0076

National University of Computer and Emerging Sciences

School of Computing | Fall 2023 | Islamabad Campus

CS4063
NLP(CS/SE)

Serial No:

Final Exam
Total Time: 3 Hours
Total Marks: 100

Friday, January 5, 2024
Course Instructor
Dr. Mirza O Beg, Mr. Saad Salman

Signature of Invigilator

Student Name | Roll No | Section | Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.
Instructions:

1. Verify at the start of the exam that you have a total of nine (9) questions printed on ten (10) pages including this title page.
2. Attempt all questions on the question-book and in the given order.
3. This exam is **closed book**. Mobiles, Internet and note-sharing is not allowed. Please see that the area in your threshold is free of any material classified as *useful in the paper*, i.e. mobile/internet or else there may be a charge of cheating.
4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
5. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
6. Use only your own stationery and calculator. If you do not have your own calculator, do manual calculations.
7. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Total
Marks Obtained	10	~~6~~	9	2.5	10	10	12	8	6	~~85~~ 79.5
Total Marks	10	10	10	10	10	10	15	15	10	100

Q1. $tf - idf$ (10 Marks) [4+2+4]

Term frequency - Inverse document frequency (tf—idf), is a numerical statistic that is intended to reflect how important a word is to a document in a corpus. Assuming that $tf(t,d) = log(1 + f_{t,d})$ where $f_{t,d}$ is the raw count of a term t in a document d and $idf(t,D) = log\frac{N}{n_t}$ where N is the total number of documents in the corpus D and n_t is the number of documents containing the term t, for the subsequent questions consider the following documents:

ID	Document Text
d_1	annoyed by oysters devouring oysters
d_2	happy at large fish devouring small fish
d_3	what is the fish devouring once more?
d_4	like crabs devouring oysters and fish

Given the set of terms $\mathcal{T} = \{crabs, devouring\ oysters, oysters, fish\}$ answer the following:

(a) Compute the tf for the terms in $\mathcal{T}$ for each document.

	crabs	devouring oysters	oysters	fish
d1	0	0.3010	0.4771	0
d2	0	0	0	0.4771
d3	0	0	0	0.3010
d4	0.3010	0.3010	0.3010	0.3010

(b) Compute idf for the terms in $\mathcal{T}$ for the corpus.

=df

df(crabs) = 1
df(devouring oysters) = 2
df(oysters) = 2
df(fish) = 3

idf(crabs) = $\log\left[\frac{4}{1}\right]$ = 0.60205
idf(devouring oysters) = $\log\left[\frac{4}{2}\right]$ = 0.301029
idf(oysters) = $\log\left[\frac{4}{2}\right]$ = 0.301029
idf(fish) = $\log\left[\frac{4}{3}\right]$ = 0.12493

(c) Compute *tf-idf(t,D,d)* for the terms in $\mathcal{T}$ for each document in the corpus.

	crabs	devouring oysters	Oyster	fish
d1	0	0.0906	0.1436	0
d2	0	0	0	0.0596
d3	0	0	0	0.0376
d4	0.1812	0.0906	0.0906	0.0376

Q2. Sentiment Analysis (10 Marks) [4+3+3]

The Boolean Naïve Bayes pseudocode given below (**Algorithm 1**) uses α-weighted Laplace smoothing to train a classification model. The algorithm uses words as features for classification.

Algorithm 1 TRAINNAIVEBAYES($\mathcal{C}, \mathcal{T}$)

```
1: procedure NAÏVEBAYESTRAINING(C, T)
2:     V ← EXTRACTVOCABULARY(C)
3:     N ← COUNTTEXTS(T)
4:     for each t ∈ T do
5:         REMOVEDUPLICATEWORDS(t)
6:     for each c ∈ C do
7:         N_c ← COUNTTEXTSINCLASS(T, c)
8:         N_w ← COUNTWORDSINALLTEXTSOFCLASS(T, c)
9:         prior[c] ← N_c / N
10:        doc_c ← CONCATENATETEXTSINCLASS(T, c)
11:        for each w_i ∈ V do
12:            N_i ← COUNTTOKENSOFWORDS(doc_c, w_i)
13:            condprob[w_i][c] ← (N_i + α) / (N_w + α|V|)
14:    return V, prior, condprob                ▷ THE NB MODEL
```

(a) A variant of Naïve Bayes called Multinomial Naïve Bayes sometimes performs worse because it does not normalize text data. Modify the above code to convert it to the Multinomial version of the Naïve Bayes.

