CSCI 544 – Applied Natural Language Processing, Spring 2021

Written Homework 3: Hidden Markov Models

OUT: February 8, 2021

Total: 18 pages.

General instructions

1. This is not a graded assignment. Do not turn it in.
2. The assignment is meant as preparation for the in-class exams. You should aim to answer all the questions on your own, without help.
3. Space is provided as it would be on the exams. Answers should be concise and fit in the space provided; in the exams it will not be possible to add space, and long and rambling answers will be penalized.
4. After solving the problems (or giving them your best try), you are encouraged to discuss and compare solutions with your classmates.
5. You are welcome to discuss the problems with us. We encourage open discussion on Piazza, so that the entire class can benefit from the discussion.
6. Answers to select problems will be distributed at a later time.

For further background on hidden Markov models, you may consult the following **optional** readings from Jurafsky and Martin, *Speech and Language Processing* (3rd edition draft):

Chapter 8: Sequence Labeling for Parts of Speech and Named Entities

Chapter A: Hidden Markov Models

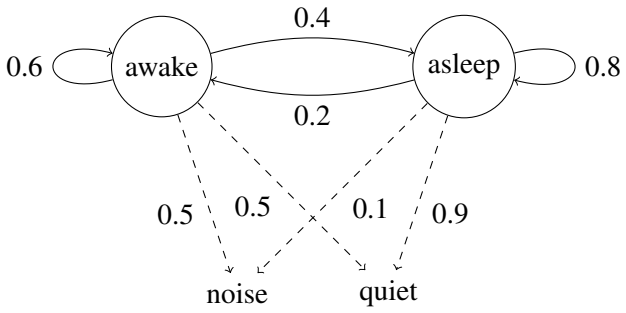
For the purpose of this assignment, please use the definitions given in the assignment, not the ones in the above chapters or any other source.

Markov chains. A Markov chain is a state machine which consists of the following:

1. A set of states $Q = \{q_1, \ldots, q_n\}$.
2. A transition probability matrix A, where each a_{ij} represents the probability of transitioning from state q_i to state q_j, such that for each i, $\sum_{j=1}^{n} a_{ij} = 1$.
$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$
3. A set of possible observations $V = \{v_1, \ldots, v_m\}$.
4. An emission probability matrix B, where each b_{ij} represents the probability of state q_i emitting the observation v_j, such that for each i, $\sum_{j=1}^{m} b_{ij} = 1$.
$$B = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}$$
5. A special start state q_0 which is not associated with observations, along with transition probabilities $a_{01} \cdots a_{0n}$ from the start state to the other states. The start state may be identical to one of the other states.

A Markov chain starts at the start state q_0, and at each time point $t_1, t_2 \ldots$ performs a transition and emits an observation. The *Markov property* states that the probability of being in a particular state at time t_i depends only on the previous state (that is, the state at t_{i-1}), and the probability of an observation at time t_i depends only on the current state (that is, the state at t_i).

Problem 1. Consider a Markov chain that represents the probability that a child left alone in her room will be awake or asleep. There are two states $\{\text{awake}, \text{asleep}\}$, and two possible observations coming from the room $\{\text{noise}, \text{quiet}\}$. The transition and emission probabilities are noted in the following diagram: transitions are shown with solid arrows, and emissions with dashed arrows. (Note that the diagram is identical to the one discussed in class, but the probabilities are different!)



The child starts by being awake, and remains in the room for 4 time points, $t_1 \ldots t_4$ (4 iterations of the Markov chain).

a. What is the most probable sequence of states for $t_1 \ldots t_4$?

b. What is the probability of the above sequence of states?

c. What is the most probable sequence of states and observations?

d. What is the probability of the above sequence of states and observations?

e. What is the least probable sequence of states?

f. What is the probability of the above sequence of states?

g. What is the least probable sequence of states and observations?

