

Viterbi Algorithm

CSCI-GA.2590 – Natural Language Processing

Ralph Grishman

Computing Probabilities

$$\begin{aligned} \mathit{viterbi} [s, t] = & \\ & \max_{s'} \left(\mathit{viterbi} [s', t-1] \times \right. \\ & \qquad \qquad \qquad \text{transition probability} \\ & \qquad \qquad \qquad P(s \mid s') \times \\ & \qquad \qquad \qquad \text{emission probability} \\ & \qquad \qquad \qquad \left. P(\text{token}[t] \mid s) \right) \end{aligned}$$

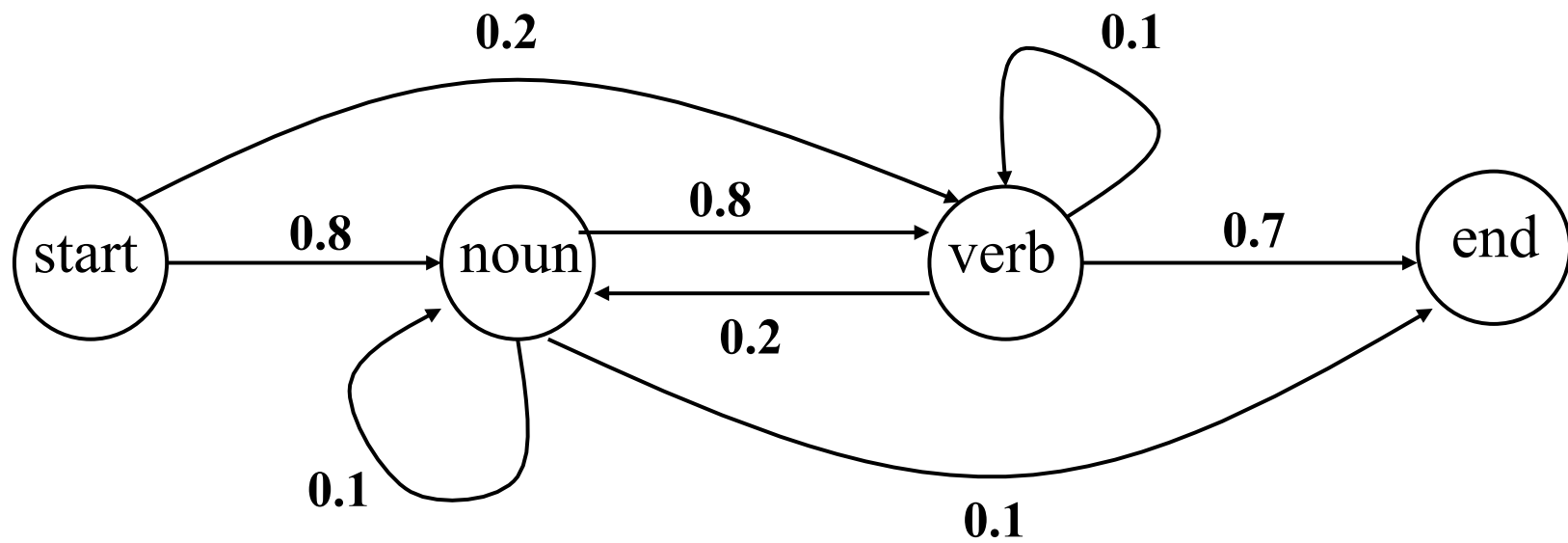
for each s, t :

record which $s', t-1$ contributed the maximum

Analyzing

Fish sleep.