(b) Some logical mistakes may have been intentionally added to the above code. Circle any error(s) and state their correction. If there are no errors, circle **None**.

(c) **Algorithm 2** shows the testing for the Boolean Naïve Bayes classifier. The probability *scores* are computed using $P(c).\prod_i P(w_i|c)$ may result in an underflow because of small probability values being multiplied. Suggest a solution and modify the algorithm given below to implement your solution.

Algorithm 2 TESTNAÏVEBAYES($\mathcal{C}, \mathcal{V}, prior, condprob, t$)

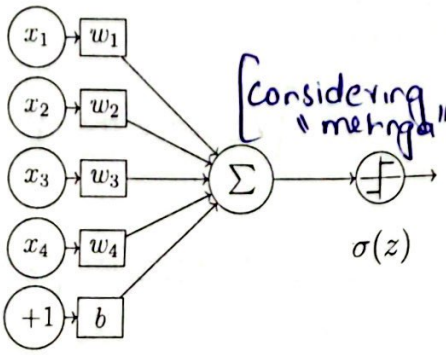
```
1: procedure NAIVEBAYESINFERENCE(C, V, prior, condprob, t)
2:     W ← EXTRACTWORDSFROMTEXT(V, t)
3:     for each c ∈ C do
4:         score[c] ← prior[c]
5:         for each w_i ∈ W do
6:             score[c] *= condprob[w_i][c]
7:     return argmax_{c∈C} score[c]              ▷ THE PREDICTED CLASS
```

Q3. Logistic Regression (10 Marks) [5+5]

Consider the following sentence used for for training the following Logistic Regressor to detect sarcasm.

$y=1$ s_1 = *agar petrol mehnga ho jae, to app ne lena nhi hai. app ne ghabrana nhi hai*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16



inputs weights

Given that s_1 is labeled positive and and ~~s_2 is labeled as negative~~, and features x_1 (*count +ive lexicon*), x_2 (*count -ive lexicon*), x_3 (*negation words*) and x_4 (*number of words*) and the initial weight and bias vector: $W, b = [0, 0, 0, 0, 0]$

Determine the weights and bias vector after training and backpropagating the logistic regressor on the above sentence, given that the learning rate $\alpha = 0.1$.

$z = wx + b$

$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 2 \\ 16 \end{bmatrix} + 0$

$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} [0\ 2\ 2\ 16] + 0$

$= 0 + (2)(0) + 0(2) + 16(0) + 0$

$z = 0$.

$\hat{y} = \sigma(2) = \frac{1}{1+e^{-0}} = 0.5$

$w_{t+1} = w_t - \alpha[\text{loss function}]$

$w_{t+1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0.1 \begin{bmatrix} \sigma(2) - y \\ \sigma(2) - y \\ \sigma(2) - y \\ \sigma(2) - y \\ \sigma(2) - y \end{bmatrix} [x]$

$w_{t+1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} 0.5 - 1 \\ 0.5 - 1 \\ 0.5 - 1 \\ 0.5 - 1 \\ 0.5 - 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 2 \\ 16 \\ 1 \end{bmatrix}$

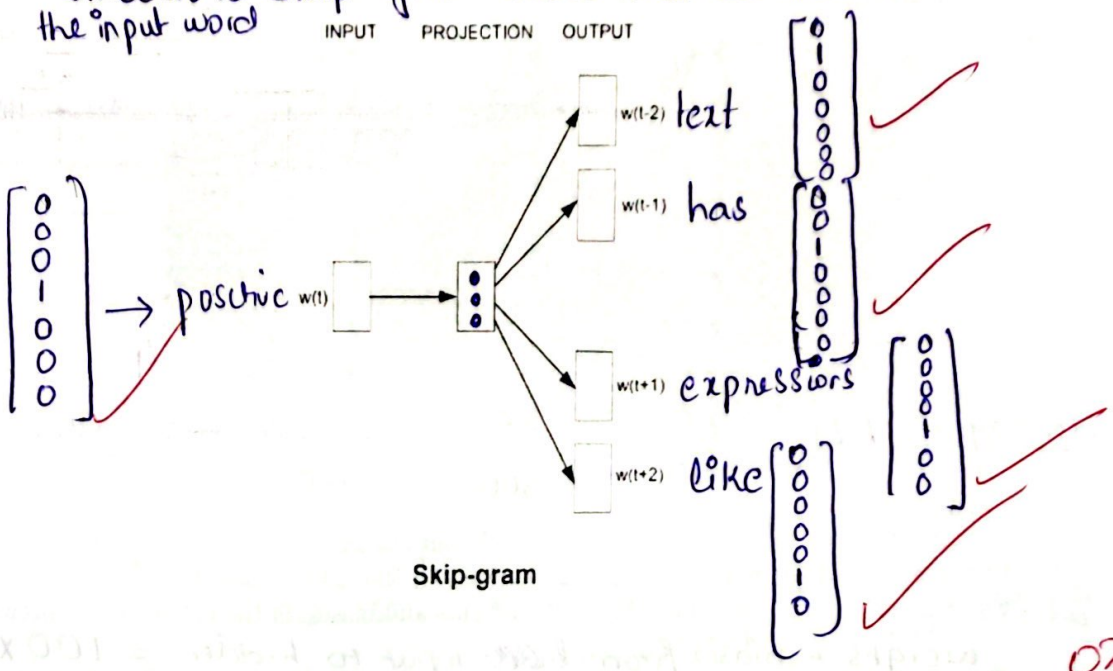
$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 8 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} 0 \\ -1 \\ -1 \\ -8 \\ -0.5 \end{bmatrix}$

