National University of Computer and Emerging Sciences
Islamabad Campus

Natural Language Processing (CS4063)

Final Examination

Total Time (Hrs): 3
Total Marks: 230
Total Questions: 5

Course Instructor(s):
Prof. Ahmad Raza Shahid, Dr. Mehreen Alam
Section(s): (C & D)

Date: Jan 4, 2025

____________ ____________ ________________________
Roll No **Course Section** **Student Signature**

Do not write below this line.

Attempt all the questions.

Q1: Consider the transformer architecture. Given the following word embedding matrix, with positional encoding embedded in them:

[100]

Embedding 1	Embedding 2	Embedding 3	Embedding 4
0.82	1.5715	1.7993	0.8111
1.45	1.2198	1.0398	1.9098
0.15	0.3401	0.2703	0.4505
1.09	1.5798	1.7398	1.1998
0.92	0.5303	0.8107	0.7611
1.67	1.2898	1.6198	1.5198

a) Calculate the Query, Key, and Value matrices with size 4x4, given: [10 + 10 + 10 = 30]

$$W_q = \begin{bmatrix} 0.75 & 0.23 & 0.89 & 0.64 \\ 0.12 & 0.78 & 0.35 & 0.45 \\ 0.56 & 0.91 & 0.27 & 0.32 \\ 0.84 & 0.49 & 0.03 & 0.71 \\ 0.22 & 0.64 & 0.97 & 0.58 \\ 0.09 & 0.15 & 0.42 & 0.81 \end{bmatrix}, W_k = \begin{bmatrix} 0.34 & 0.76 & 0.12 & 0.91 \\ 0.58 & 0.43 & 0.64 & 0.27 \\ 0.21 & 0.89 & 0.03 & 0.48 \\ 0.92 & 0.67 & 0.17 & 0.34 \\ 0.45 & 0.31 & 0.52 & 0.76 \\ 0.83 & 0.57 & 0.25 & 0.62 \end{bmatrix}, W_v = \begin{bmatrix} 0.11 & 0.87 & 0.56 & 0.14 \\ 0.78 & 0.22 & 0.92 & 0.35 \\ 0.67 & 0.41 & 0.05 & 0.81 \\ 0.32 & 0.65 & 0.84 & 0.29 \\ 0.44 & 0.09 & 0.72 & 0.53 \\ 0.98 & 0.37 & 0.19 & 0.68 \end{bmatrix}$$

b) Calculate the attention scores matrix of size 4x4 using the mathematical expression (for single-head attention): [10 + 10 + 10 = 30]

$$Attention(Q, K, V) = softmax_{row}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Note: Softmax is applied row-wise to cater to the fact that $\frac{QK^T}{\sqrt{d_k}}$ matrix has rows that correspond to scores of tokens.

c) Perform linear transformation to get a matrix of size 4x6 using the following matrix: [10]

$$\begin{bmatrix} 0.8117 & 0.9810 & 0.0307 & 0.0532 & 0.9309 & 0.2345 \\ 0.1340 & 0.2609 & 0.3543 & 0.4715 & 0.1626 & 0.9243 \\ 0.1818 & 0.3091 & 0.5975 & 0.8781 & 0.2104 & 0.3940 \\ 0.4413 & 0.9529 & 0.4678 & 0.9564 & 0.8582 & 0.6896 \end{bmatrix}$$

d) Perform add and normalization. For addition, add the original embeddings matrix to the resultant matrix in part c. Then perform row-wise z-score normalization. [10 + 20 = 30]

Solution:

The query, key, and value matrices can be computed as follows:

Q = Combined Embedding (word+positional) . WQ

K = Combined Embedding (word+positional) . WK

V = Combined Embedding (word+positional) . WV

Let's say, for computing the query matrix, the set of weights matrix (WQ)must have the number of rows the same as the number of columns of the transpose matrix of Combined Embedding, while the columns of the weights matrix can be any.

