Please, for the Final Exam, use a thicker pen /no pencil and make sure your writup is readable

Student Name: Chaoran Li Student NetID: cxl190012

1. a 10
b. 4
c. 10
d. 6
e. 12
42

76

University of Texas at Dallas
Department of Computer Science
CS6320 – Natural Language Processing
Spring 2021

Instructor: Dr. Sanda Harabagiu

Take-Home Mid-Term Exam

Issued: March 25th 2021

Due: March 27th 2021 – before MidNight

Submit in eLEarning as PDF file

DO NOT DELETE ANYNITHING, Simply add your answers!!!!!

If you submit only the solution with no problems, you will receive 0 points!!!!

2. a 5
b 3
c 6
14

3. 20

Instructions: Do not communicate with anyone in any shape or form. This is an independent exam. Do not delete any problem formulation, just attach your answer in the space provided. If the problem is deleted and you send only the answer, you shall receive ZERO points. If you do not write your name and netid, it will be considered that you did not submit your Midterm exam, and will obtain ZERO points.

Copy and paste the Midterm Exam into a Word document, enter your answers (either by typing in Word, or by inserting a VERY CLEAR picture of your hand-written solution) and transform the file of the exam into a PDF format. If we cannot clearly read the picture, you will get ZERO for that answer! If you create an enormous file for your final exam (i.e. larger than 5 Mbytes) you will receive ZERO for your entire exam. Please follow the instructions from the attached **instructions_submission_EXAM.pdf** file to make sure your final pdf file is of reasonable size.
If you will use a pencil instead of a black pen, you will receive a ZERO for the entire final exam.

Make sure that you insert EACH answer immediately after EACH question. Failure to do so will result in ZERO points for the entire exam! Submit the PDF file with the name **Final_Exam_netID.pdf**, where netID is your unique netid provided by UTD. If you submit your exam in any other format your will receive ZERO points.

The MidTerm exam shall be submitted in eLearning before the deadline. No late submissions shall be graded! Any cheating attempt will determine the ENTIRE grade of the final exam to become ZERO.
Write your answers immediately after the problem statements. If you enter multiple possible answers to the same problem – the most incorrect answer shall be selected and graded (as you are not sure about which answer is the correct one!)

<u>Problem 1</u> Language Models [TOTAL: **50 points**]

You are given the Bigram probability matrix **M** containing evaluations on the Training corpus of the maximum likelihood estimations of the probabilities of the bigrams for the words "Tom", "talks", "a", "lot" and "sometimes". The corpus has 2500 sentences (with not punctuation signs!) and a vocabulary **|V|=25**. You also know that the unigram counts for the words are: Tom = 1500, a = 1000, talks = 2000; sometimes=550 and lot = 400. The matrix **M** is:

	<s>	**Tom**	**talks**	**a**	**lot**	**sometimes**	**</s>**
<s>	0.0	0.65	0.0	0.15	0.0	0.2	0.0
Tom	0.0	0.0	0.7	0.0	0.0	0.3	0.0
talks	0.0	0.0	0.0	0.65	0.0	0.15	0.2
a	0.0	0.0	0.0	0.0	0.9	0.1	0.0
lot	0.0	0.2	0.1	0.0	0.0	0.6	0.1
sometimes	0.0	0.2	0.5	0.0	0.0	0.0	0.3
</s>	1	0.0	0.0	0.0	0.0	0.0	0.0

a) [**10 points**] Reconstruct the bigram counts matrix from the bigram probabilities. Show how you obtained your calculations!