A Simple POS HMM



Word Emission Probabilities

$$P(\text{word} \mid \text{state})$$

- A two-word language: “fish” and “sleep”
- Suppose in our training corpus,
 - “fish” appears 8 times as a noun and 5 times as a verb
 - “sleep” appears twice as a noun and 5 times as a verb
- Emission probabilities:
 - Noun
 - $P(\text{fish} \mid \text{noun}) : 0.8$
 - $P(\text{sleep} \mid \text{noun}) : 0.2$
 - Verb
 - $P(\text{fish} \mid \text{verb}) : 0.5$
 - $P(\text{sleep} \mid \text{verb}) : 0.5$

Viterbi Probabilities

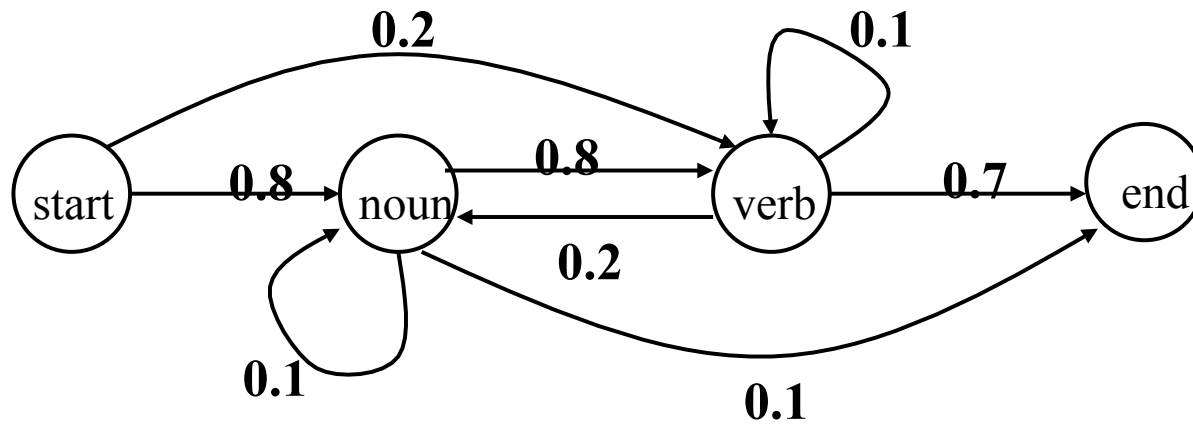
0 1 2 3

start

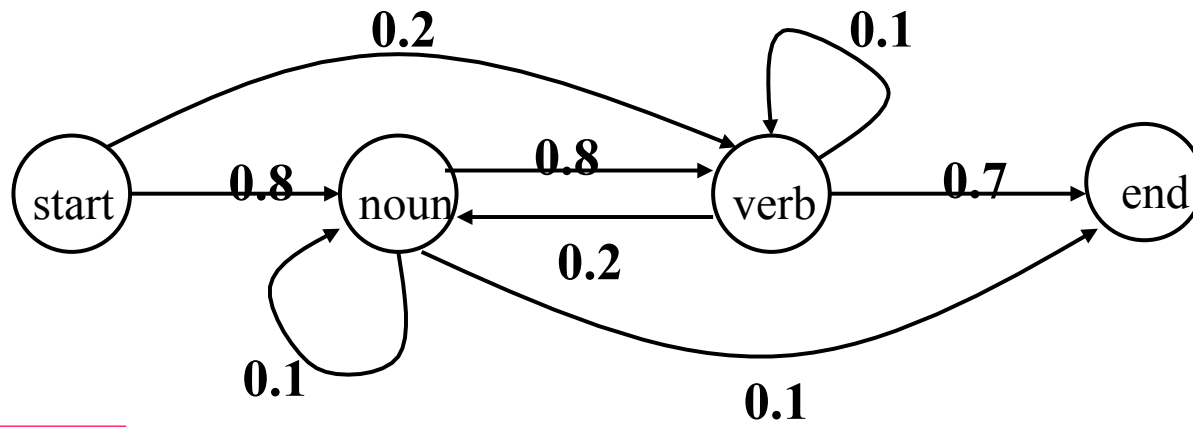
verb

noun

end

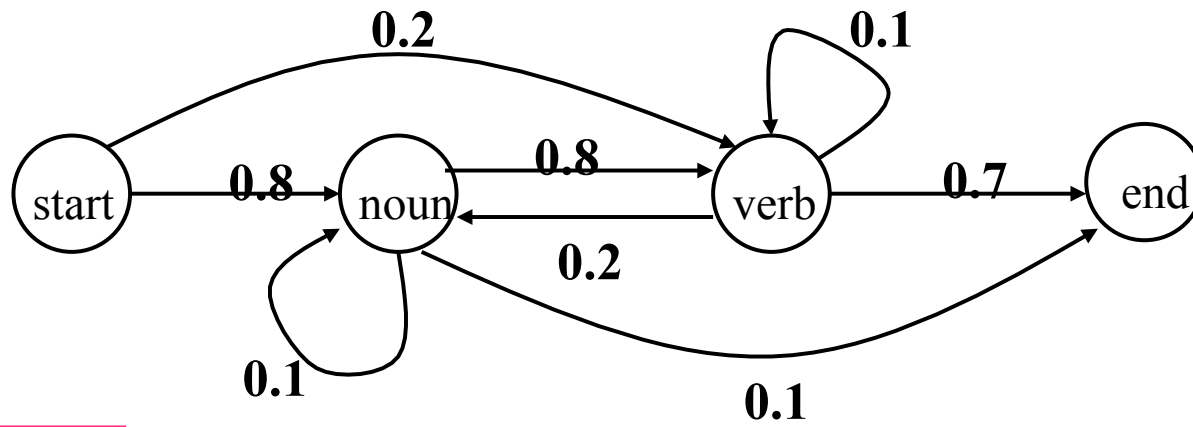


	0	1	2	3
start	1			
