National University of Computer & Emerging Sciences, Karachi
FAST School of Computing
AI Department
Fall 2023
Mid I Examination
27th September 2023 10:00 AM - 11:00 AM

Course Code: AI4001	**Course Name:** Fundamentals of Natural Language Processing
Instructor Name: Sumaiyah Zahid	
Student Roll No:	**Section No:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **4 questions and 2 pages.**
- In case of any ambiguity, you may make assumptions. But your assumption should not contradict any statement in the question paper.
- Show all steps clearly.

Time: 60 minutes. **Max Marks**: 30 points

Question 1 [5 Points]: **[CLO 1]**

1. Compute the minimum edit distance (using insertion cost 1, deletion cost 1, substitution cost 2) of "fastian" to "fusion". Show your work (using the edit distance grid). **[3 Points]**

Solution:

		f	u	s	i	o	n
	0	1	2	3	4	5	6
f	1	0	1	2	3	4	5
a	2	1	2	3	4	5	6
s	3	2	3	2	3	4	5
t	4	3	4	3	4	5	6
i	5	4	5	4	3	4	5
a	6	5	6	5	4	5	6
n	7	6	7	6	5	6	5

2. Write a Regex expression: **[2 Points]**
   a. To identify phone numbers in a simple format, such as "+92-123-1231234" or "0123-1231234."
      Solution : (\+\d{2}-\d{3}-\d{7}|\d{4}-\d{7})
   b. To validate usernames, allowing only alphanumeric characters and underscores.
      Solution: /\b \w+ \b /

Question 2 [10 Points]: **[CLO 1]**

Consider the following training data:
<s> to be who or not to be who just <\s>
<s> be who you want to be<\s>

1. Compute probability of the test sentence **"be who you want"** using linear interpolation starting from trigram LM for the above training dataset with lambda 1=0.5, lambda 2=0.3 and lambda 3=0.2. **[3 Points]**

Solution:
Tokenize the test sentence into trigrams:

Trigrams: {<s> <s> be, <s> be who, be who you, who you want, you want <\s>}
P(<s> <s> be) = Count(<s> <s> be) / Count(<s> <s>) = 1 / 2 = 0.5
P(<s> be who) = Count(<s> be who) / Count(<s> be) = 1 / 1 = 1
P(be who you) = Count(be who you) / Count(be who) = 1 / 3 = 0.3
P(who you want) = Count(who you want) / Count(who you) = 1 / 1 = 1
P(you want <\s>) = Count(you want <\s>) / Count(you want) = 0/ 1 = 0

Bigrams: {<s> be, be who, who you, you want, want <\s>}
P(<s> be) = Count(<s> be) / Count(<s>) = 1 / 2 = 0.5
P(be who) = Count(be who) / Count(be) = 3 / 4 = 0.75
P(who you) = Count(who you) / Count(who) = 1 / 3 = 0.3
P(you want) = Count(you want) / Count(you) = 1 / 1 = 1
P(want <\s>) = Count(want <\s>) / Count(want) = 0/ 1 = 0

Unigrams: {<s>, be , who , you , want, <\s>}
P(<s>) = 2 / 19
P(be) = 4 / 19
P(who) = 3 / 19
P(you) = 1 / 19
P(want) = 1/ 19
P(</s>)=2/19

P("be who you want") = λ1 *trigram + λ2 * bigram + λ3 *Unigram

P("be who you want") = (0.5 * 0) + (0.3 * 0) + (0.2 * 48/19^6) = 2.04 x10^-7

2. Compute the probability of the test sentence **"to be not who"** using stupid backoff starting from quadgram LM. **[3 Points]**

   Quadgrams: {<s><s><s> to, <s><s> to be,<s> to be not, to be not who, be not who <\s>} so many 0 probability then move to
   Trigram: {<s> <s> to, <s> to be , to be not, be not who, not who <\s>}
   Bigram: { <s> to, to be , be not, not who, who <\s>}
   Unigram: { <s>, to , be, not , who, <\s>}

   P(<s>) = 2 / 19
   P(to)= 3/19
   P(be) = 4 / 19
   P(not) = 1/19
   P(who) = 3 / 19
   P(</s>)=2/19

   P(<s> to be n0t who </s>) =0.4*0.4*0.4 * 2/19*3/19*4/19*1/19*3/19*2/19
3. Compute the perplexity of the test sentence **"to be who"** for bigram and trigram. **[2 Points]**
   For Bigram:
   Perplexity=(1/ P(to be who))^⅓
   P (to be who)= 1 *¾ // skipping <s> </s>
   Perplexity= (4/3)^⅓
   For Trigram:
   P(to be who)= ⅔
   Perplexity= (3/2)^1/3
4. Generate 5 more words for the sentence starting from **"to…"** using bigram LM. **[2 Points]**
   to be who you want