For example, we have 6*4 matrix of Combined Embedding. Now by transposing it we get 4*6 and WQ should have (6*any column) matrix to perform multiplication. we suppose 4 columns in our weights matrix.

e1 + p1	0.82	1.45	1.9098	1.09	0.92	1.67
e2 + p2	1.5715	0.15	0.3401	1.5798	0.5303	1.2898
e3 + p3	1.7993	1.2198	0.2703	1.7398	0.8107	1.6198
e4 + p4	0.8111	1.0398	0.4505	1.1998	0.7611	1.5198

Transpose of Combined Embedding (4*6)

×

0.75	0.23	0.89	0.64
0.12	0.78	0.35	0.45
0.56	0.91	0.27	0.32
0.84	0.49	0.03	0.71
0.22	0.64	0.97	0.58
0.09	0.15	0.42	0.81

WQ (6*4)
Linear weight for Query

Calculating Q

Similarly, we can compute the **key** and **value** matrices using the same procedure, but the values in the weights matrix must be different for both.

e1 + p1	0.82	1.45	1.9098	1.09	0.92	1.67
e2 + p2	1.5715	0.15	0.3401	1.5798	0.5303	1.2898
e3 + p3	1.7993	1.2198	0.2703	1.7398	0.8107	1.6198
e4 + p4	0.8111	1.0398	0.4505	1.1998	0.7611	1.5198

Transpose of Combined Embedding (4*6)

×

0.34	0.76	0.12	0.91
0.58	0.43	0.64	0.27
0.21	0.89	0.03	0.48
0.92	0.67	0.17	0.34
0.45	0.31	0.52	0.76
0.83	0.57	0.25	0.62

WK (6*4)
Linear weight for Key

e1 + p1	0.82	1.45	1.9098	1.09	0.92	1.67
e2 + p2	1.5715	0.15	0.3401	1.5798	0.5303	1.2898
e3 + p3	1.7993	1.2198	0.2703	1.7398	0.8107	1.6198
e4 + p4	0.8111	1.0398	0.4505	1.1998	0.7611	1.5198

Transpose of Combined Embedding (4*6)

×

0.11	0.87	0.56	0.14
0.78	0.22	0.92	0.35
0.67	0.41	0.05	0.81
0.32	0.65	0.84	0.29
0.44	0.09	0.72	0.53
0.98	0.37	0.19	0.68

WV (6*4)
Linear weight for Value

Calculating K, V

So, after multiplying matrices, the resultant **query**, **key**, and **values** are obtained:

Query (4*4)

3.1268	4.4309	3.3794	4.4486
2.9469	2.0949	2.6465	3.6561
3.4328	3.2256	3.6202	4.8045
2.2974	2.7105	2.6200	3.6555

Value (4*4)

4.8910	3.2246	3.7840	4.1085
2.5206	3.0915	2.9890	2.1643
3.8313	3.7477	4.4962	2.9335
3.4103	2.5298	3.2779	2.6272

4.3238	4.9138	2.1649	4.1596
3.4553	3.5196	1.1616	3.3736
4.6859	4.4728	2.1270	4.3084
3.6812	3.3706	1.7560	3.1637

Key (4*4)

Q, K, V matrices

Now that we have all three matrices, let's start calculating single-head attention step by step.

Single Head Attention Formula

Query (4*4)

3.1268	4.4309	3.3794	4.4486
2.9469	2.0949	2.6465	3.6561
3.4328	3.2256	3.6202	4.8045
2.2974	2.7105	2.6200	3.6555

×

Transpose of (Key) 4*4

4.3238	3.4553	4.6859	3.6812
4.9138	3.5196	4.4728	3.3706
2.1649	1.1616	2.1270	1.7560
4.1596	3.3736	4.3084	3.1637

=

48.2450	51.4896	50.3649	67.8378
48.8342	52.7092	50.9437	68.5383
21.5281	23.6464	22.6911	30.5159
45.0061	47.9705	46.8713	63.1032

Q * Transpose of K (4*4)

Q * transpose of K

For scaling the resultant matrix,

$$\frac{\begin{bmatrix} 48.2450 & 51.4896 & 50.3649 & 67.8378 \\ 48.8342 & 52.7092 & 50.9437 & 68.5383 \\ 21.5281 & 23.6464 & 22.6911 & 30.5159 \\ 45.0061 & 47.9705 & 46.8713 & 63.1032 \end{bmatrix}}{\sqrt{d_k}} = \begin{bmatrix} 24.1225 & 25.7448 & 25.1824 & 33.9189 \\ 24.4171 & 26.3546 & 25.4718 & 34.2691 \\ 10.76405 & 11.8232 & 11.3455 & 15.2579 \\ 22.5031 & 23.9852 & 23.4356 & 31.5516 \end{bmatrix}$$

dk= 4

scaling (Q * transpose of K)

The next step of **masking is optional**, and we won't be calculating it. Masking is like telling the model to focus only on what's happened before a certain point and not peek into the future while figuring out the importance of different words in a sentence. It helps the model understand things in a step-by-step manner, without cheating by looking ahead.

