Q1. Generative Modeling (10 Marks) [10]

Recall your recent work in Assignment 2. Given the following sample text (from Eliot's *The Hollow Men*) train a model for poem generation.

```
Between the idea
And the reality
Between the motion
And the act
Falls the Shadow
For Thine is the Kingdom

Between the conception
And the creation
Between the emotion
And the response
Falls the Shadow
Life is long
For Thine is the Kingdom

...
```

We want to generate poems such as the above. Given that we have a large corpus of Eliot's work T, write an algorithm that can learn a bigram model to generate a similar couplet. Be specific on how words and patterns are selected for generating each verse in the couplet.

Algorithm 1 GENERATECOUPLET(T)

- Create a dictionary named bigrams and tokenise corpus
- ~~Start with~~ loop till the length of corpus.
- if corpus[i] not in dictionary add it and extend a dictionary from it and enter key corpus[i+1] with eg corpus[i][corpus[i+1]] = 1 value
- if corpus[i] in dictionary check if corpus[i+1] is in corpus[i] dictionary if present add 1 else enter it and assign value 1 to it.
- Once the bigram dictionary is created extract a start words list consisting of the most used start words.

1: **return** C ▷ generated couplet

continued on page 9

National University of Computer and Emerging Sciences

0051

School of Computing — Islamabad Campus

Q2. Language Modeling (15 Marks) [5+10]

a) How many bigrams can be generated from the following sentence, after *replacing punctuations by a single space*. List them and calculate their probabilities in the bigram language model.

s = *Big fish eat fish eat fish eat fish eat small fish #*

5 Bigrams

$P(\text{Fish} \mid \text{Big}) = 1$

$P(\text{fish} \mid \text{small}) = 1$

$P(\text{eat} \mid \text{fish}) = \frac{4}{5}$

$P(\text{fish} \mid \text{eat}) = \frac{3}{4}$

$P(\text{small} \mid \text{eat}) = \frac{1}{4}$

5

(b) Perplexity measures how good a model is at predicting an unseen test set. Consider two bigram models that use Laplace Smoothing and are trained on the two corpora given below:

c_1 = *oysters eat oysters eat oysters eat small oysters*
c_2 = *oysters eat oysters*

Calculate the perplexity for the two models given that the test sentence is

s = *oysters eat oysters eat oysters*

bigrams in s: for c_1

P(eat|oysters) = ?, seeing how this occurs in c_1

$= \frac{3+1}{4+3} = 4/7$ ✓

P(oysters|eat)

$= \frac{2+1}{3+3} = 3/6 = 1/2$ ✓

P(eat|oysters)

$= \frac{3+1}{4+3} = 4/7$

P(oysters|eat)

$= \frac{2+1}{3+3} = 1/2$

for c_2

$P(eat|oysters) = \frac{1+1}{2+2} = \frac{2}{4} = 1/2$ ✓

$P(oysters|eat) = \frac{1+1}{1+2} = 2/3$

$P(eat|oysters) = 1/2$ ✓

$P(oysters|eat) = 2/3$

$P_{c_1} = \sqrt[5]{\frac{1}{4/7 \times 1/2 \times 4/7 \times 1/2}} = 1.65$ ✓

$P_{c_2} = \sqrt[5]{\frac{1}{1/2 \times 2/3 \times 1/2 \times 2/3}} = 1.55$ ✓ 10

c_2 has lesser perplexity

Q3. Spelling Correction (18 Marks) [10+8]

Given that the following table that shows the confusion matrix for character transposition in spelling errors and that the probability of transposition is defined as $P(t|c) = \frac{trans[c, c_{i+1}]}{count[c, c_{i+1}]}$.