verb	0			
noun	0			
end	0			



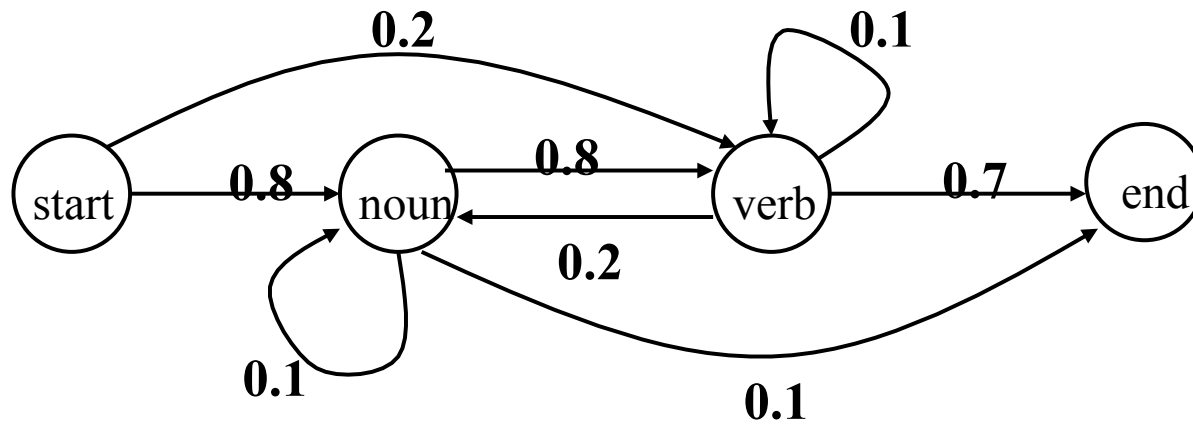
Token 1: fish

	0	1	2	3
start	1	0		
verb	0	.2 * .5		
noun	0	.8 * .8		
end	0	0		



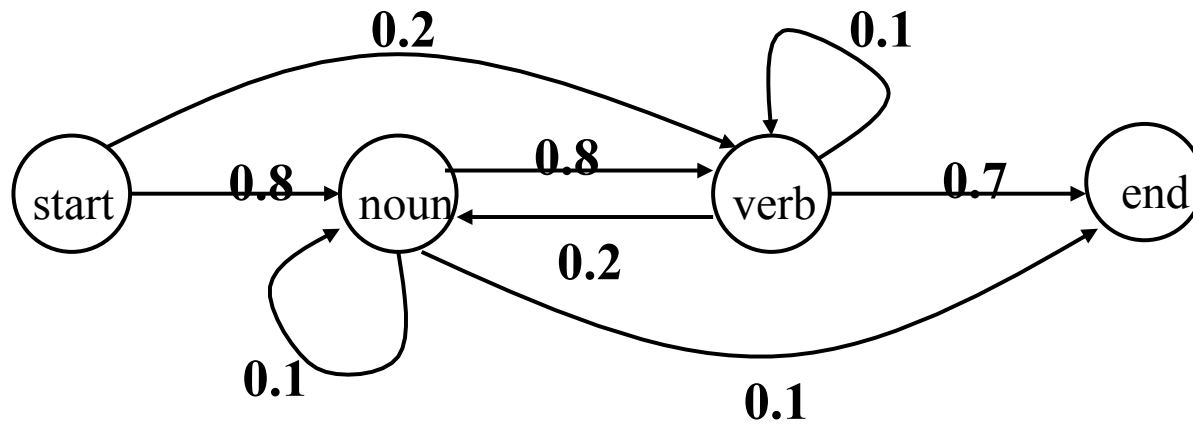
Token 1: fish

	0	1	2	3
start	1	0		
verb	0	.1		
noun	0	.64		
end	0	0		



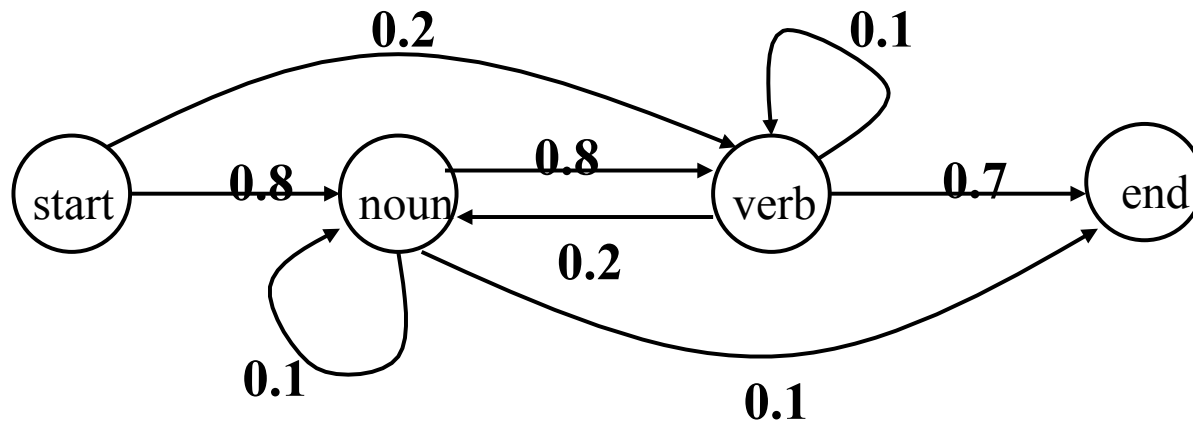
Token 2: sleep
(if 'fish' is verb)