Problem 1 a): Solution:

MLE n-grams (without smoothing):

$P(w_n | w_{n-N+1}^{n-1}) = C(w_{n-N+1}^{n-1} w_n) / C(w_{n-N+1}^{n-1})$

$\Rightarrow C(w_{n-1} w_n) = P(w_{n-1} w_n) C(w_{n-1})$

Besides, 2500 sentences => 2500 <s> and 2500 </s>

	<s>	Tom	talks	a	lot	sometimes	</s>
<s>	0.0×2500	0.65×2500	0.0×2500	0.15×2500	0.0×2500	0.2×2500	0.0×2500
Tom	0.0×1500	0.0×1500	0.7×1500	0.0×1500	0.0×1500	0.3×1500	0.0×1500
talks	0.0×2000	0.0×2000	0.0×2000	0.65×2000	0.0×2000	0.15×2000	0.2×2000
a	0.0×1000	0.0×1000	0.0×1000	0.0×1000	0.9×1000	0.1×1000	0.0×1000
lot	0.0×400	0.2×400	0.1×400	0.0×400	0.0×400	0.6×400	0.1×400
sometimes	0.0×550	0.2×550	0.5×550	0.0×550	0.0×550	0.0×550	0.3×550
</s>	1×2500	0.0×1500	0.0×1500	0.0×2500	0.0×2500	0.0×2500	0.0×2500

Hence, we can get the below results:

	<s>	Tom	talks	a	lot	sometimes	</s>
<s>	0	1625	0	375	0	500	0
Tom	0	0	1050	0	0	450	0
talks	0	0	0	1300	0	300	400
a	0	0	0	0	900	100	0
lot	0	80	40	0	0	240	40
sometimes	0	110	275	0	0	0	165
</s>	2500	0	0	0	0	0	0

10

b) [**5 points**] Using the bigram probability matrix, compute the probability of the sentence "*Tom talks a lot sometimes*" and compare it to the probability of the sentence "*Sometimes Tom talks a lot*".

4/5

Problem 1 b) : Solution:

S1: Tom talks a lot sometimes.

$P_1 = P(Tom \mid <s>)\, P(talks \mid Tom)\, P(a \mid talks)\, P(lot \mid a)\, P(sometimes \mid lot)\, P(</s> \mid sometimes)$

$= 0.65 \times 0.7 \times 0.65 \times 0.9 \times 0.6 \times 0.3 = 0.0479115$

S2: Sometimes Tom talks a lot.

$P_2 = P(sometimes \mid <s>)\, P(Tom \mid sometimes)\, P(talks \mid Tom)\, P(a \mid talks)\, P(lot \mid a)\, P(</s> \mid lot)$

$= 0.2 \times 0.2 \times 0.7 \times 0.65 \times 0.9 \times 0.1 = 0.001638$

"Tom talks a lot sometimes" has a higher possiblility. → incorrect

$0.0479115 > 0.001638$

c) [**10 points**] You also know that in the training corpus there is only one other word in the vocabulary that forms bigrams with w_1=*Tom*, w_2= *talks*, w_3=*a*, w_4=*lot*, namely w_5=*home*.

10

The unigram count for w_5 is 15000 in the Training corpus. You also know the following bigram probabilities from the Training corpus:

	<s>	Tom	talks	a	lot	home	</s>
<s>	0.0	0.7	0.0	0.1	0.1	0.1	0.0
Tom	0.0	0.0	0.75	0.1	0.1	0.05	0.0
talks	0.0	0.0	0.0	0.65	0.2	0.1	0.05
a	0.0	0.0	0.0	0.0	0.3	0.6	0.1
lot	0.0	0.2	0.4	0.3	0.0	0.05	0.05
home	0.0	0.5	0.0	0.1	0.3	0.0	0.1
</s>	1.0	0.0	0.0	0.0	0.0	0.0	0.0

Considering the following test set:

TEST: [*Tom talks home a lot. Home Tom talks. Home talks.*]

Show how you compute the perplexity of this bigram model smoothed with the Laplace method (**5 points**) and compare it to the perplexity of the language model when no smoothing was applied (**5 points**)

Problem 1 c): Solution:

For Bigram, $PP(w) = P(w_1 \cdots w_N)^{\frac{1}{N}} = \left[\prod_{i=2}^{N} \frac{1}{P(w_i|w_{i-1})}\right]^{\frac{1}{N}}$