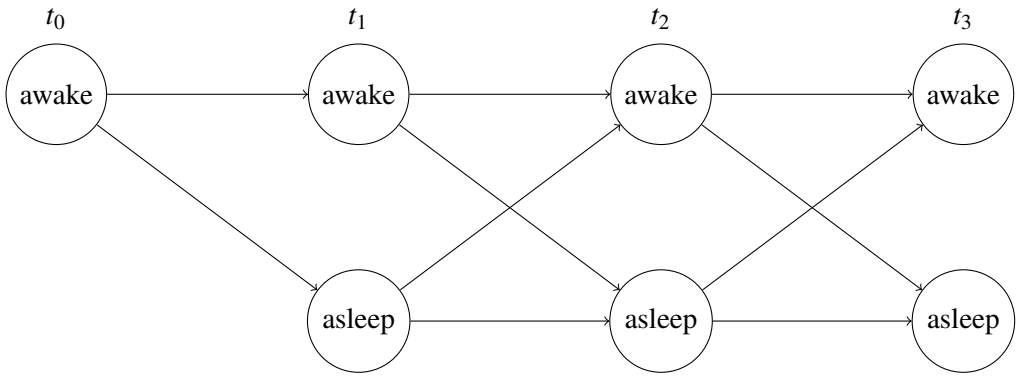
h. What is the probability of the above sequence of states and observations?

The Viterbi algorithm. A Hidden Markov Model (HMM) is a Markov chain where we cannot observe the states directly, but we can observe the emissions. The Viterbi algorithm is used for decoding the sequence of states, that is finding the most likely sequence of states that could give rise to a sequence of observations. Given a set of states Q and a sequence of time points $1 \dots T$, the algorithm builds two matrices of size $Q \times (1 \dots T)$: a probability matrix representing the probability of each state at each time point, and a backpointer matrix which points from each state at each time point to the most likely previous state. At the final time point T, the algorithm selects the state with the highest probability, and returns the path of backpointers from that state, representing the most likely sequence of states to give rise to the observations. The following is pseudocode for the algorithm: the notation $a(q', q)$ represents the transition probability between states q' and q, and $b(q, o_t)$ represents the emission probability by state q of the observation noted at time t.

 # Initialization step at $t = 1$
for q **in** Q :
    $\text{probability}(q, 1) = a(q_0, q) * b(q, o_1)$
    $\text{backpointer}(q, 1) = q_0$
 # Recursion step for the remaining time points
for t **from** 2 **to** T :
    for q **in** Q :
        $\text{probability}(q,t) = \max_{q' \in Q} \text{probability}(q', t-1) * a(q', q) * b(q, o_t)$
        $\text{backpointer}(q,t) = \arg\max_{q' \in Q} \text{probability}(q', t-1) * a(q', q)$
 # Termination step
$\text{most\_probable\_state}(T) = \arg\max_{q' \in Q} \text{probability}(q', T)$
return the backtrace path by following the backpointers from the most probable state

Problem 2. Consider the same Markov chain from problem 1, this time as a hidden Markov model. The child starts by being awake, and remains in the room for 3 time points, $t_1 \dots t_3$ (3 iterations of the Markov chain). The observations are: quiet, quiet, noise.

a. Using the Viterbi algorithm, identify the most likely sequence of states that would give rise to these observations. Show your work.