	0	1	2	3
start	1	0	0	
verb	0	.1	.1*.1*.5	
noun	0	.64	.1*.2*.2	
end	0	0	-	



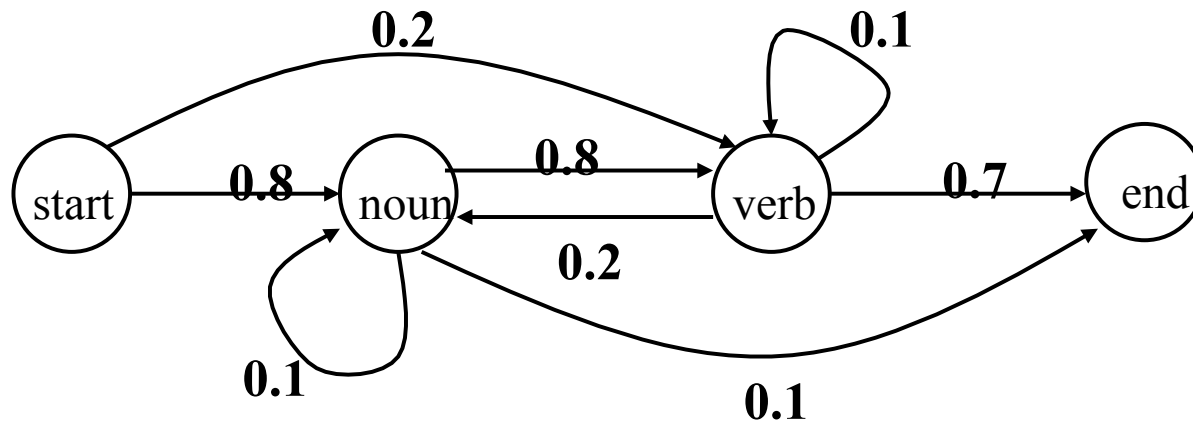
Token 2: sleep
(if 'fish' is verb)

	0	1	2	3
start	1	0	0	
verb	0	.1	.005	
noun	0	.64	.004	
end	0	0	-	



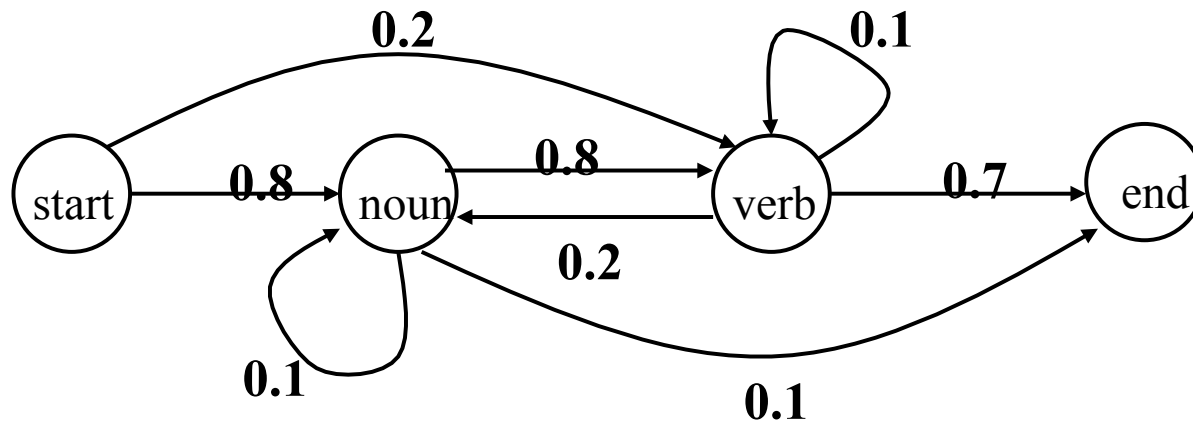
Token 2: sleep
(if 'fish' is a noun)

	0	1	2	3
start	1	0	0	
verb	0	.1	.005 $.64 * .8 * .5$	
noun	0	$.64$.004 $.64 * .1 * .2$	
end	0	0	-	



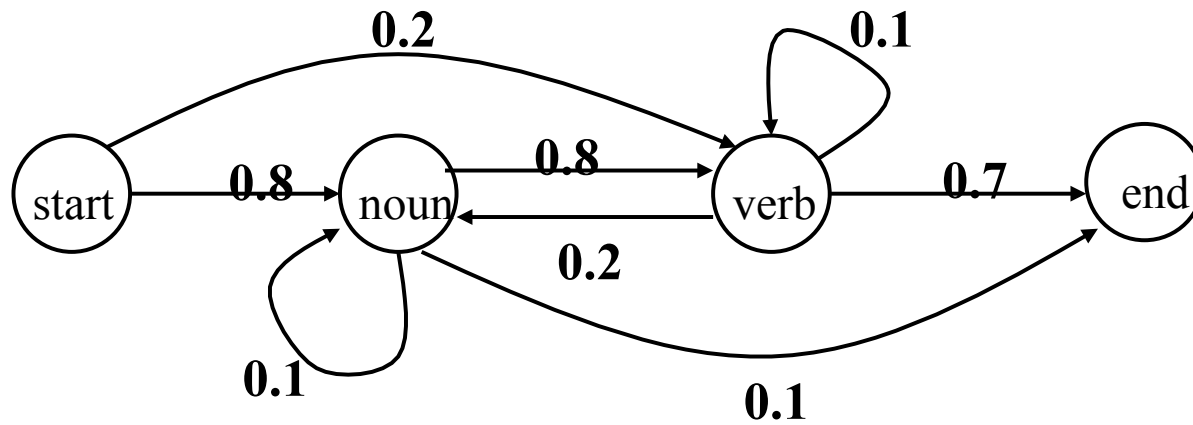
Token 2: sleep
(if 'fish' is a noun)

