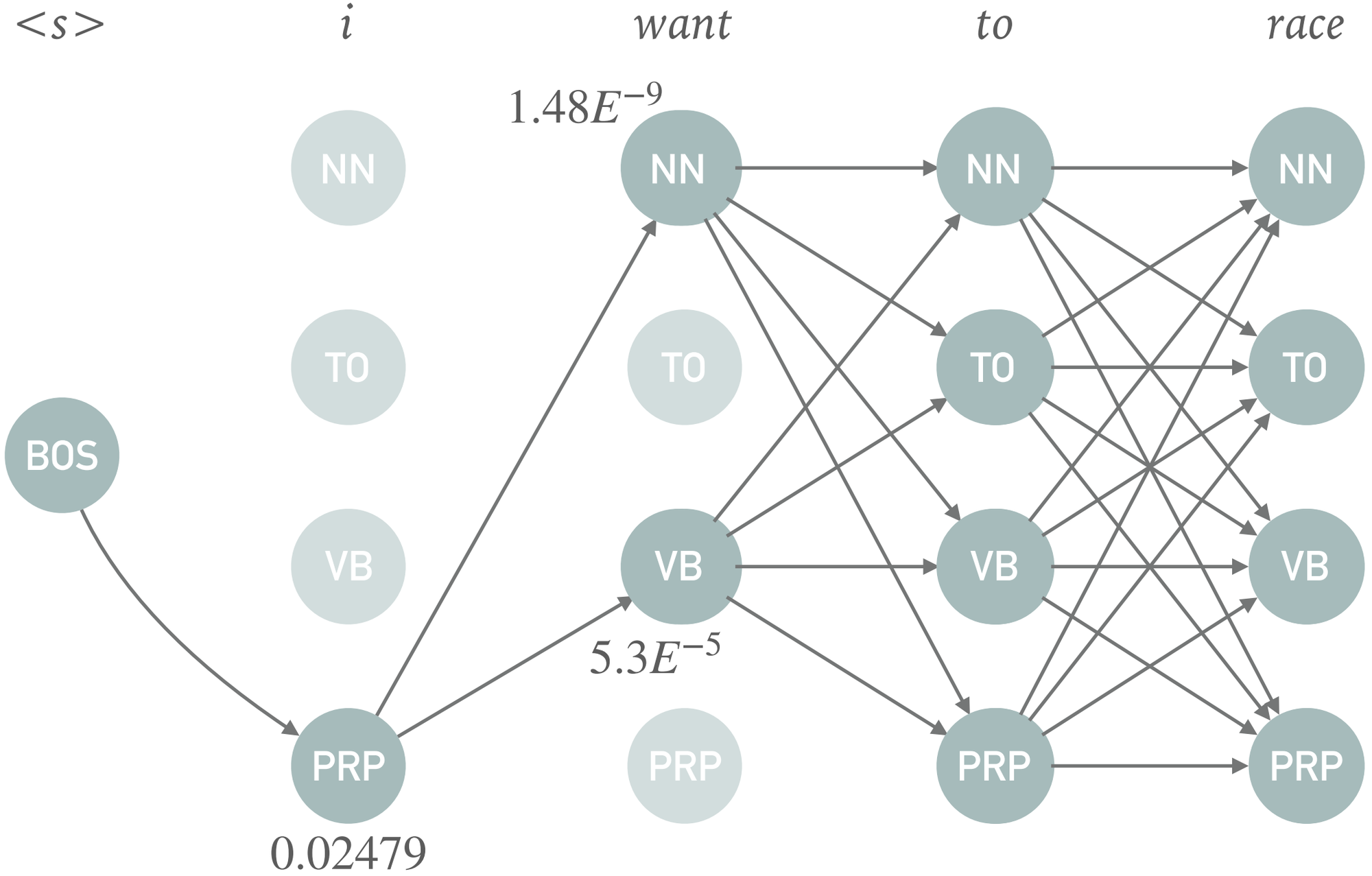
B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

<s> *i* *want* *to* *race*



A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0



A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

function Viterbi(observations, state-graph) -> best path

You'll want to store a 2D list for the best probability at each node of the trellis:

viterbi [# of states + 2] [length of observations list]

(Example: 4 states, 4 observations should make a 2D list of 6x4)

You'll also want a 2D list to represent which node of the previous column the best probability came from. This is called a "back pointer".

backpointer [#of states + 2][length of observations list]

viterbi	I	want	to	race
BOS				
NN				
TO				
VB				
PRP				
EOS				

backpointer	I	want	to	race
BOS				
NN				
TO				
VB				
PRP				
EOS				

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//initialize our first column of the trellis
for state_index from 1 to # of states:
    viterbi[ state_index ][ 1 ] = transition_probabilities[ 0 ][ state_index ]
                        * emission_probabilities[ state_index ][ observation[ 1 ] ]
    backpointer[ state_index ][ 1 ] = 0
```

viterbi	I	want	to	race
BOS				
NN				
TO				
VB				
PRP				
EOS				

backpointer	I	want	to	race
BOS				
NN				
TO				
VB				
PRP				
EOS				

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//initialize our first column of the trellis
for state_index from 1 to # of states:
    viterbi[ state_index ][ 1 ] = transition_probabilities[ 0 ][ state_index ]
                        * emission_probabilities[ state_index ][ observation[ 1 ] ]
    backpointer[ state_index ][ 1 ] = 0
```

viterbi	I	want	to	race
BOS				
NN	.041*0			
TO	0.0043*0			
VB	0.019*0			
PRP	0.067*.37			
EOS				