$w_{t+1} = \begin{bmatrix} 0 \\ 0.1 \\ 0.1 \\ 0.8 \\ 0.05 \end{bmatrix}$

after back propagation
updated bias = 0.05.
updated weights = $\begin{bmatrix} 0 \\ 0.1 \\ 0.1 \\ 0.8 \end{bmatrix}$

Q4. Vector Semantics (10 Marks) [2+3+2+3]

Consider the test corpus [*"the text has positive expressions like good"*] for training the *skip-gram* model below for generating word embeddings, where the vocabulary size is 7.



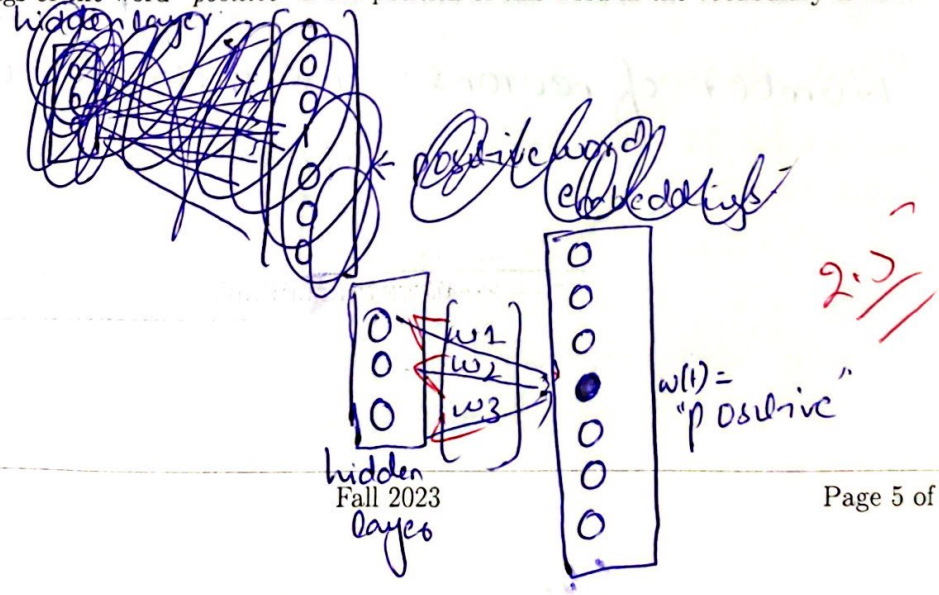
Skip-gram

(a) What is the format of the words in the output layer? Show an example on the diagram above.

(b) Label the diagram above to show the predicted words if the input word is *"positive"*.