trans[X, Y] = number of times XY was written as YX

X \ Y	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	Total
a	0	0	2	1	1	0	0	0	19	0	1	14	4	25	10	3	0	27	3	6	31	0	0	0	0	0	145
b	0	0	0	0	2	0	0	0	0	0	0	1	1	0	2	0	0	0	2	0	0	0	0	0	0	0	8
c	0	0	0	0	1	0	0	1	85	0	0	15	0	0	13	0	0	0	3	0	7	0	0	0	0	0	125
d	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	10
e	1	0	4	5	0	0	0	0	80	0	0	21	8	16	11	2	0	29	6	0	85	0	0	0	2	0	247
f	0	0	0	0	0	0	0	0	12	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13
g	4	0	0	0	2	0	0	0	0	0	0	1	0	18	0	0	0	3	0	0	3	0	0	0	0	0	28
h	12	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	37
i	18	8	31	3	66	1	3	0	0	0	0	0	0	0	11	0	1	13	23	35	0	0	0	0	0	3	213
j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
l	11	0	0	12	20	0	1	0	4	0	0	0	0	0	1	3	0	0	1	1	3	0	0	0	7	0	73
m	8	0	0	0	20	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	4	0	0	0	0	0	35
n	15	0	6	2	12	0	8	0	1	0	0	0	3	0	0	0	0	0	6	4	6	0	0	0	0	0	57
o	5	0	2	0	4	0	0	0	6	0	0	1	0	5	0	1	0	11	1	1	0	0	7	1	0	0	44
p	17	0	0	0	4	0	0	1	0	0	0	0	0	0	1	0	0	5	3	0	0	0	0	0	0	0	31
q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	12	0	0	0	24	0	3	0	14	0	2	2	0	7	30	1	0	0	0	2	10	0	0	0	2	0	109
s	4	0	0	0	9	0	0	0	5	15	0	0	6	2	0	1	22	0	0	1	3	0	0	0	16	0	83
t	4	0	3	0	4	0	0	21	12	0	0	4	0	0	3	0	0	5	0	0	11	0	2	0	0	0	106
u	22	0	2	1	1	0	2	0	2	0	0	2	1	0	20	2	0	11	11	2	0	0	0	0	0	0	71
v	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
w	0	0	0	0	0	0	0	4	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	8	0	15
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
y	0	1	2	0	0	0	1	0	0	0	0	3	0	0	0	2	0	1	10	0	0	0	0	0	0	0	20
z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	131	9	55	23	187	1	18	32	277	0	3	79	18	70	104	37	1	106	84	67	159	15	9	1	35	3	1500

(a) What would the noisy channel model return for the mispelled word *mdeol* given the following candidate words and their priors? Circle the values that you use in the matrix given above.

w	$P(w)$	$P(x\|w)$	$P(x\|w).P(w)$
mdeol	0.0009	$\frac{11}{1500}$	$\frac{9}{10000} \times \frac{11}{1500} = \frac{99}{1500 0000}$
model	0.157	0	$\frac{157}{1000} \times \frac{0}{1500}$
mdole	0.077	$\frac{20}{1500}$	$\frac{77}{1000} \times \frac{20}{1500} = \frac{154}{150000}$
dmoel	0.0045	0	0

arg max? W=? 8

(b) Consider the following text where the ordering of letters between words is reversed by *transposition*. Use the above matrix to find the probability of each spelling error.

s = fi yuo cna raed tihs, yuo hvae a srtagne mnid too

P(trans [i,f]) -> $\frac{1}{1500}$

P(trans [o,u]) -> 0

P(trans [c,a]) -> $\frac{1}{1500}$

P(trans[h,i]) -> 0

P(trans[a,v]) -> 0

P(trans[t,r]) -> $\frac{5}{1500}$

P(trans[n,g]) -> $\frac{8}{1500}$

Done on Blue Page 5

continued

This page has been intentionally left blank

Q3 P(trans[i,n]) → 5/1500

P(if|fi) → 1/1500 ✓
P(you|you) → 0 ✓
P(read|raed) → 1/1500 ✓
P(have|hvae) → 0 ✓
P(this|tihs) → 0 ✓

can?