Question 3 [10 Points]: **[CLO 3]**

1. Write the Context Free Grammar rules for the following instructions of Assembly Language. **[5 Points]**

   i. LOAD R0, 100;
   ii. STORE R1, 200;
   iii. ADD R0, R2, R3;
   iv. SUB R2, R0, R1;

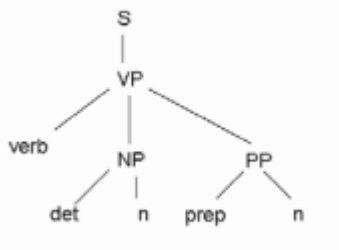
S —> INS <REG>, <CONS>; | INS <REG>, <REG> <REG>;
INS —> LOAD | STORE| ADD |SUB
REG —> R0 | R1 | R2 | R3 | ... | Rn;
CONS —> \d+

2. Draw the top-ranked parse tree for the sentence below by applying the given PCFG. Does the result seem reasonable to you? Why or why not? **[5 Points]**

   Cut the envelope with scissors

Consider the following PCFG:

production rule	probability
S → VP	1.0
VP → Verb NP	0.7
VP → Verb NP PP	0.3
NP → NP PP	0.3
NP → Det Noun	0.7
PP → Prep Noun	1.0
Det → the	0.1
Verb → Cut | Ask | Find| ...	0.1
Prep → with | in | ...	0.1
Noun → envelope | grandma | scissors | men | suits | summer | ...	0.1



The top-ranked sentence structure is shown. (The leaf nodes representing words are omitted.) The probability of the resulting parse tree is 1.0 * 0.3 * 0.7 * 1.0 * (0.1)^5, which is larger than
1.0 * 0.7 * 0.3 * 0.7 * 1.0 * (0.1)^5, the probability of the alternative parse tree (with the [VP → Verb NP] rule expansion) .

Question 4 [5 Points]: **[CLO 1]**

Consider the following short article reviews each labeled with a type, either political or scientific.Use a naive Bayes classifier to classify the test data.

vote,election, policy, policy—> Political
experiments, artificial, intelligence—> Scientific
debate, vote, budget, election, policy—> Political

vote, budget, policy, vote, gun —> Political
intelligence, research, artificial —> Scientific

D: **budget, policy, intelligence, vote.**

1. Compute the most likely class for D using Naive Bayes and add-1 smoothing. **[2.5 points]**

P(Political) = 3/5
P(Scientific) = 2/5

P(budget | Political) = (2 + 1) / (14 + 10) = 3/24 = 1/8
P(policy | Political) = (4 + 1) / (14 + 10) = 5/24
P(intelligence | Political) = (0 + 1) / (14 + 10) = 1/24
P(vote | Political) = (4 + 1) / (14 + 10) = 5/24

P(budget | Scientific) = (0 + 1) / (6 + 10) = 1/16
P(policy | Scientific) = (0 + 1) / (6 + 10) = 1/16
P(intelligence | Scientific) = (2 + 1) / (6 + 10) = 3/16
P(vote | Scientific) = (0 + 1) / (6 + 10) = 1/16

P(D | Political) = P(Political) * P(budget | Political) * P(policy | Political) * P(intelligence | Political) * P(vote | Political) * P(research | Political)
= (3/5) * (1/8) * (5/24) * (1/24) * (5/24)

P(D | Scientific) = P(Scientific) * P(budget | Scientific) * P(policy | Scientific) * P(intelligence | Scientific) * P(vote | Scientific) * P(research | Scientific)
= (2/5) * (1/16) * (1/16) * (3/16) * (1/16)

Document D is Political.

2. What is the class of the test sentence if you use Binary Naive Bayes? . **[2.5 points]**

P(Political) = 3/5
P(Scientific) = 2/5

P(budget | Political) = 2 / 12
P(policy | Political) = 3/12
P(intelligence | Political) = 0
P(vote | Political) = 3/12

P(budget | Scientific) =0
P(policy | Scientific) = 0
P(intelligence | Scientific) = 2/6
P(vote | Scientific) = 0

P(D | Political) = (3/5) * 2/12 *3/12*0*3/12

P(D | Scientific) = (2/5) * 0*0*2/6*0

*****Best of luck while transforming Text into Machine Intelligence.*****