	0	1	2	3
start	1	0	0	
verb	0	.1	.005	.256
noun	0	.64	.004	.0128
end	0	0	-	



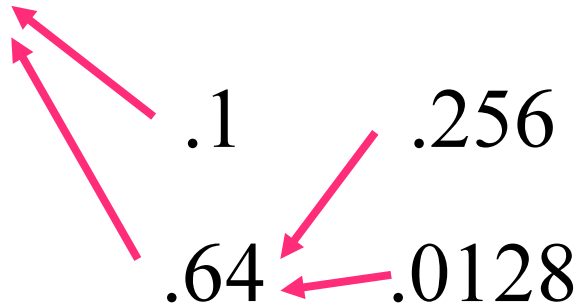
Token 2: sleep
 take maximum,
 set back pointers

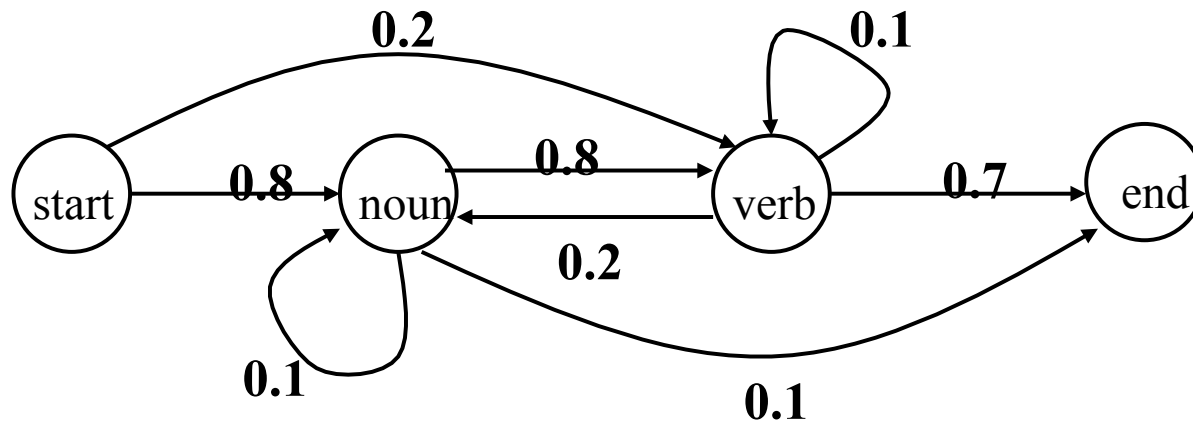
	0	1	2	3
start	1	0	0	
verb	0	.1	.005	
noun	0	.64	.004	.256
end	0	0	-	.0128