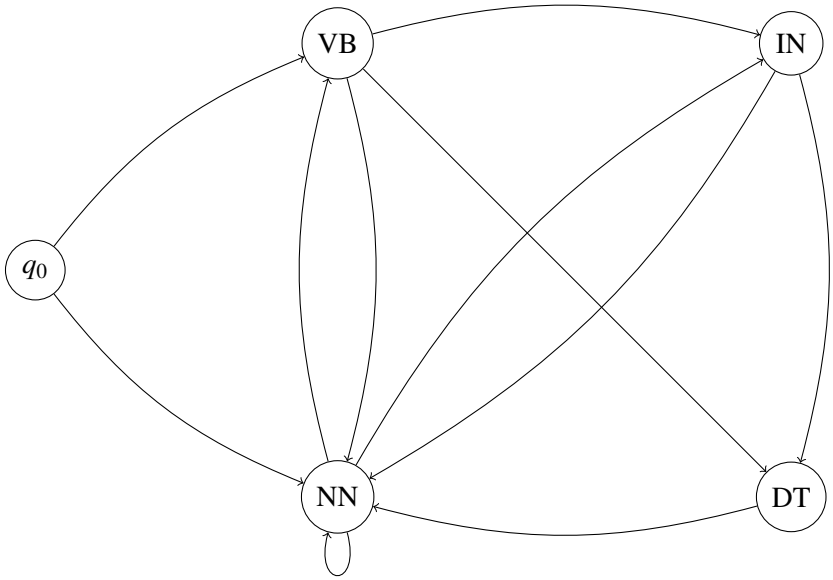
b. After the first iteration (at t_1), the sum of the probabilities is less than 1. Why? Where is the rest of the probability mass?

c. Suppose we are not interested in decoding the sequence of states (that is, whether the child was awake or asleep at each point), but only in the overall most likely state at the end (that is, whether the child is awake or asleep at the end). Obviously we can remove the backpointer lines from the Viterbi algorithm; however, this would still give us the probability of only the most likely path to each end state. What additional change can we make to the algorithm so that instead of giving us the probability of the most likely path to each state at each time, it will give the overall probability of being at each state at each time? Explain why.

Problem 3. In part-of-speech tagging, we learn a hidden Markov model from a tagged corpus: each state is a part-of-speech tag, transition probabilities are the conditional probabilities of tags given the previous tag, observations are words, and emission probabilities are the conditional probabilities of words given tags. The start state is the beginning of a sentence, which is not a part-of-speech tag. In this problem we will look at some data that will help us tag the sentence *Time flies like an arrow.* We will use the following sentences as a corpus of training data (the notation word/TAG means word tagged with a specific part-of-speech tag).

eat/VB breakfast/NN at/IN morning/NN time/NN
take/VB time/NN with/IN arrow/NN projects/NN
horse/NN riders/NN like/VB the/DT airport/NN
paper/NN flies/VB on/IN hydrogen/NN gas/NN
bees/NN sting/VB like/IN some/DT flies/NN
beans/NN soil/VB an/DT iron/NN grill/NN
flies/NN smell/VB an/DT arrow/NN drink/NN
people/NN like/VB an/DT army/NN arrow/NN
dinner/NN time/NN flies/VB all/DT day/NN
horse/NN flies/NN time/VB morning/NN rays/NN

a. Based on the corpus, fill in the transition probabilities in the state chart below.



b. Fill in the emission probabilities for the 4 states; only write down the probabilities for the words *time flies like an arrow*.

	time	flies	like	an	arrow
VB					
NN					
IN					
DT					

c. Now use the Viterbi algorithm with the above model to tag the sentence *Time flies like an arrow*. What is the most likely tag sequence, based on the training data? Show your work.

Problem 4. You are building a system to tag utterances in English and Spanish. You collect the following sample of 20 short utterances, tagged with part-of-speech labels.

English			Spanish
hay/NN fever/NN	bread/NN pudding/NN	field/NN ***llamas***/NN	***llamas***/VB médico/NN
barn/NN ***hay***/NN	goat/NN meat/NN	desert/NN ***hay***/NN	***hay***/VB queso/NN
hay/NN burns/VB	fish/NN swim/VB	dogs/NN bark/VB	vemos/VB ***llamas***/NN
llamas/NN spit/VB	people/NN run/VB	birds/NN fly/VB	comemos/VB bebemos/VB
ride/VB donkeys/NN	make/VB ***hay***/NN	eat/VB drink/VB	alimentos/NN bebidas/NN

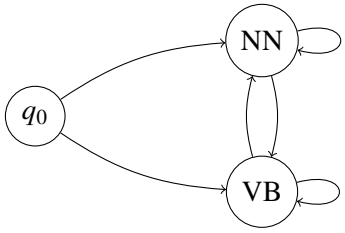
Of course, the above is a very biased sample of general English and Spanish (and some of the utterances are barely grammatical), but let's assume that they are a good sample of the corpus we have at hand. We will tag and identify the language of the unseen utterance *hay llamas*.

a. Based on the training data, what is the prior probability that an utterance selected at random will be English? Spanish?

