National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	NLP	Course Code:	CS4063
	Degree Program:	BS-CS	Semester:	Spring 2023
	Exam Duration:	60 Minutes	Total Marks:	80
	Paper Date:	10-04-2023	Weight	10
	Sections:	ALL	No of Page(s):	2
	Exam Type:	Midterm II		

Student : Name:________________________________ Roll No.________________ Section:_______

Instruction/Notes: Attempt all questions. *Do not use pencil or red ink to answer the questions*. Attempt all questions. Programmable calculators are not allowed.
Solve Q3 on the question paper and others on the answer sheet.

Q1: In a population of 100, a medical test identifies 55 sick individuals. Out of these, 33 are actually sick. Of the remaining 45 individuals, 37 are also sick. Please fill in the following blanks: (mention formulas where required) **(14)**

a. True positives ____________________________ 33
b. False positives ____________________________ 22
c. True negatives ____________________________ 8
d. False negatives ___________________________ 37
e. Accuracy __________________________________ = (tn+tp)/(tn+tp+fn+fp) = (33+8)/100 = 0.41
f. Precision _________________________________ = tp/tp+fp = 33/55 = 0.6
g. Recall _____________________________________= tp/(tp+fn) = 33/(33+37) = 33/70 = 0.47

Q2: You are given the output confusion matrix for a 3-class email classification task. Compute **micro-averaged** and **macro-averaged, precision** and **recall**. Show your work for partial credit, mention the formulas too. (5x5)

System Output \ Gold Labels	urgent	normal	spam
urgent	9	12	2
normal	5	45	40
spam	23	33	300

i. Micro-averaged precision = ____0.754797_________
ii. Macro-averaged precision = _____0.578__________
iii. Micro-averaged recall = _______0.754797__________
iv. Macro-averaged recall = ________0.540145___________
v. Which one of the two measures (micro vs macro) reflected the true picture in this case and why?

________MACRO_________________________________

Q3. Suppose you are working on a speech recognition system that is designed to recognize three different phonemes: "aah", "eeh", and "ooh". You have a training dataset of recorded speech samples that have been labeled with the corresponding phoneme for each sample. However, the transitions between phonemes in the speech samples are not well defined, and there is some overlap between the phonemes in some of the samples. To model the transitions between phonemes and the probabilities of observing each phoneme in the dataset, you want to use the Baum-Welch algorithm to estimate the A and B matrices for a hidden Markov model (HMM).

Here is some additional information about the problem:

The training dataset consists of 100 speech samples, with an average length of 50 phonemes per sample.
The observations are discrete, with each phoneme represented by a vector of 10 features.
You have already determined the number of hidden states (N) in the HMM, which is 5.
For this problem, how would you initialize the A and B matrices for the Baum-Welch algorithm? What factors would you consider when choosing the initial values for the matrices? (10)

A: 5x5 (random, all ones, specify the underlying structure of the data and then decide, uniform distribution)
B: 5x10x50 (random, row sum should be 1 for each state)
Therefore, the B matrix needs to have 5 rows (one for each hidden state) and 10 columns (one for each feature of the phoneme representation, and 50 phonemes per sample).

Name: ________________ Reg #: ________________ Section: ___________

National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Natural Language Processing**	**Course Code:**	**CS 535**
Program:	**BS(Computer Science)**	**Semester:**	**Spring 2018**
Duration:	**60 Minutes**	**Total Marks:**	**24**
Paper Date:	**12-April-18**	**Weight**	**13%**
Section:	**ALL**	**Page(s):**	**5**
Exam:	**Midterm 2 Solution**		

Q1) A sentence can easily have more than one parse tree that is consistent with a given CFG. How do PCFGs and non-probability-based CFGs differ in terms of handling parsing ambiguity? [2 Marks]

Answer:
PCFGs define probability of each rule which can be used to find probability of parse tree. Tree with maximum probability is selected.

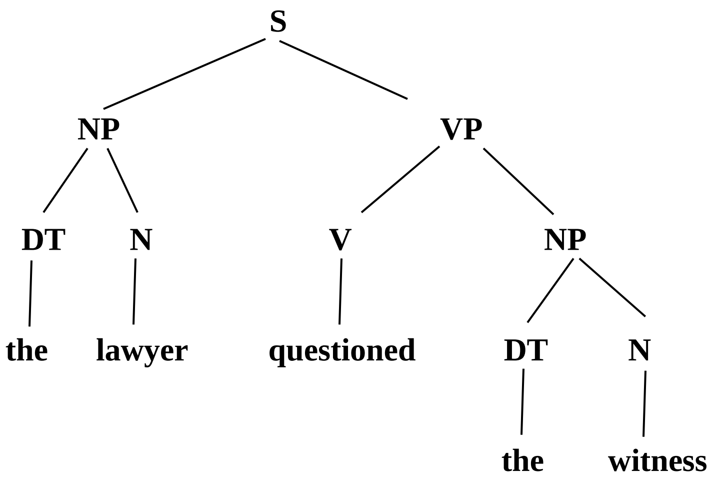
Q2) Which of the following is a false statement about PCFGs: [2 Marks]

a) The rules impose independence assumptions that effect poor modeling of structural dependency across the tree.
b) The rules do not model syntactic facts about particular words, which causes a variety of problems.
c) The joint probability of a sentence, S, and a parses of it, T, is the same as the probability of the parse, T.