So now we will be applying the **softmax** operation on our scaled resultant matrix.

24.1225	25.7448	25.1824	33.9189
24.4171	26.3546	25.4718	34.2691
10.76405	11.8232	11.3455	15.2579
22.5031	23.9852	23.4356	31.5516

Softmax

$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

$$\text{Softmax of } 24.1225 = \frac{e^{24.1225}}{e^{24.1225} + e^{25.7448} + e^{25.1824} + e^{33.9189}}$$

=

0.0000556	0.0002817	0.0001605	0.9995021
0.0000526	0.0003652	0.0001511	0.9994312
0.0105106	0.0303116	0.0188005	0.9403774
0.0001175	0.0005171	0.0002985	0.9990670

Calculating SoftMax

Doing the final multiplication step to obtain the resultant matrix from single-head attention.

Matrix after softmax (4*4)

0.0000556	0.0002817	0.0001605	0.9995021
0.0000526	0.0003652	0.0001511	0.9994312
0.0105106	0.0303116	0.0188005	0.9403774
0.0001175	0.0005171	0.0002985	0.9990670

×

Value (4*4)

4.8910	3.2246	3.7840	4.1085
2.5206	3.0915	2.9890	2.1643
3.8313	3.7477	4.4962	2.9335
3.4103	2.5298	3.2779	2.6272

=

3.4102	2.5302	3.2780	2.6272
3.4101	2.5302	3.2780	2.6272
3.4068	2.5770	3.2974	2.6345
3.4101	2.5305	3.2782	2.6272

calculating the final matrix of single head attention

Output of SHA (4*4)

3.4102	2.5302	3.2780	2.6272
3.4101	2.5302	3.2780	2.6272
3.4068	2.5770	3.2974	2.6345
3.4101	2.5305	3.2782	2.6272

×

Linear Weights(Random Matrix) 4*6

0.8117	0.9810	0.0307	0.0532	0.9309	0.2345
0.1340	0.2609	0.3543	0.4715	0.1626	0.9243
0.1818	0.3091	0.5975	0.8781	0.2104	0.3940
0.4413	0.9529	0.4678	0.9564	0.8582	0.6896

=

4.8626	7.5221	4.1888	6.7656	6.5304	6.2413
4.8625	7.5220	4.1888	6.7655	6.5303	6.2413
4.8728	7.5439	4.2203	6.8114	6.5452	6.2965
4.8626	7.5222	4.1890	6.7658	6.5304	6.2417

Output

normalizing single head attention matrix

Step 2: Add and Normalization

Once we obtain the resultant matrix from multi-head attention, we have to add it to our original matrix.

Word + Positional Embeddings

e1 + p1	0.82	1.45	1.9098	1.09	0.92	1.67
e2 + p2	1.5715	0.15	0.3401	1.5798	0.5303	1.2898
e3 + p3	1.7993	1.2198	0.2703	1.7398	0.8107	1.6198
e4 + p4	0.8111	1.0398	0.4505	1.1998	0.7611	1.5198

+

Output of Multi-Head Attention

4.8626	7.5221	4.1888	6.7656	6.5304	6.2413
4.8625	7.5220	4.1888	6.7655	6.5303	6.2413
4.8728	7.5439	4.2203	6.8114	6.5452	6.2965
4.8626	7.5222	4.1890	6.7658	6.5304	6.2417

=

5.6826	8.9721	6.0986	7.8556	7.4504	7.9113
6.4340	7.6720	4.5289	8.3453	7.0606	7.5311
6.6721	8.7637	4.4906	8.5512	7.3559	7.9163
5.6737	8.5620	4.6395	7.9656	7.2915	7.7615

Adding matrices to perform add and norm step

To normalize the above matrix, we need to compute the mean and standard deviation row-wise for each row first.