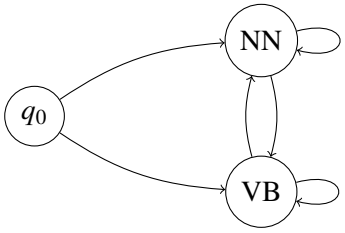
English: _____ Spanish: _____

b. Assume the tags are generated by a separate hidden Markov model for each language. Based on the training data, estimate the *transition probabilities* for each language.

English



Spanish

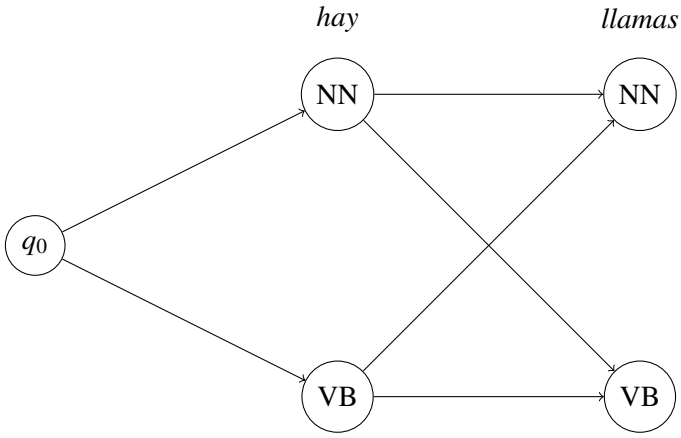


c. Now estimate the *emission probabilities*, again separately for each language. Only write down the probabilities for the words *hay* and *llamas*.

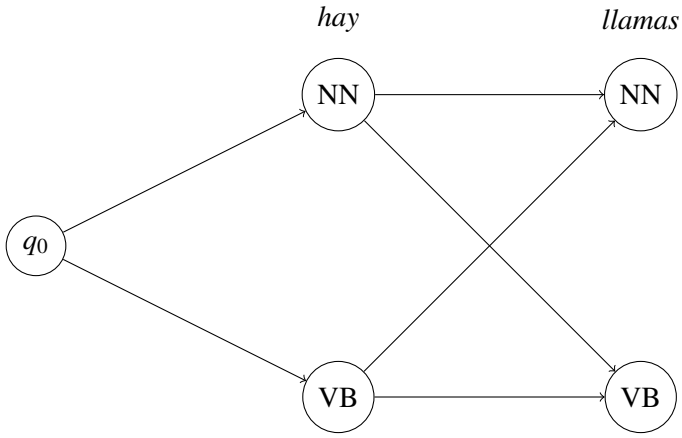
	English		Spanish	
	hay	llamas	hay	llamas
VB				
NN				

d. For each language, use the Viterbi algorithm to decode the most likely tag sequence.

English



Spanish



e. What is the most likely language and tag sequence for the utterance *hay llamas*? Why? (Don't forget the prior probabilities!)

f. What is the most likely language (under any tag sequence) for the utterance *hay llamas*? Why? (Don't forget the prior probabilities!)

