Exercise Part 1

Q1.1 Write regular expressions for the following languages. By “word”, we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

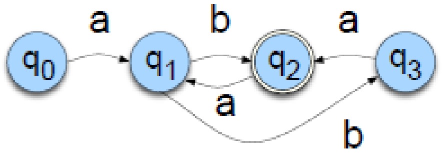
1. The set of all lower case alphabetic strings ending with a letter *b*;

2. The set of all strings with two consecutive repeated words (e.g., “Humbert Humbert” and “the the” but not “the bug” or “the big bug”);

3. All strings that have both the word *grotto* and the word *raven* in them (but not, e.g., words like *grottos* that merely *contain* the word *grotto*);

Q1.2. Design an FSA that accepts a subset of valid web addresses. A commonly seen web address typically starts with “http” or “https”, followed by a “://www.” and name of the organization or company, then “.com/” or “.org/”. The address then has directory nesting like “abc/def/ghi”

Q1.3 Write a regular expression for the language accepted by the following NFSA.



Q1.4 You are given a collection of 4 documents. Draw the inverted index that would be built this document collection. State your assumptions about any preprocessing.

D1: School offers a new course
D2: A new course is offered
D3: The new course is good
D4: School offers many courses

Q1.5 In ordinary English text, there are about 4.5 characters per word on average. After indexing, the average length of a dictionary word is 8 characters. Suppose an index has a dictionary with 100,000 words. Assume that a byte is the smallest storage unit and one character occupies one byte. Estimate the space usage in number of bytes of this dictionary by using each of the following two storage strategies.

A. Dictionary-as-a-string without blocking.
B. Dictionary-as-a-string, using blocked dictionary storage with block size of 8.

Q1.6 Describe an approach to compute sentence-level word co-occurrences in a given document collection using the MapReduce framework. Your approach shall produce the output in the format of (w_a, w_b, n), where w_a and w_b are two words, and n is the number of sentences that both words appear in.

Q1.7 Given the following three word sequences (i.e., the corpus).

very good tennis player in US open

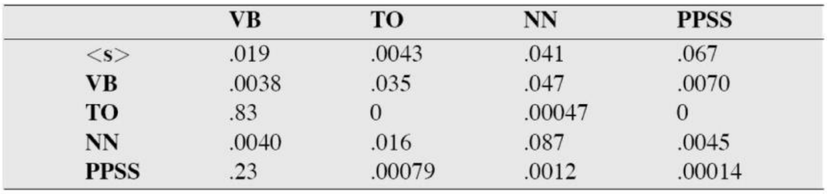
tennis player US Open

tennis player qualify play US Open

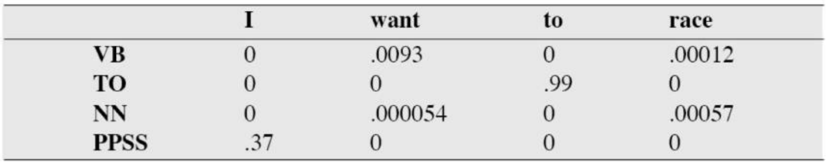
(i) Build a table of bigram counts from the word sequences.

(ii) Compute the bigram probabilities using Laplace smoothing.

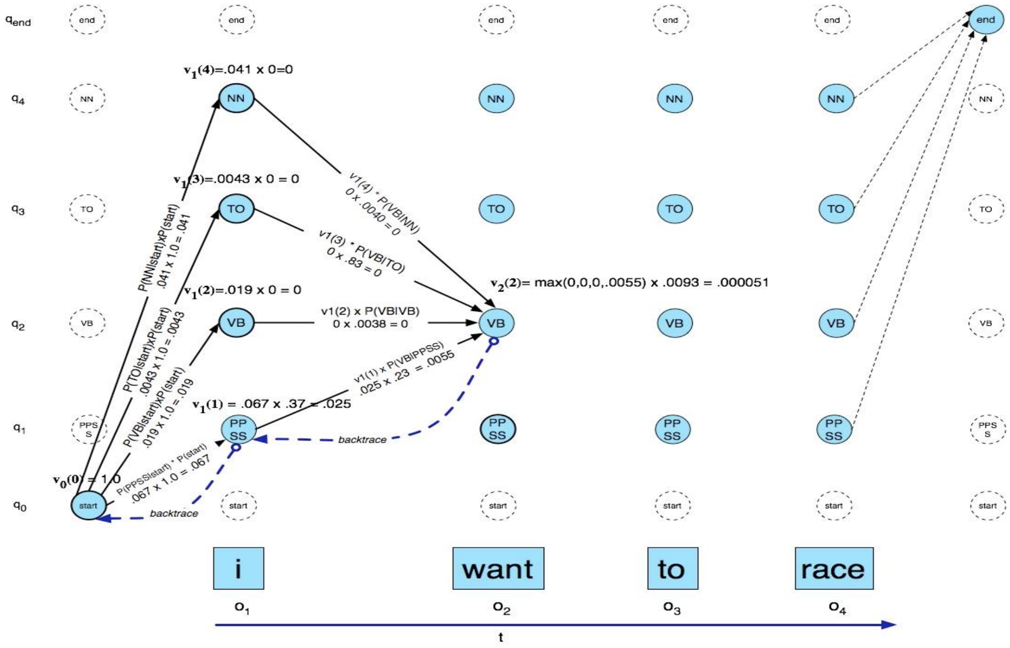
Q1.8 Finish the computation of the Viterbi algorithm in the Viterbi example used in class for HMM. The transition probability and word likelihood probabilities are in the following tables.



	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014



	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0


q_{end}
q_4
q_3
q_2
q_1
q_0
end
NN
TO
VB
PPSS
PP SS
start
$v_1(4)$=.041 x 0=0
$v_1(3)$=.0043 x 0 = 0
$v_1(2)$=.019 x 0 = 0
$v_1(1)$ = .067 x .37 = .025
$v_0(0)$ = 1.0
P(NN|start)xP(start) .041 x 1.0 = .041
P(TO|start)xP(start) .0043 x 1.0 = .0043
P(VB|start)xP(start) .019 x 1.0 = .019
P(PPSS|start) * P(start) .067 x 1.0 = .067
v1(4) * P(VB|NN) 0 x .0040 = 0
v1(3) * P(VB|TO) 0 x .83 = 0
v1(2) x P(VB|VB) 0 x .0038 = 0
v1(1) x P(VB|PPSS) .025 x .23 = .0055
$v_2(2)$= max(0,0,0,.0055) x .0093 = .000051
backtrace
backtrace
i
want
to
race
o_1
o_2
o_3
o_4
t