TEST: <s> Tom talks home a lot </s> <s> Home Tom talks </s>
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10} w_{11} w_{12}
<s> Home talks </s>
w_{13} w_{14} w_{15} w_{16} $N=16$ ~~merge replicated items~~

$\therefore PP(w) = \Big[\Big(\frac{1}{P(Tom|<s>)} \cdot \frac{1}{P(talks|Tom)} \cdot \frac{1}{P(home|talks)} \cdot \frac{1}{P(a|home)} \cdot \frac{1}{P(lot|a)} \cdot \frac{1}{P(</s>|lot)}$
$\frac{1}{P(<s>|</s>)} \cdot \frac{1}{P(home|<s>)} \cdot \frac{1}{P(Tom|home)} \cdot \frac{1}{P(talks|Tom)} \cdot \frac{1}{P(</s>|talks)} \cdot \frac{1}{P(<s>|</s>)}$
$\frac{1}{P(home|<s>)} \cdot \frac{1}{P(talks|home)} \cdot \frac{1}{P(</s>|talks)}\Big]^{\frac{1}{20}}$

We can get all no smoothing probabilities from given table. But we need to calculate Laplace smoothing probabilities. Fortunatelly, we can merge some replicated items. Besides, $|V| = 25$

$P_{Laplace}(w_i) = \frac{C_i+1}{N+V}$, $P_{MLE}(w_i) = \frac{C_i}{N}$. (Use P_L short for $P_{Laplace}$ and P_M short for P_{MLE} below)

$\therefore P_L(w_i) = \frac{P_M(w_i)N+1}{N+V}$

In the final exam, please type your answers!
Your writeup is

VERY HARD to read!

$P_L(\text{Tom}|\text{<s>}) = \frac{0.7 \times 2500 + 1}{2500 + 25} = 0.693465$ $\quad P_L(\text{talks}|\text{Tom}) = \frac{0.75 \times 1500 + 1}{1500 + 25} = 0.738361$

$P_L(\text{home}|\text{talks}) = \frac{0.1 \times 2000 + 1}{2000 + 25} = 0.099259$ $\quad P_L(\text{a}|\text{home}) = \frac{0.1 \times 15000 + 1}{15000 + 25} = 0.099900$

$P_L(\text{lot}|\text{a}) = \frac{0.3 \times 1000 + 1}{1000 + 25} = 0.293658$ $\quad P_L(\text{</s>}|\text{lot}) = \frac{0.05 \times 400 + 1}{400 + 25} = 0.049412$

$P_L(\text{<s>}|\text{</s>}) = \frac{1.0 \times 2500 + 1}{2500 + 25} = 0.990495$ $\quad P_L(\text{home}|\text{<s>}) = \frac{0.1 \times 2500 + 1}{2500 + 25} = 0.099406$

$P_L(\text{Tom}|\text{home}) = \frac{0.5 \times 15000 + 1}{15000 + 25} = 0.499234$ $\quad P_L(\text{talks}|\text{Tom}) = \frac{0.75 \times 1500 + 1}{1500 + 25} = 0.738361$

$P_L(\text{</s>}|\text{talks}) = \frac{0.05 \times 2000 + 1}{2000 + 25} = 0.049876$ $\quad P_L(\text{<s>}|\text{</s>}) = 0.990495$

$P_L(\text{home}|\text{<s>}) = 0.099406$ $\quad P_L(\text{talks}|\text{home}) = \frac{0 \times 15000 + 1}{15000 + 25} = 0.000066$

$P(\text{</s>}|\text{talks}) = 0.049876$

$$PP_{Laplace}(w) = [0.693465 \times 0.738361 \times 0.099259 \times 0.099900 \times 0.293658 \times 0.049412 \times 0.990495 \cdot 0.099406 \times 0.499234 \times 0.738361 \times 0.049876 \times 0.990495 \times 0.099406 \times 0.000066 \times 0.049876]^{-\frac{1}{16}} = 6.843318$$