Token 2: sleep
 take maximum,
 set back pointers

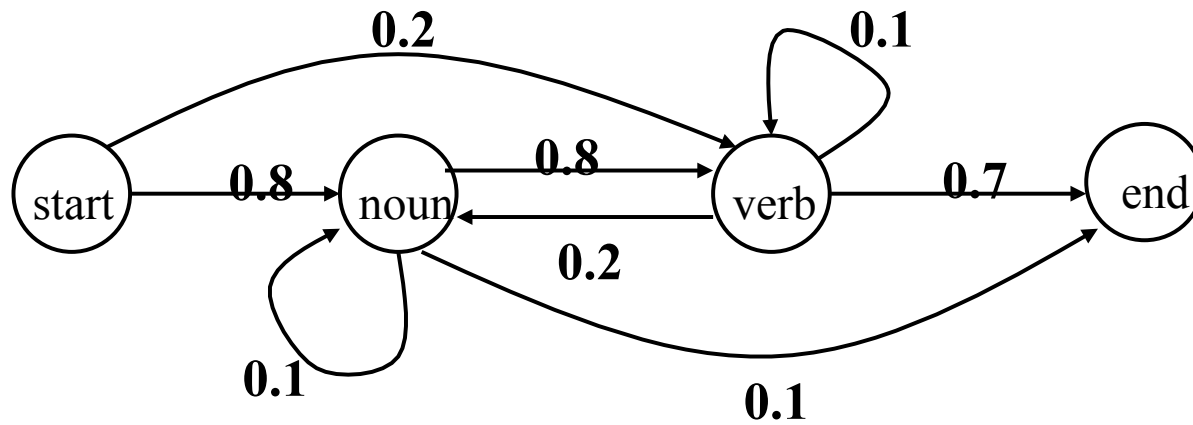
	0	1	2	3
start	1	0	0	
verb	0	.1	.256	
noun	0	.64	.0128	
end	0	0	-	





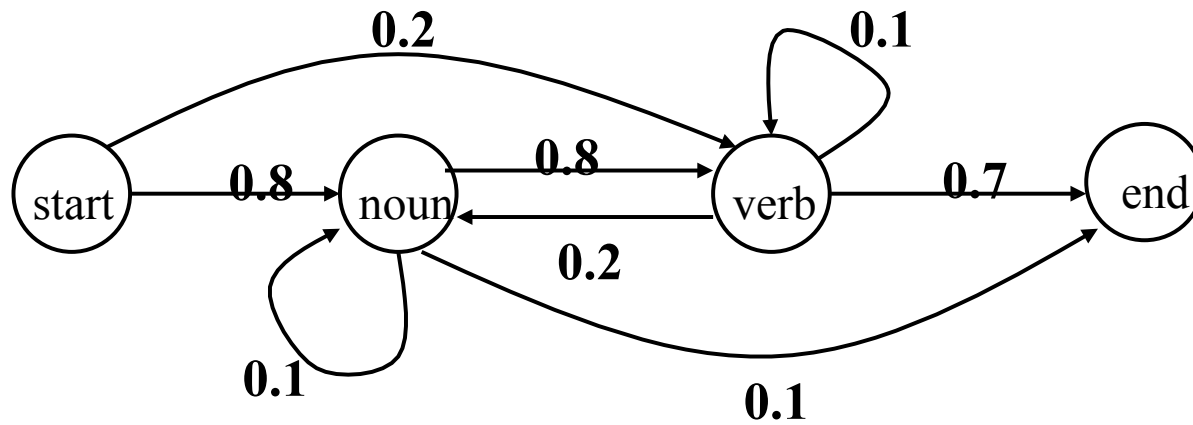
Token 3: end

	0	1	2	3
start	1	0	0	0
verb	0	.1	.256	-
noun	0	.64	.0128	-
end	0	0	-	.256*.7 .0128*.1



Token 3: end
 take maximum,
 set back pointers

	0	1	2	3
start	1	0	0	0
verb	0	.1	.256	-
noun	0	.64	.0128	-
end	0	0	-	.256*.7 .0128*.1



Decode:
 fish = noun
 sleep = verb

	0	1	2	3
start	1	0	0	0
verb	0	.1	.256	-
noun	0	.64	.0128	-
end	0	0	-	.256*.7

Complexity?

- How does time for Viterbi search depend on number of states and number of words?

Complexity

$$\text{time} = O (s^2 n)$$

for s states and n words

(Relatively fast: for 40 states and 20 words,
32,000 steps)