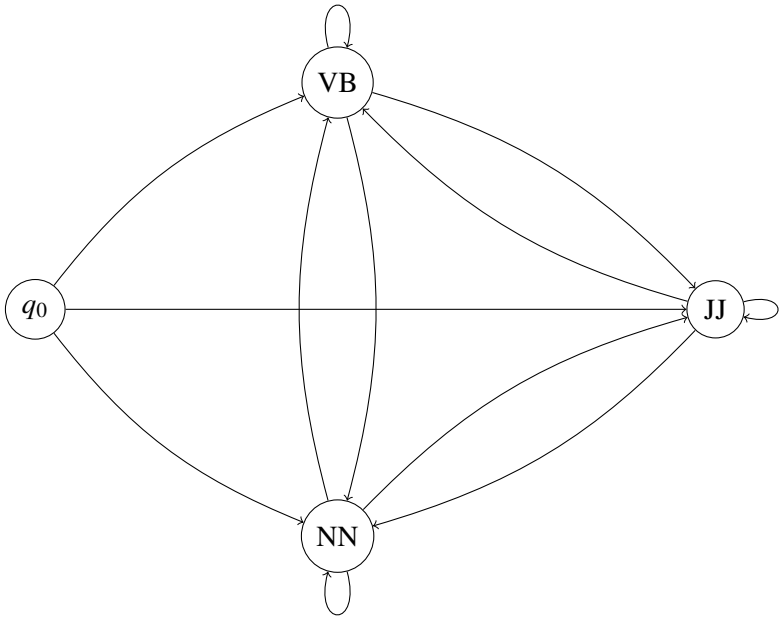
Problem 5. Suppose we want to use a hidden Markov model to tag a text with the most likely tag for each word, regardless of context. (Yes, this is overkill, as there are much simpler ways to do this, but for this exercise assume we want to use the mechanism of a hidden Markov model.) How would we set the transition probabilities? How would we set the emission probabilities? Why?

Problem 6. You are building a system to tag utterances in English. You collect the following sample of 20 short utterances, tagged with part-of-speech labels.

mister/NN jones/NN	**mister**/NN smith/NN	color/NN **gray**/NN	**squirrels**/NN eat/VB
money/NN talks/VB	**gray**/NN wins/VB	**squirrels**/NN drink/VB	**mister**/NN healthy/JJ
mister/NN runs/VB	princess/NN royal/JJ	**squirrels**/VB food/NN	earn/VB **money**/NN
drink/VB tea/NN	cook/VB potatoes/NN	fix/VB things/NN	**gray**/JJ **squirrels**/NN
blue/JJ **gray**/JJ	happy/JJ **squirrels**/NN	yellow/JJ red/JJ	mellow/JJ green/JJ

Using the above data, we will tag the unseen sentence *mister gray squirrels money*.

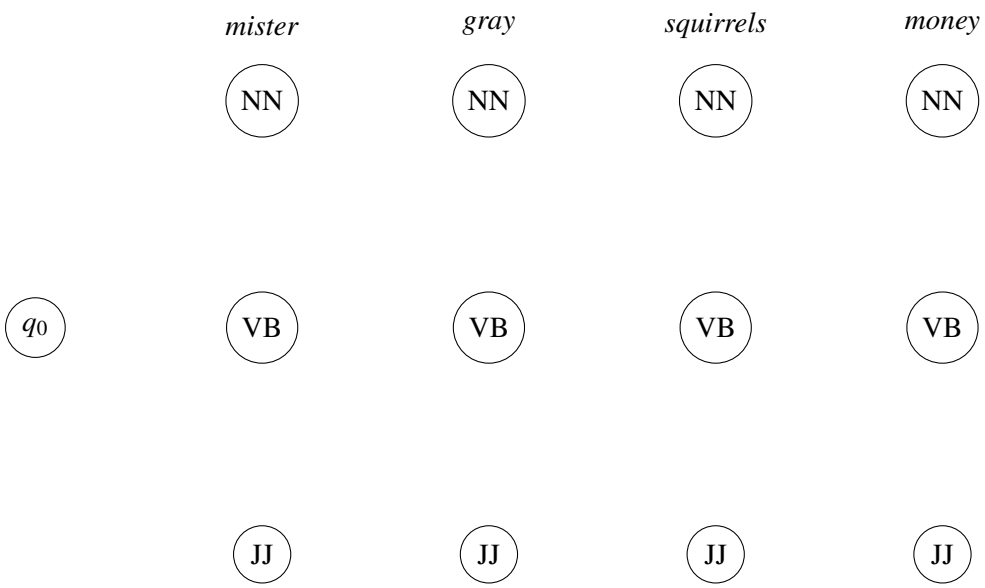
a. Assume the tags are generated by a hidden Markov model. Based on the training data, estimate the *transition probabilities* of the model.