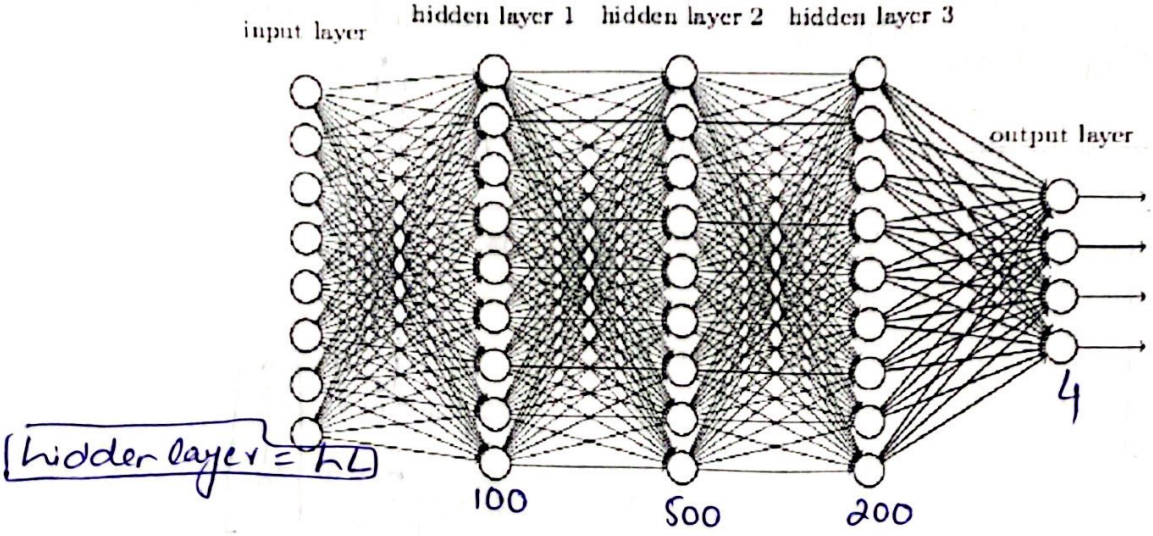
(d) What is the significance of the size of the hidden layer in the above network?

(d) Consider the size of the hidden layer to be 3. Now draw the weights that would represent the embeddings of the word *"positive"* if the position of this word in the vocabulary is 4.



Q5. Feed Forward Networks (10 Marks) [7+3]

Consider the Feed-Forward Deep Network architecture below:



(a) If the network is used to classify datapoits into four classes and there are 100 input features to the network and the hidden layers 1, 2, 3 have 100, 500, 200 neurons respectively, how many weights are there in total, including both weights and biases, in the entire neural network?

weights + biasis from input to hidden layer 1 = 100 x 100 + 100
= 10100

weights + biasis from hidden L1 to hL2 = 100 x 500 + 500
= 50500

" " hL2 — hL3 = 500 x 200 + 200
= 100200

hL3 — output = 200 x 4 + 4 = 804

total = 10100 + 50500 + 100200 + 804 = 161604

(b) How many neurons are there in total, in the neural network above?

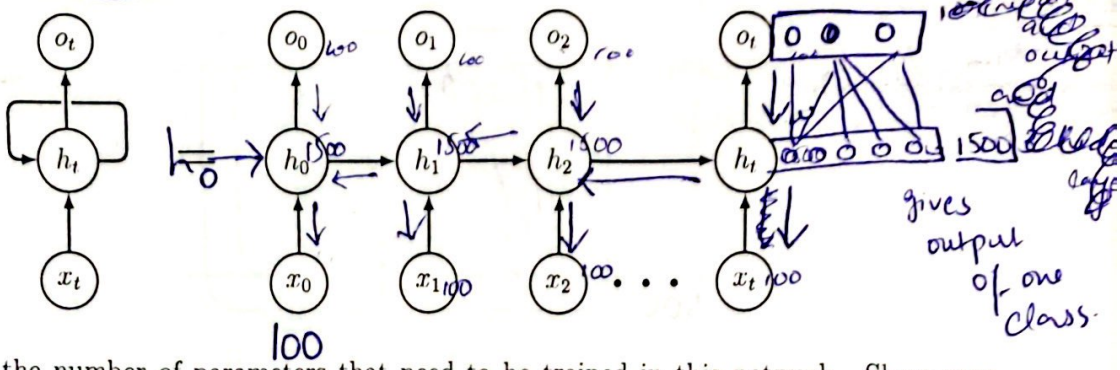
Number of neurons = 100 + 100 + 500 + 200 + 4
= 904

10

Q6. Recurrent Neural Networks (10 Marks) [4+3+3]

Consider the RNN given below and its representation unrolled in time. Assume that the network is being used for classifying sentiments as $C = \{+, -, =\}$, the inputs are vectors of size 100 and the hidden layer contains 1500 neurons.

3 neurons since 3 cla



(a) Determine the number of parameters that need to be trained in this network. Show your work. biasis are 1500.