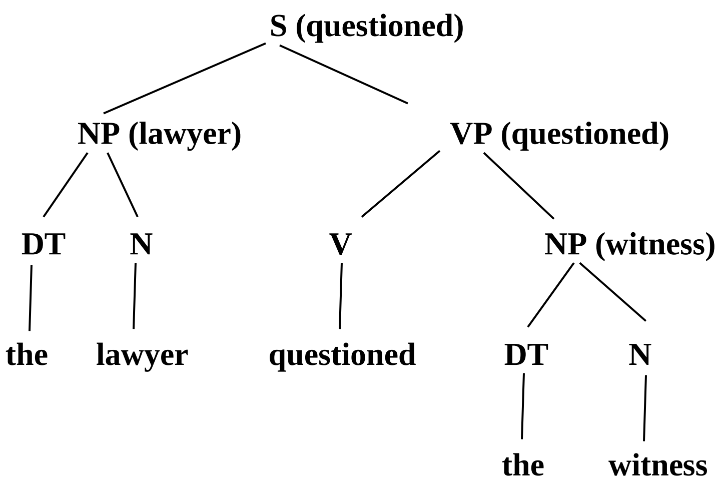
Answer: They are true

Q3) a) Which of the following trees is a lexicalized tree? [1 Mark]

Tree 1



Tree 2



Solution: Tree2

b) For the trees above, when you count and estimate the probability for rules, which tree is most likely to encounter sparseness problem? [2 Marks]

Tree2

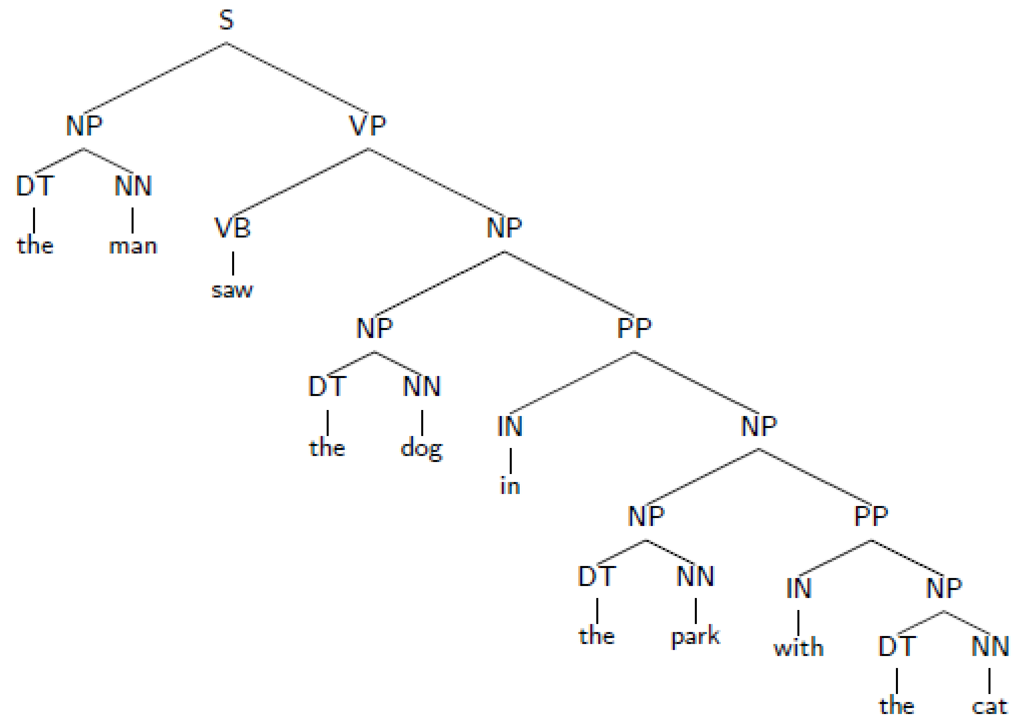
c) How can you alleviate sparseness problem encountered in estimating probability for parse trees? [2 Marks]