7

P(strange|strtagne) = 5/1500 × 3/1500 = 90/2250000

Q1 continued
- Once we have all the Bigrams from bigram.txt
- Sort all the start words
- Select a start word from startwordlist randomly and store in sent ← startword ; word ← that word to be seen as previous start word.
- loop till rand(3,6) words
- sent ← sent + " " + Bigram[word] (randomly select a key from loop)

loop
- C += sentence + "\n"
- If Outer loop integer is a multiple of rand(5,7) C += "\n"

Return C

0051

(7 Marks) [3+4]

Q4. Parts-of-Speech

Consider the following POS annotated story.

```
Little/JJ Red/NNP Riding/NNP Hood/NNP lived/VBD in/IN the/DT woods/NNS with/IN her/PRP mother/NN .
One/CD day/NN Little/NNP Red/NNP Riding/NNP Hood/NNP went/VBD to/TO visit/VB her/PRP granny/NN .
She/PRP had/VBD a/DT nice/JJ cake/NN in/IN her/PRP basket/NN .
On/IN her/PRP way/NN Little/JJ Red/NNP Riding/NNP Hood/NNP met/VBD a/DT wolf/NN .
Hello/UH ! the/DT wolf/NN said/VBD . Where/WRB are/VBP you/PRP going/VBG ?/.
I/PRP m/RB going/VBG to/TO see/VB my/PRP grandmother/NN . She/PRP lives/VBZ in/IN a/DT house/NN
behind/IN those/DT trees./NNS
The/DT wolf/NN ran/VBD to/TO Granny/NNP '/NNP s/NN bed/NN . A/DT little/JJ later/RB , Little/NNP Red/NNP Riding/NNP
got/VBD into/IN Granny/NNP '/NNP s/NN house/NN and/CC ate/NN Granny/NNP up/RB . He/PRP
Hood/NNP reached/VBD the/DT house/NN . She/PRP looked/VBD at/IN the/DT wolf/NN .
Granny/NNP , what/WP big/JJ eyes/NNS you/PRP have/VBP !
All/PDT the/DT better/JJR to/TO see/VB you/PRP with/IN ! said/VBD the/DT wolf/NN .
Granny/NNP , what/WP big/JJ ears/NNS you/PRP have/VBP !
All/PDT the/DT better/JJR to/TO hear/VB you/PRP with/IN ! said/VBD the/DT wolf/NN .
Granny/NNP , what/WP a/DT big/JJ nose/NN you/PRP have/VBP !
All/PDT the/DT better/JJR to/TO smell/VB you/PRP with/IN ! said/VBD the/DT wolf/NN .
Granny/NNP , what/WP big/JJ teeth/NNS you/PRP have/VBP !
All/PDT the/DT better/JJR to/TO eat/VB you/PRP with/IN ! shouted/VBD the/DT wolf/NN .
A/DT woodcutter/NN was/VBD in/IN the/DT wood/NN . He/PRP heard/VBD a/DT loud/JJ scream/NN and/CC
ran/NN to/TO the/DT house/NN . The/DT woodcutter/NN hit/VBD the/DT wolf/NN over/IN the/DT head/NN
. The/DT wolf/NN opened/VBD his/PRP mouth/NN wide/JJ , gave/VBD a/DT cry/NN and/CC Granny/NNP
jumped/VBD out/RP . The/DT wolf/NN ran/VBD away/RB and/CC Little/JJ Red/NNP Riding/NNP Hood/NNP
never/RB saw/VBD the/DT wolf/NN again/RB .
```

(a) The `similarity()` method in NLTK can be used for semi-supervised training. It uses the context of surrounding word annotations to find similar words i.e. words used in the same context. Highlight or underline the words in the above text that are used in a similar context as the word **'eyes'**.

3

(b) How do you plan to use POS tagging in your course project?

haven't decided on the course project yet but would use it most probably to understand the content and infer implicit knowledge

3