b. Now estimate the *emission probabilities*. Only write down the probabilities for the words *mister*, *gray*, *squirrels*, and *money*.

	mister	gray	squirrels	money
NN				
VB				
JJ				

c. Now use the Viterbi algorithm to decode the utterance *mister gray squirrels money*. Remember the backpointers. Do not draw arrows for zero probabilities. Show your work.



d. What is the most likely tag sequence for the utterance *mister gray squirrels*?

e. What is the most likely tag sequence for *mister gray squirrels money*?

f. Did any of the tags change from part (d) to part (e)? Why or why not?

Problem 7. A program intended to be a hidden Markov model part-of-speech tagger had a bug which caused it to ignore the previous state when estimating transition probabilities. That is, instead of estimating transition probabilities $a(q_i, q_j) \leftarrow P(state = q_j | prev\_state = q_i)$, it estimated the transition matrix as $a(q_i, q_j) \leftarrow P(state = q_j)$ for every state q_i. The emission matrix was not affected by the bug, so it was estimated properly: $b(q_i, v_j) \leftarrow P(observation = v_j | state = q_i)$.

a. Is the buggy model still a Markov chain? Explain why or why not.

b. If the buggy model is used for decoding, what tag will it give to each word? Explain why.

Problem 8. You are building a system to identify named entities in English headlines using the B-I-O tagging scheme: each word receives a tag with the following meaning:

B: The word is the first word (*beginning*) of a named entity.

I: The word is part of a named entity, but not the first word (*inside*).

O: The word is not part of a named entity (*outside*).

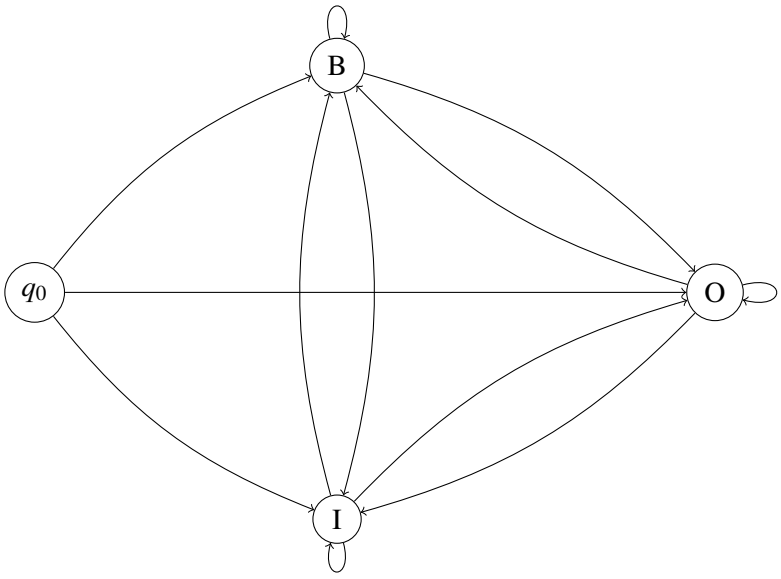
You collect the following sample of 10 headlines, tagged with B-I-O labels.

Man/O **Will**/O **Buy**/O **Apple**/O	Give/O Microsoft/B **Corporation**/I Peace/O
Apple/B Sues/O Google/B Inc/I	Child/O Enters/O Whole/B Foods/I
Dancer/O Likes/O **Will**/B Smith/I	Save/O Whole/B Foods/I **Corporation**/I
Apple/B Google/B Inc/I Gain/O	Billionaires/O **Buy**/O Samsung/B **Corporation**/I
Woman/O **Will**/O Eat/O **Apple**/O	Mister/B Ronald/I Reagan/I Died/O

Using the above data, we will tag the unseen headline *Apple Will Buy Corporation*.