Lets start calculate number of parameters.
we have input size of 100 and neurons in the hidden layer are 1500 so $[100 \times 1500]$ $= 2^k[1500\,0]$ [multiply with 2 because (100→1500→100)
then this is a recurrent processe so we will added in them is
$1500 \times 1500 = 2250000$
now at final layer output we have 3 classes so $= 100 \times 3 = 300$
So total we have $= 2[15000] + 2250000 + 300 + \text{biasis}$
$2280300 + \text{biasis}$

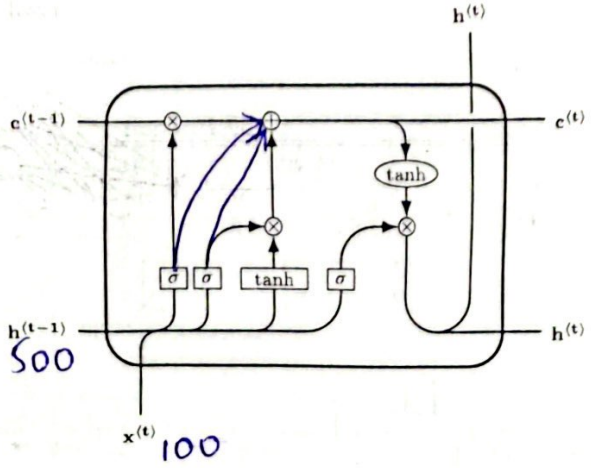
(b) Modify the unrolled RNN diagram above to show what the output layer would look like for the given classification task?

(c) How do vanishing and exploding gradient problems affect the training of RNNs? What are the consequences of these gradient issues during the backpropagation process?

vanishing gradient problem occurs when the value becomes nearer to 0 or 0 a while back propogating as we move far in the sequence so the value for the earlier vectors in sequence we loose them thus context of far away or earlier words we forget and in exploding gradient problem the values number of bits become higher that they are hard to store.

Q7. Long-Short Term Memory (15 Marks) [5+5+5]

Consider the LSTM cell given below.



500

100

(a) Given that the LSTM cell is being used in a network for classifying text into 5 classes, the input $x^{(t)}$ is a vector of size 100 and the hidden state vector $h^{(t)}$ is of size 500, find the total number of parameters to be trained for the classification network containing the LSTM cell above.

$4\left[(100+500)\times 500 + 500\right] + 500\times 5 + 5$

$= 1202000 + 12500$

$= 1214500.$

total number of parameters to be trained.

(b) We want to modify the LSTM cell by directly adding whatever we have learnt from the input and hidden state at timestep t to update the cell state. Make changes to the diagram above to reflect this modification.

whatever learn't from input

(c) What would be the updated equation for the new cell state with the modification made in part (b) above.

before $C_t = \left[f_t \times C_{t-1}\right] + \left[I_t \times \tilde{C}_t\right]$

since we are directly adding the input and hidden state

so after $C_t = \left[f_t\right] + \left[I_t\right]$.

C_t = forget gate + Input gate.

updated $C_t = \sigma\left[W^T_{forget}\left[h_{t-1} | x_t\right] + b_f\right] + \sigma\left[W^T_{input}\left[h_{t-1} | x_t\right] + b_i\right]$

8

Q8. Attention and Transformers (15 Marks) [4+3+3+5]

The following diagram shows the Encoder-Decoder architecture of the Transformer model.

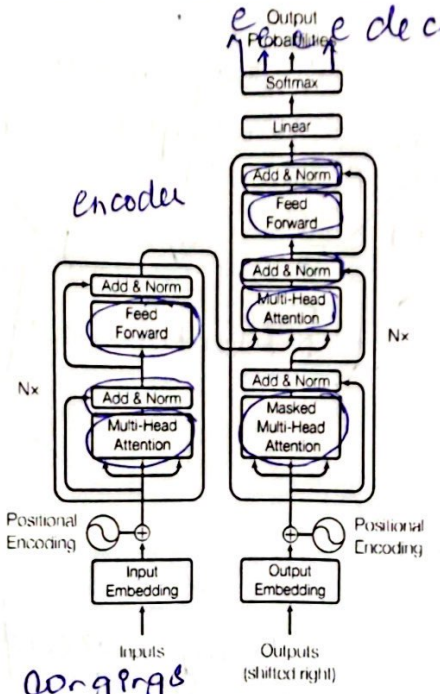


Figure 1: Transformer Model Architecture.

(a) Consider that the model is trained on English-to-Urdu translation. Given that the input is *"Tell these longings to go dwell elsewhere"*, use the diagram above to label and describe the output of the Encoder for this input. Assume that the transformer embeddings are of size 3.