Mean

$$\mu = \frac{\Sigma X}{N}$$

Standard Deviation

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

							Mean	Standard Deviation
5.6826	8.9721	6.0986	7.8556	7.4504	7.9113	→	7.3284	1.1222
6.4340	7.6720	4.5289	8.3453	7.0606	7.5311	→	6.9287	1.2209
6.6721	8.7637	4.4906	8.5512	7.3559	7.9163	→	7.2916	1.4363
5.6737	8.5620	4.6395	7.9656	7.2915	7.7615	→	6.9823	1.3764

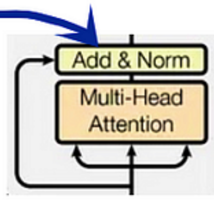
Calculating mean and standard deviation

5.6826	8.9721	6.0986	7.8556	7.4504	7.9113
6.4340	7.6720	4.5289	8.3453	7.0606	7.5311
6.6721	8.7637	4.4906	8.5512	7.3559	7.9163
5.6737	8.5620	4.6395	7.9656	7.2915	7.7615

Mean	Standard Deviation
7.3284	1.1222
6.9287	1.2209
7.2916	1.4363
6.9823	1.3764

$$\frac{\text{Value - Mean}}{\text{Std + error}} = \frac{5.6826 - 7.3284}{1.1222 + 0.0001}$$

-1.4665	1.4646	-1.0959	0.4697	0.1087	0.5194
-0.4051	0.6088	-1.9654	1.1603	0.1081	0.4934
-0.4313	1.0248	-1.9501	0.8769	0.0447	0.4349
-0.9507	1.1477	-1.7020	0.7144	0.2246	0.5661



Q2: Draw the computation graph for the function L=ab(b+1) for a=2 and b=5. Also, find the derivatives of L w.r.t. a & b. Show all the proper derivatives at the nodes and edges with their values and use the chain rules as per your graph to get credit. [10 + 10 = 20]

Q3: Given the movie review as below, do the following: [10 + 10 + 10 = 30]

"snake eyes" is the most aggravating kind of movie: the kind that shows so much potential then becomes unbelievably disappointing. it's not just because this is a brian depalma film, and since he's a great director and one who's films are always greeted with at least some fanfare. and it's not even because this was a film starring nicolas cage and since he gives a brauvara performance, this film is hardly worth his talents.

a. Extract the following features:
   i. Count of positive words
   ii. Count of negative words
   iii. Count of negations
   iv. Log of number of tokens

b. Make a weight vector with weights assigned to each of the features above according to your own assessment of whether a particular feature helps in identifying the polarity (+ve. or -ve.) of the sentiment and its degree (how much positive or negative). Hint: a higher positive weight for a feature would indicate that it is guiding the classifier to classify it positively, and negative features should be sufficiently negative to counter its impact.

c. Use LR to classify the sentiment as either positive or negative.

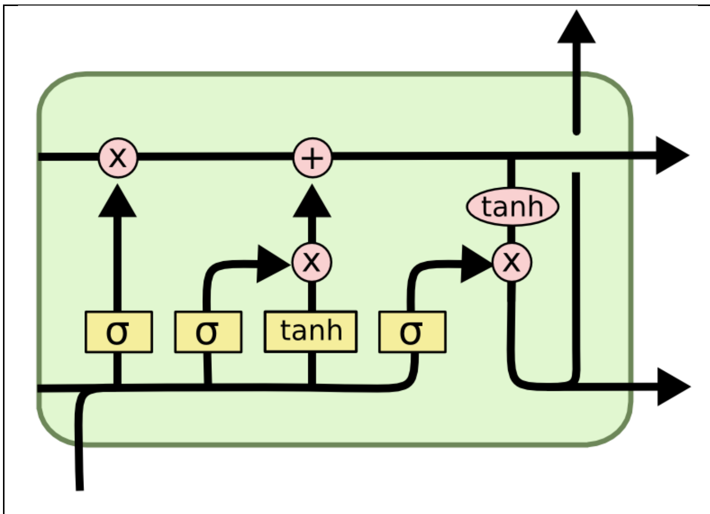
d. Calculate the cross-entropy loss using the expression:

$$L_{CE}(\hat{y}, y) = -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))]$$

Q3: Beam Search is a good compromise between Greedy Search and Exhaustive Search. For a total of four tokens to be predicted in a sequential manner (with one at each time stamp), how many predictions need to be done for a total vocabulary of 10 words for:

A. Greedy search
B. Exhaustive search
C. Beam search with width 5. You may make a drawing for better understanding. [10+10+20 = 40]

Q4.Consider the lstm and its corresponding equations.



$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$h_t = o_t \odot \sigma_h(c_t)$$

For For the value $\mathbf{x_t=[2,4,5]}$, $\mathbf{h_{t-1}=[4,2]}$, **and** $\mathbf{c_{t-1}=[-4,2]}$**.** Make suitable assumptions when needed. You must show vector and matrix dimensions.

A. Find $\mathbf{h_t}$ and $\mathbf{c_t}$

B. Find $\mathbf{y_t}$ [30+10 = 40]