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:	82
	Paper Date:	28-02-2023	Weight	
	Sections:	ALL	No of Page(s):	
	Exam Type:	Midterm I		

Student : Name:_____ **Roll No.**_____ **Section:**_____

Instruction/Notes: Attempt all questions. Programmable calculators are not allowed.

Q1. You are given the following training corpus. (1+2+2+5+2) 12

<s> i want to eat thai food </s>
<s> we ate pakistani food </s>
<s> i ate apples </s>
<s> they ate thai food </s>

a) Calculate the probability of the following test sentence. Include </s> in your counts just like any other token.
<s> i ate thai food </s>
   i) Unigram Model
   ii) Bigram Model
   iii) Trigram Model
   iv) Bigram model with linear interpolation $\lambda_1 = 0.7$, $\lambda_2 = 0.3$

b) Calculate the perplexity of the test sentence using bigram model.

<s> I ate Chinese food </s>

i. Unigram Model

$= P(I) * P(ate) * P(Chinese) * P(food) * P(</s>)$

$= \frac{2}{21} * \frac{3}{21} * \frac{2}{21} * \frac{3}{21} * \frac{4}{21} = 3.52$ (4)

ii. Bigram Model

$= P(I|<s>) * P(ate|I) * P(Chinese|ate) * P(food|Chinese) * P(</s>|food)$

$= \frac{2}{4} * \frac{1}{2} * \frac{1}{3} * \frac{2}{2} * \frac{3}{3} = \frac{1}{12} = 0.83$ (2)

iii. Trigram Model

$= P(I|<s><s>) * P(ate|<s>I) * P(Chinese|I\ ate) * P(food|ate\ Chinese) * P(</s>|Chinese\ food)$

$= \frac{2}{4} * \frac{1}{2} * \frac{0}{1} * \frac{1}{1} * \frac{2}{2} = 0$ (2)

iv. Bigram language model with linear interpolation ... 0.7 0.3

$\lambda_1 = 0.7, \lambda_2 = 0.7$

$= (\lambda_1 P(Bigram) + \lambda_2 P(Unigram))$

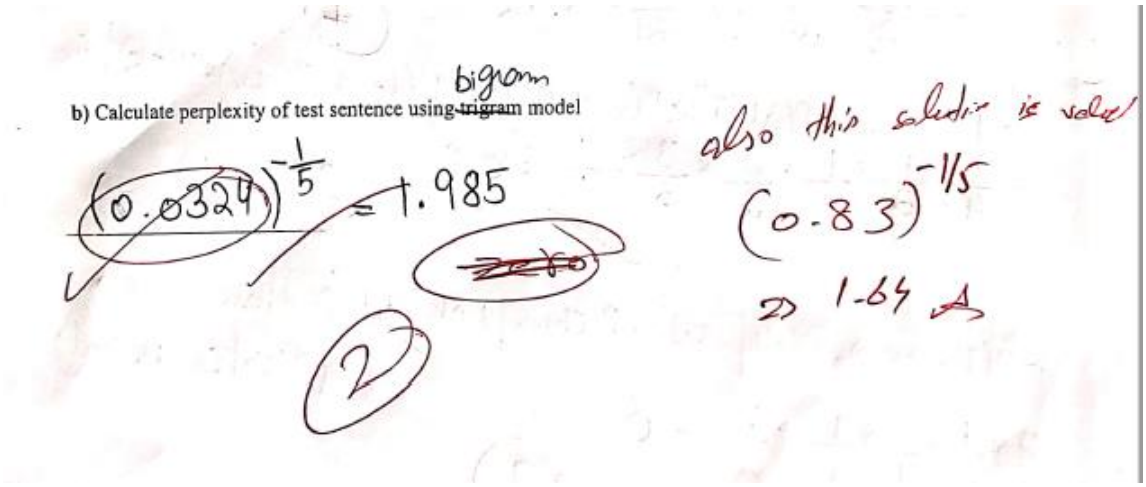
$= \left[0.7\left(\frac{2}{4}\right) + (0.3)\left(\frac{2}{21}\right)\right] * \left[(0.7)\left(\frac{1}{2}\right) + (0.3)\left(\frac{3}{21}\right)\right] * \left[(0.7)\left(\frac{1}{3}\right) + (0.3)\left(\frac{2}{21}\right)\right]$