Since $P_M(\text{talks}|\text{home}) = 0$ $\quad$ [Connect to Problem 1 c)]

$$PP_{no\text{-}smoothing}(w) = [\tfrac{1}{0}]^{\frac{1}{16}} = \infty$$

$\therefore PP_{Laplace}(w) = 6.843318 \quad PP_{no\text{-}smoothing}(w) = \infty$ ✓

That is why we apply Laplace smoothing for calculating perplexity

We have $PP_{Laplace}(w) = 6.843318$ and $PP_{no\text{-}smoothing}(w) = \infty$. ✓

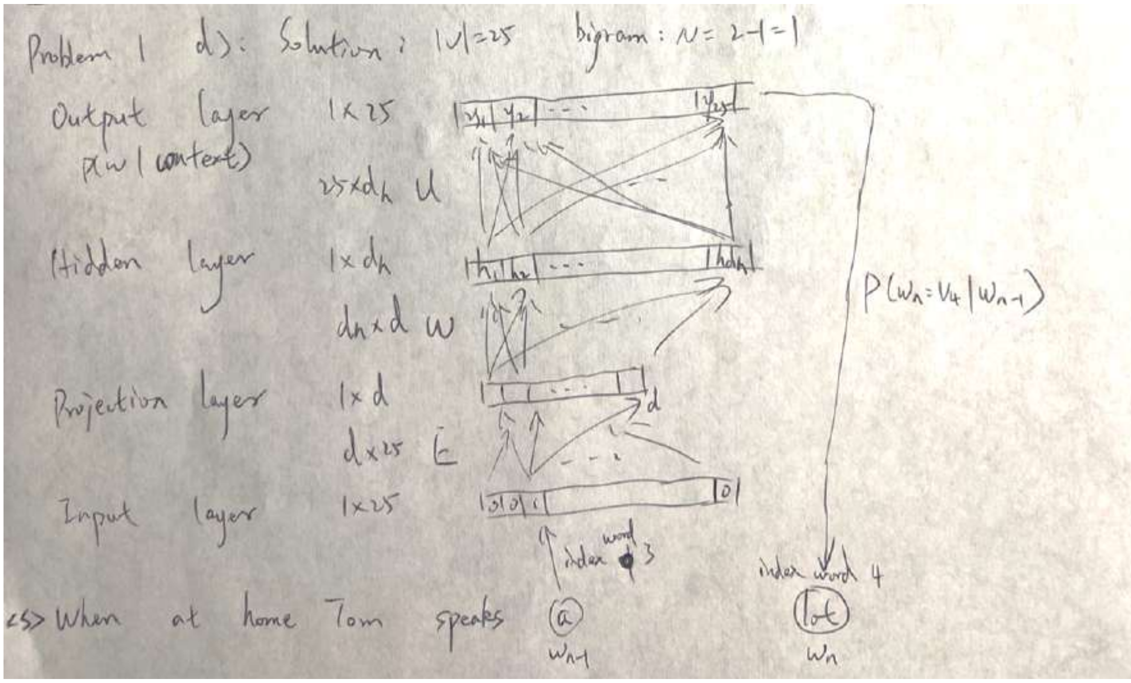
d) [**10 points**] Draw a non-recursive neural architecture that allows you to (1) learn a neural language model from the training set and (2) learn in the same time the embeddings of the words. Exemplify the neural architecture working on the word sequence "*When at home Tom speaks a*" and show how it will predict the word "*lot*". **[3 points]** Detail the parameters of the neural model and explain how they are learned. Write the equations that define the working of this neural architecture **[7 points]**

6/10

3

Which parameters?