Note that in the above data there are 10 B tags, 10 I tags, and 20 O tags.
Also, there are 10 B→… transitions, 5 I→… transitions, and 15 O→… transitions.
All the probabilities in parts *a* and *b* are multiples of 0.1.

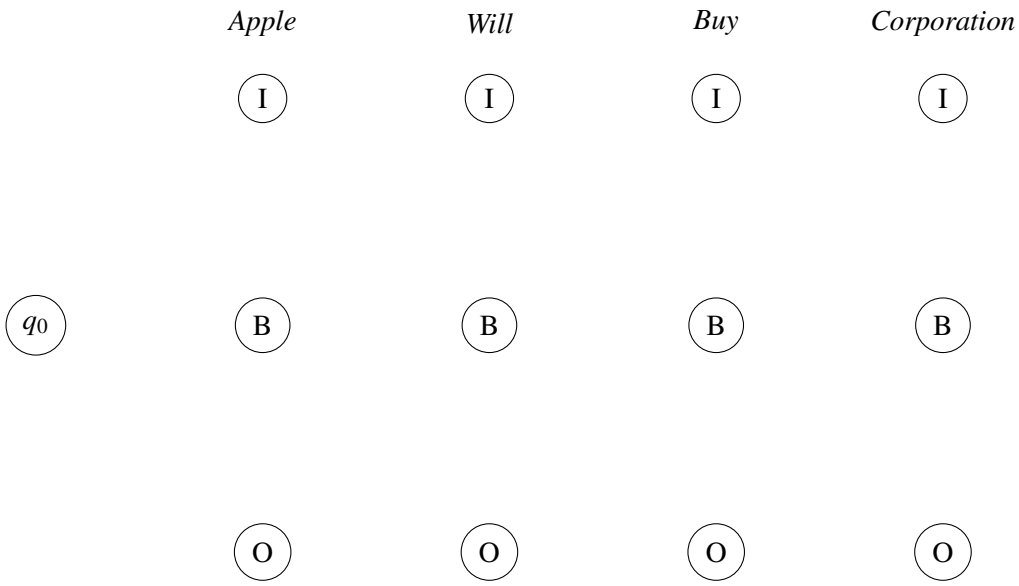
a. Assume the tags are generated by a hidden Markov model. Based on the training data, estimate the *transition probabilities* of the model. Do not perform any smoothing.



b. Estimate the *emission probabilities*. Only write down the probabilities for the words *Apple*, *Will*, *Buy*, and *Corporation*. Do not perform any smoothing.

	Apple	Will	Buy	Corporation
B				
I				
O				

c. Now use the Viterbi algorithm to decode the headline *Apple Will Buy Corporation*. Remember the backpointers. Do not draw arrows for zero probabilities. Show your work.
Note: If your numbers are correct, the algorithm will fail to tag the last word.



d. Briefly explain why the algorithm failed to tag the last word.

e. Without calculating any numbers: Would it be a good idea to fix the tagger by smoothing the *transition* probabilities? Why or why not?

f. Without calculating any numbers: Would it be a good idea to fix the tagger by smoothing the *emission* probabilities? Why or why not?

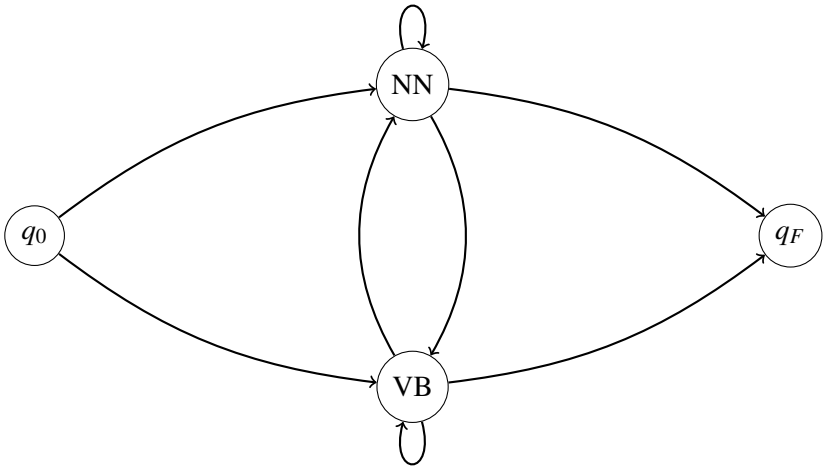
g. The Ratnaparkhi paper on part-of-speech tagging recommends using a tag dictionary, because it greatly improves runtime and does not have a noticeable effect on accuracy. As discussed in class, this is analogous to not smoothing the emission probabilities. Is this reasoning valid for the present B-I-O tagging problem? Why or why not?