smoothing

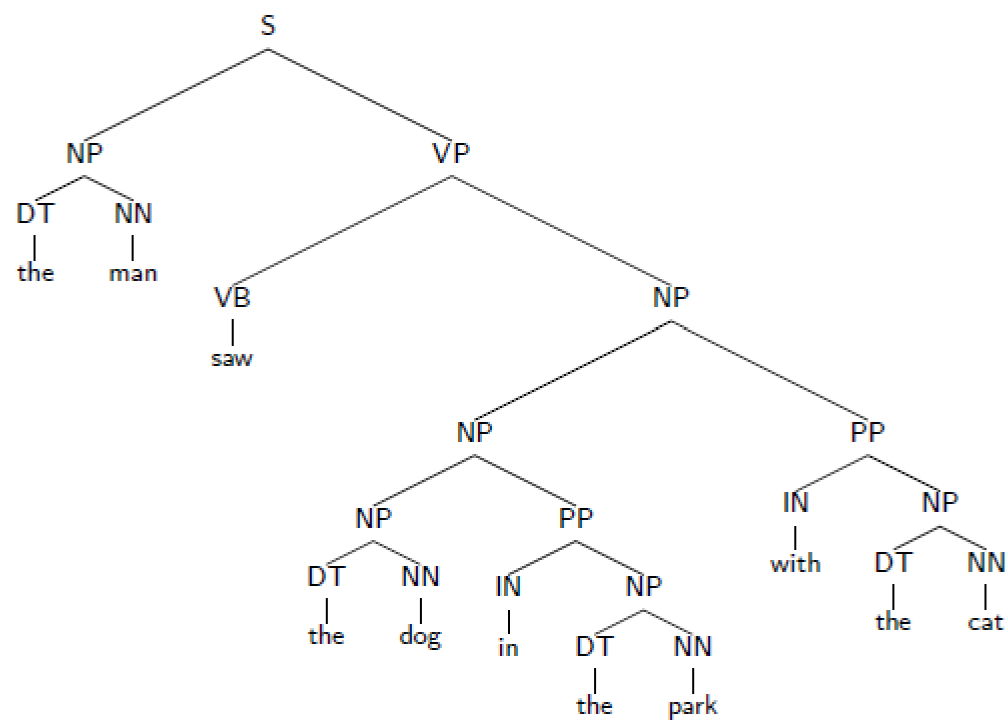
Q4) Consider a context-free grammar with the following rules (assume that S is the start symbol):
S → NP VP
NP → DT NN
NP →NP PP
PP → IN NP
VP → VB NP
DT → the
NN →man
NN → dog
NN → cat
NN → park
VB → saw
IN → in
IN → with
IN → under

How many parse trees for "the man saw the dog in the park with the cat"? Draw all the parse trees for this sentence. [5 Marks]

Two parse trees, parse tree 1:



Two parse trees, parse tree 2:



Name: ________________ Reg #: ________________ Section: ___________

Q3) Consider a trigram HMM, as introduced in class. We saw that the Viterbi algorithm could be used to find $max_{y1, \ldots\ldots y\,n+1}\, p(x1, \ldots.x\,n; y1, \ldots\ldots y\,n+1)$ where the max is taken over all sequences $y1, \ldots\ldots y\,n+1$ such that $y_i \in K$ for i = 1 …. n, and y_{n+1} = STOP. (Recall that K is the set of possible tags in the HMM.) In a trigram tagger we assume that p takes the form

$$p(x1, \ldots.x\,n; y1, \ldots\ldots y\,n+1) = \prod_{i=1}^{n+1} q¿¿$$

Recall that we have assumed in this definition that $y_0 = y_{-1} = y_{-2} = *$, and y_{n+1} = STOP. The Viterbi algorithm is shown in figure below.

Input: A sentence $x1, \ldots.x\,n$, paramters *q (s|u, v)* and *e(x | s)*

Dedinitions: Define *K* to be the set of possible tags. Define $K_{-1} = K_0 = \{*\}$, and $K_k = K$ for *k = 1 …n.*

Initialization: *Set* $\pi(0,*,*) = 1$

Algorithm: For *k = 1 …. n,*

For $u \in K_{k-1}, v \in K_k,$

$$\pi(k,u,v) = max_{w \in K_{k-2}} (\pi(k-1,w,u) \times q(v \mid w,u) \times e(x_k \mid v))$$

Return $max_{u \in K_{n-1}, v \in K_n} (\pi(n,u,v) \times q(STOP \mid u,v))$

Now consider a four-gram tagger, where p takes the form

$$p(x1, \ldots.x\,n; y1, \ldots\ldots y\,n+1) = \prod_{i=1}^{n+1} q¿¿$$

We have assumed in this definition that $y_0 = y_{-1} = y_{-2} = *$, and y_{n+1} = STOP. In the box on next page, give a version of the Viterbi algorithm that takes as input a sentence $x1, \ldots.x\,n$, and finds $max_{y1, \ldots\ldots y\,n+1}\, p(x1, \ldots.x\,n; y1, \ldots\ldots y\,n+1)$ for a four-gram tagger, as defined above equation.
[10 Marks]

Input: A sentence $x_1, \dots x_n$, paramters $q(w|t, u, v)$ and $e(x \mid s)$
Dedinitions: Define K to be the set of possible tags. Define $K_{-2} = K_{-1} = K_0 = \{*\}$, and $K_k = K$ for $k = 1 \dots n$.
Initialization: Set $\pi(0,*,*) = 1$
Algorithm: For $k = 1 \dots n$,

For $t \in K_{k-2}, u \in K_{k-1}, v \in K_k$,

$\pi(k,t,u,v) = \max_{w \in K_{k-3}} (\pi(k-2,w,t,u) \times q(v \mid w,t,u) \times e(x_k \mid v))$

Return $\max_{t \in K_{n-2}, u \in K_{n-1}, v \in K_n} (\pi(n,t,u,v) \times q(STOP \mid t,u,v))$