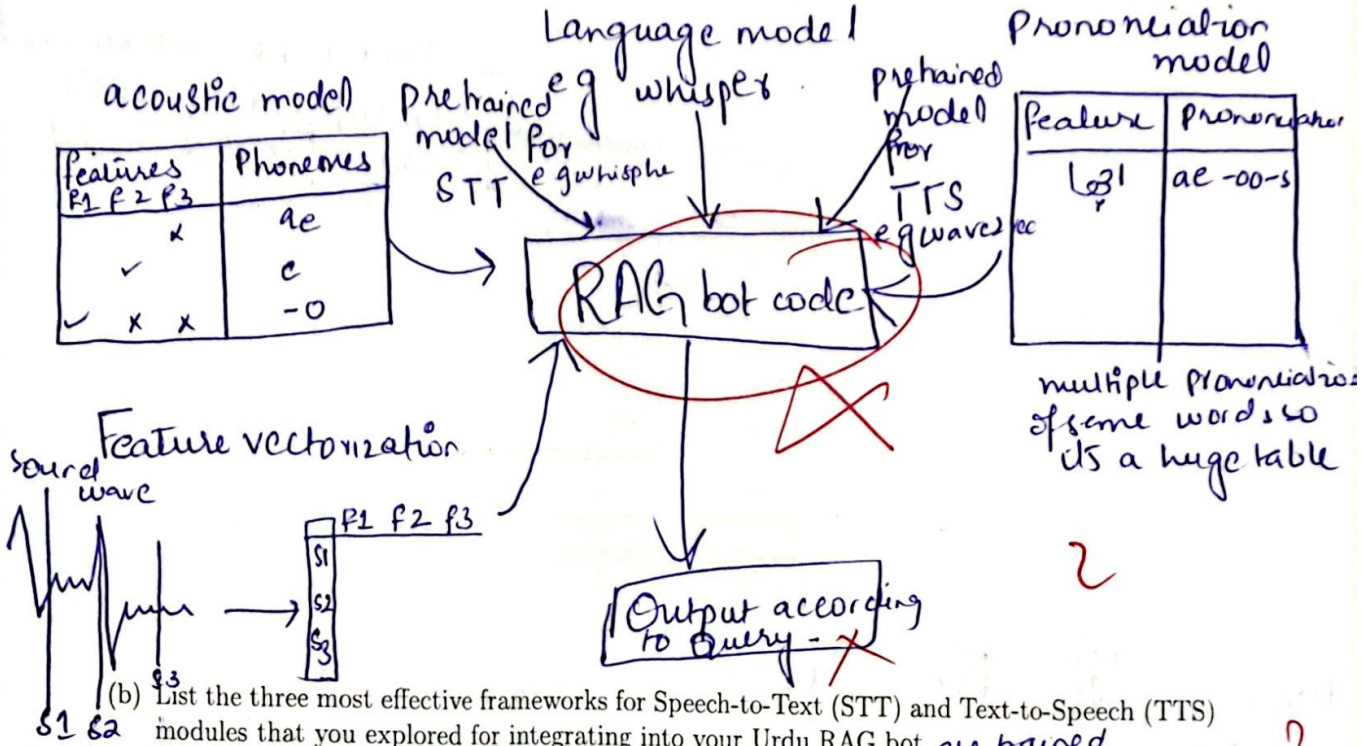
(b) Label the residual connections of the Decoder on the diagram above.

(c) Highlight the units/connections that contain the parameters in the transformer encoder.

(d) Describe using the example sentence *"Tell these longings to go dwell elsewhere"* how *Positional Encoding* is determined? Assume that the input embeddings are of size 3.

Q9. Retrieval Augmented Generation (10 Marks) [5+2+3]

(a) Recall your Urdu Retrieval Augmented Generation (RAG) bot code. Draw the architecture of your pipeline: identify and label the key components.



(b) List the three most effective frameworks for Speech-to-Text (STT) and Text-to-Speech (TTS) modules that you explored for integrating into your Urdu RAG bot.

→ hugging face where we found our pre trained models for speech to text we and text to speech we have wave2vec, tortoiso, MMS, whisper model.

(c) List any observed limitations or issues in the voice integration process and propose potential improvements. Your response should showcase a thoughtful analysis of the impact of voice integration on the overall user experience.

The voice generation for all sorts of texts whether it be a normal reading, some sort of story, or poetry its monotone. All read the same way so to improve the user experience the voice generated should do a semantic analysis and read poetry or generate poetry ~~Our~~ voice in away that it is read with emphasis laid on words that ~~all~~ required highlighted and tune the pitch, sound, tone according to the text.