Problem 9. You are building a system to tag sentences in English. You collect the following sample of 10 short sentences, tagged with part-of-speech labels.

love/VB **dog**/NN **kisses**/NN	kiss/VB cat/NN fish/NN
cats/NN **fly**/VB planes/NN	cat/NN **kisses**/VB dogs/NN
flies/NN **dog**/VB cats/NN	**dog**/NN eats/VB frog/NN
fruit/NN **fly**/NN lands/VB	cat/NN **kisses**/NN hurt/VB
horse/NN **fly**/NN wins/VB	city/NN birds/NN **fly**/VB

Using the above data, we will tag the unseen sentence *dog kisses fly*. We will assume the tags are generated by a hidden Markov model with an *end state*. That is, in addition to transitions from an initial state q_0 to the first word of a sentence, and transitions between words, we also estimate transition probabilities from the final word of a sentence to a special end state q_F. (You can find more on end states in an earlier draft of the Jurafsky and Martin chapter on Hidden Markov Models.)

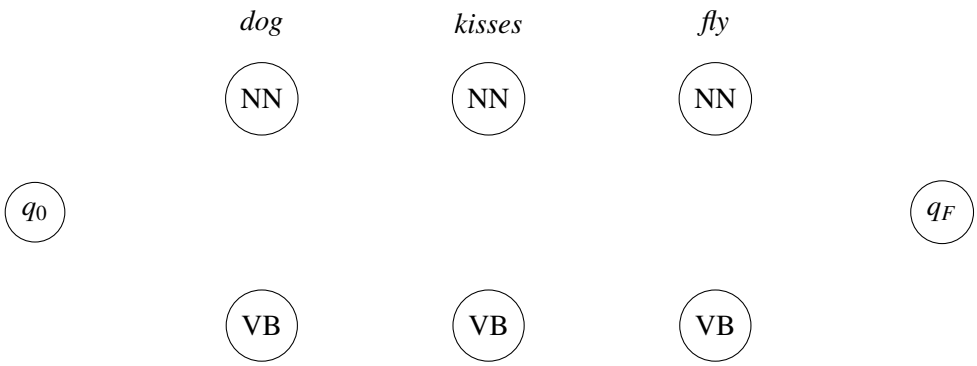
a. Based on the training data, estimate the *transition probabilities* of the model. Note that there are 20 NN tags and NN $\rightarrow \ldots$ transitions, and 10 VB tags and VB $\rightarrow \ldots$ transitions. Remember that transitions out of each state must form a probability distribution.



b. Now estimate the *emission probabilities*. Only write down the probabilities for the words *dog*, *kisses*, and *fly*.

	dog	kisses	fly
NN			
VB			

c. Now use the Viterbi algorithm to decode the utterance *dog kisses fly*. Remember the back-pointers. Do not draw arrows for zero probabilities. Show your work.



d. If the decoder were to stop at the word *fly* before reaching the end state, what would be the most likely tag sequence (or sequences) for the utterance *dog kisses fly*?

e. After reaching the end state, what is (or are) the most likely tag sequence (or sequences) for the utterance *dog kisses fly*?

f. Does having an end state make a difference? What properties of the data does the end state capture, and how do these bear upon the decision made by the Viterbi decoder?