3



Equations:

$$e_i = E x_i$$
$$h = \sigma(We+b)$$
$$z = Uh$$
$$y = \text{softmax}(z)$$

[Connect to Problem 1 d)]

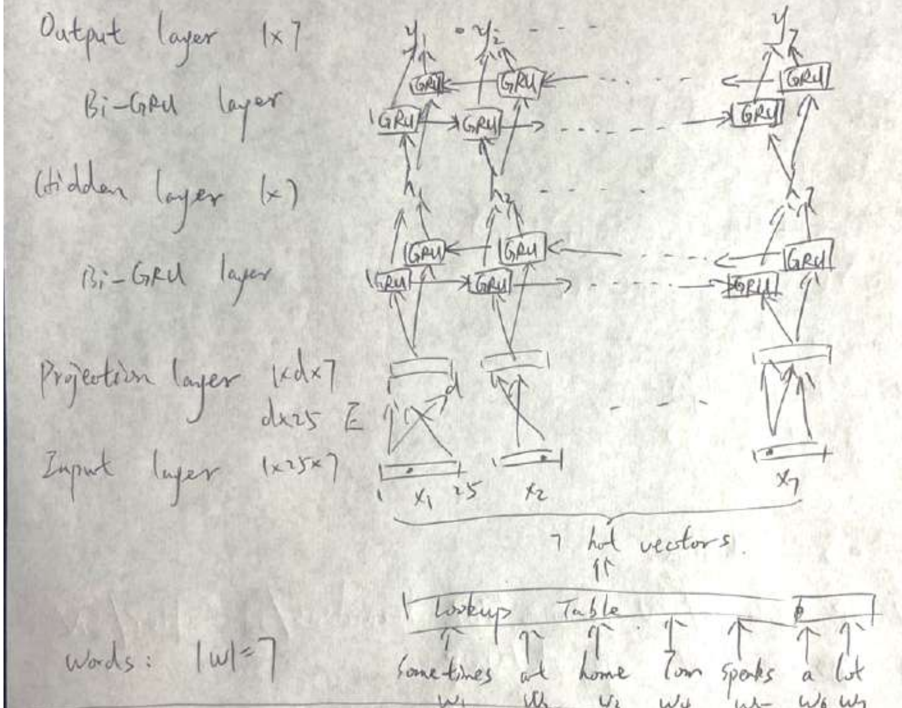
Equations ✓

e) [**15 points**] Draw a recursive neural architecture that uses two stacked bi-directional GRU layers for learning a neural language model that operates on the word sequence "*Sometimes at home Tom speaks a lot*". [**3 points]** Detail the parameters of the neural model and explain how teacher forcing can be used and why. [**5 points]** Write the equations that define the working of this neural architecture [**7 points]**

12

3 7 2/5

Problem 1 e: Solution:



Teacher Forcing:

$$L_{CE}(\hat{y}^t, y^t) = -\log \hat{y}^t_{w_{t+1}}$$

The correct distribution y comes from knowing the next word.

At time t, the CE loss is the negative log probability assigned to the next word in the training sequence.

Equations:

[Connects to Question 1 e]

$x_i = P_i \times W_{embedding}$

$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$

$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$

$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$

$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$

$h_t = w^t h^t_m + w^b h^b_o$

~~$r_{t_2} = \sigma(W_r h_t +$~~

$r_{t_2} = \sigma(W_{r_2} h_t + U_r h_{t-1_2} + b_{r_2})$

$z_{t_2} = \sigma(W_{z_2} h_t + U_z h_{t-1_2} + b_{z_2})$

$\hat{h}_{t_2} = \tanh(W_{h_2} h_t + U_{h_2}(r_{t_2} \odot h_{t-1_2}) + b_{h_2})$

$h_{t_2} = (1 - z_{t_2}) \odot h_{t-1_2} + z_{t_2} \odot \hat{h}_{t_2}$

$e_t = w^t_2 h^t_{m_2} + w^b_2 h^b_{o_2}$

$$P(y=j|e) = \frac{\exp(w_j^T e + b_j)}{\sum_{i=1}^{k} \exp(w_i^T e + b_i)}$$

very hard to read!