backpointer	I	want	to	race
BOS				
NN	0			
TO	0			
VB	0			
PRP	0			
EOS	0			

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//initialize our first column of the trellis
for state_index from 1 to # of states:
    viterbi[ state_index ][ 1 ] = transition_probabilities[ 0 ][ state_index ]
                        * emission_probabilities[ state_index ][ observation[ 1 ] ]
    backpointer[ state_index ][ 1 ] = 0
```

viterbi	I	want	to	race
BOS				
NN	0			
TO	0			
VB	0			
PRP	0.02479			
EOS				

backpointer	I	want	to	race
BOS				
NN	0			
TO	0			
VB	0			
PRP	0			
EOS	0			

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//loop through the rest of the trellis
for observation from 2 to # of observations:
    for curr_state from 1 to # of states:
        viterbi[ curr_state ][ observation ]
                = max of viterbi[prev_state][prev_observation]
                        * transition_probabilities[prev_state][curr_state]
                        * emission_probabilities[curr_state][ observation ]
                over all prev_states
```

viterbi	I	want	to	race
BOS				
NN	0			
TO	0			
VB	0			
PRP	0.02479			
EOS				

backpointer	I	want	to	race
BOS				
NN	0			
TO	0			
VB	0			
PRP	0			
EOS	0			

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//loop through the rest of the trellis
for observation from 2 to # of observations:
    for curr_state from 1 to # of states:
        viterbi[ curr_state ][ observation ]
                = max of viterbi[prev_state][prev_observation]
                        * transition_probabilities[prev_state][curr_state]
                        * emission_probabilities[curr_state][ observation ]
                over all prev_states
```

viterbi	I	want	to	race
BOS				
NN	0			
TO	0			
VB	0			
PRP	0.02479			
EOS				

max of
NN-> 0*0.087*0.00005 = 0
TO-> 0*0.016*0.00005 = 0
VB-> 0*0.004*0.00005 = 0
PRP-> 0.025*0.0045*0.0005 = 5.5E-8

backpointer	I	want	to	race
BOS				
NN	0			
TO	0			
VB	0			
PRP	0			
EOS	0			

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//loop through the rest of the trellis
for observation from 2 to # of observations:
    for curr_state from 1 to # of states:
        viterbi[ curr_state ][ observation ]
                = max of viterbi[prev_state][prev_observation]
                        * transition_probabilities[prev_state][curr_state]
                        * emission_probabilities[curr_state][ observation ]
                over all prev_states
```

viterbi	I	want	to	race
BOS				
NN	0	5.5E-08		
TO	0			
VB	0			
PRP	0.02479			
EOS				

backpointer	I	want	to	race
BOS				
NN				
TO				
VB				
PRP				
EOS	0			

max of
NN-> 0*0.087*0.00005 = 0
TO-> 0*0.016*0.00005 = 0
VB-> 0*0.004*0.00005 = 0
PRP-> 0.025*0.0045*0.0005 = 5.5E-8

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//loop through the rest of the trellis
for observation from 2 to # of observations:
    for curr_state from 1 to # of states:
        backpointer[ curr_state ][ observation ]
                = argmax of viterbi[prev_state][prev_observation]
                        * transition_probabilities[prev_state][curr_state]
```

viterbi	I	want	to	race
BOS				
NN	0	5.5E-08		
TO	0			
VB	0			
PRP	0.02479			
EOS				

backpointer	I	want	to	race
BOS				
EOS	0			

argmax of
NN-> 0*0.087*0.00005 = 0
TO-> 0*0.016*0.00005 = 0
VB-> 0*0.004*0.00005 = 0
PRP-> 0.025*0.0045*0.0005 = 5.5E-8

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//loop through the rest of the trellis
for observation from 2 to # of observations:
    for curr_state from 1 to # of states:
        backpointer[ curr_state ][ observation ]
                = argmax of viterbi[prev_state][prev_observation]
                        * transition_probabilities[prev_state][curr_state]
```

viterbi	I	want	to	race
BOS				
NN	0	5.5E-08		
TO	0			
VB	0			
PRP	0.02479			
EOS				

backpointer	I	want	to	race
BOS				
NN				
TO				
VB				
PRP				
EOS	0			

argmax of
NN-> 0*0.087*0.00005 = 0
TO-> 0*0.016*0.00005 = 0
VB-> 0*0.004*0.00005 = 0
PRP-> 0.025*0.0045*0.0005 = 5.5E-8

A	NN	TO	VB	PRP
BOS	0.041	0.0043	0.019	0.067
NN	0.087	0.016	0.004	0.0045
TO	0.00047	0	0.83	0
VB	0.047	0.035	0.0038	0.007
PRP	0.0012	0.00079	0.23	0.00014

B	I	want	to	race
NN	0	0.00005	0	0.00057
TO	0	0	0.99	0
VB	0	0.0093	0	0.00012
PRP	0.37	0	0	0

```
//loop through the rest of the trellis
for observation from 2 to # of observations:
    for curr_state from 1 to # of states:
        backpointer[ curr_state ][ observation ]
                = argmax of viterbi[prev_state][prev_observation]
                        * transition_probabilities[prev_state][curr_state]
```

viterbi	I	want	to	race	</s>
NN	0	1.4E-09	0	very smaller	0
TO	0	0		0	0
VB	0	5.3E-05	0	very small	0
PRP	0.02479	0	0	0	0
EOS	0	0	0	0	best prob

backpointer	I	want	to	race	
NN	-	PRP	-	TO	-
TO	-	-	VB	-	-
VB	-	PRP	-	TO	-
PRP	BOS	-	-	-	-
EOS	-	-	-	-	VB

```
//last column of trellis
viterbi[-1][-1] = max of viterbi[prev_state][time-1]
                        * transition_probabilities[prev_state][end_state]
backpointer[-1][-1] = argmax of viterbi[prev_state][time-1]
                        * transition_probabilities[prev_state][end_state]

return backtrace of backpointer path
```