$\left[(0.7)\left(\frac{2}{2}\right) + (0.3)\left(\frac{3}{21}\right)\right] + \left[(0.7)\left(\frac{3}{3}\right) + (0.3)\left(\frac{4}{21}\right)\right]$

$= (0.35 + 0.028)(0.35 + 0.042)(0.23 + 0.028)(0.7 + 0.4$

$(0.7 + 0.057)$

$= 0.0324$ (3)



Q3: You are given two documents:
Doc 1 = the car is in the parking and the bike is in the garage (13)
Doc 2 = the truck is driven on the highway and the tractor is in the farm parking (15)

a) Compute the normalized term frequency and un-smoothed logarithmic inverse document frequency for the given corpus. You can use log of base 10 for the calculation of IDF. Fill the table based on your calculations: (15+10)
IDF: log(N/n)

Term	Count (doc1)	Count (doc2)	TF (doc1)	TF (doc2)	IDF	TF*IDF (doc1)	TF*IDF (doc2)

Term	Count (doc 1)	Count (doc 2)	Tf (doc 1)	Tf (doc 2)	IDF	TF*IDF (doc 1)	TF*IDF (doc 2)
the	4	4	0.30769231	0.26666667	0	0	0
car	1	0	0.07692308	0	0.301	0.02315385	0
is	2	2	0.15384615	0.13333333	0	0	0
in	2	1	0.15384615	0.06666667	0	0	0
parking	1	1	0.07692308	0.06666667	0	0	0
and	1	1	0.07692308	0.06666667	0	0	0
bike	1	0	0.07692308	0	0.301	0.02315385	0
garage	1	0	0.07692308	0	0.301	0.02315385	0
truck	0	1	0	0.06666667	0.301	0	0.02006667
driven	0	1	0	0.06666667	0.301	0	0.02006667
on	0	1	0	0.06666667	0.301	0	0.02006667
highway	0	1	0	0.06666667	0.301	0	0.02006667
tractor	0	1	0	0.06666667	0.301	0	0.02006667
farm	0	1	0	0.06666667	0.301	0	0.02006667

b) From TF*IDF vectors calculated in part a, compute Euclidean Distance and Cosine Similarity (write the formulae too).

Euclidean Distance:

0.0634

Cosine Similarity:

0

Q4: Answer the following. (10)

I. Which of the following is a type of Minkowski distance? _____
   (a. **Hamming**, b. Levenshtein, c. Jaro)

II. Damerau-Levenshtein allows __4__ edit operations, while Hamming allows __1__ operations. (0, 1, 2, 3, 4, 5)

III. Root of the word "antidisestablishmentarianism" is establish and the lemma is disestablishmentarianism

IV. What are the derivational morphemes in each of these words "entitled", "replays", "cities", "undid", "realism", and "higher"? **(en-, re-, none, un-, -ism, none)**

V. Mention all of the following expressions that the regex /bla+..?!/ will select?
blaa! blat! black! bla?! bla.?! bla+?!

a. bla! **b. blaa! c. blat! d. black! e. bla**?! f. bla+..?! g. bla..?! **h. bla.?! i. bla**+?! j. bla+.?!

Q5. Find the Levenshtein distance between PAYMENTS and APARTMENTS. Use the same algorithm and weights as discussed in the class (i.e. cost(Insertion)=1, cost(Deletion)=1, cost(Substitution)=2). (20)

	#	A	P	A	R	T	M	E	N	T	S
#	0	1	2	3	4	5	6	7	8	9	10
P	1	2	1	2	3	4	5	6	7	8	9
A	2	1	2	1	2	3	4	5	6	7	8
Y	3	2	3	2	3	4	5	6	7	8	9
M	4	3	4	3	4	5	4	5	6	7	8
E	5	4	5	4	5	6	5	4	5	6	7
N	6	5	6	5	6	7	6	5	4	5	6
T	7	6	7	6	7	6	7	6	5	4	5
S	8	7	8	7	8	7	8	7	6	5	4

Minimum Edit Distance: _____ 4

- Show the optimal alignment between the sequences and one possible minimal edit sequence (a sequence of inserts *I*, deletes *D* and substitutions *S*) that would result in an optimal conversion from PAYMENTS to APARTMENTS.

```
     P   A     Y   M   E   N   T   S
     |   |     |   |   |   |   |   |
A    P   A  R  T   M   E   N   T   S

I    S   S  I  S   S   S   S   S   S
```