Problem 2 (POS Tagging) (**30 points**)

a) (**5 points**) Using the Penn Treebank Part-of-Speech (POS) tag-set, manually assign tags for the following sentences:

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

S1: *Iceland volcano remains hazardous after eruption near Reykjavik.*

S2: *Concerns were raised about the proximity of the volcano to the country's main airport, Keflavik International Airport, which is just a 25 min car ride from the peninsula.*

To annotate the POS tags, you may use the following format:

John/NNP and/CC Mary/NNP bought/VBD a/DT refrigerator/NN with/IN three/CD doors/NNS

S1: (**2 points**)

Problem 2• a): Solution:
S1: Iceland/NNP volcano/NN remains/VBZ hazardous/JJ after/IN eruption/NN ~~after~~ near/IN Reykjavik/NNP.

2

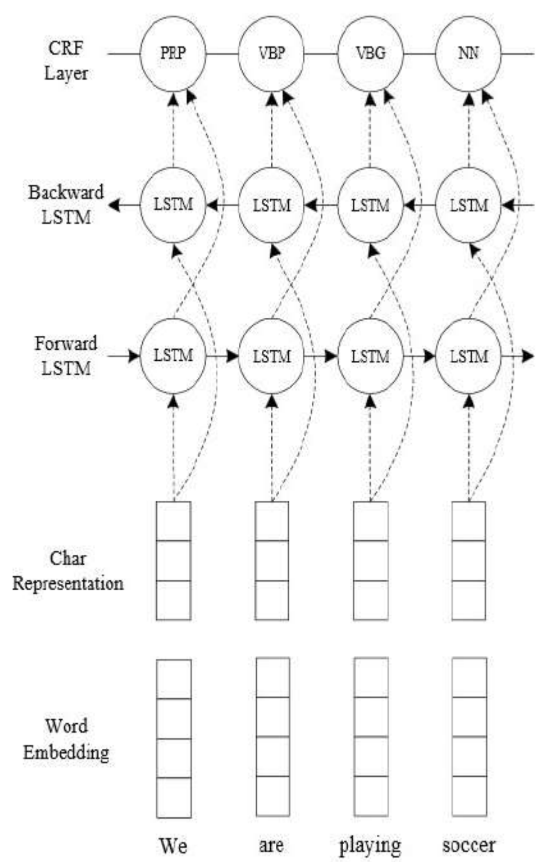
S2: (**3 points**)

3

S2: Concerns/NNS were/VBD raised/VBN about/IN the/DT proximity/NN of/IN the/DT volcano/NN to/IN the/DT country/NN 's/POS main/JJ airport/NN. Keflavik/NNP International/NNP Airport/NNP, which/WDT is/VBZ just/RB a/DT 25/CD min/NN car/NN ride/NN from/IN the/DT peninsula/NN.

b) Neural POS Tagging (**10 points**)

The architecture of a Neural Conditional Random Field (CRF) used for Part-of-Speech (POS) Tagging is represented below (where word embeddings and character embeddings are concatenated:



3/5

Explain how the POS tags are learned to be assigned. (**5 points**) What parameters does this neural architecture have and explain how are they used and learned? (**5 points**) 0

3/10

Question 2 b): Solution:

First, words are transformed into hot vectors by word embeddings and characters embeddings.

Then, with bi-directional LSTM layers. we did not compute the probability for each tag at each step. Instead, we used log-linear functions to get a global probability for the whole sequence.

impossible to read!

Possible tag sequences: $\hat{T} = \underset{T}{\arg\max}\, P(T|w)$

$$P(Y|X) = \frac{\exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)}{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)}$$

Linear Chain CRF:

$$F_k(X,Y) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)$$

current output token y_i
previous output token y_{i-1}
input string X
current position i

Then, we have:

$$\hat{T} = \underset{Y \in \mathcal{Y}}{\arg\max} \sum_{i=1}^{n} \sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i, X, i)$$

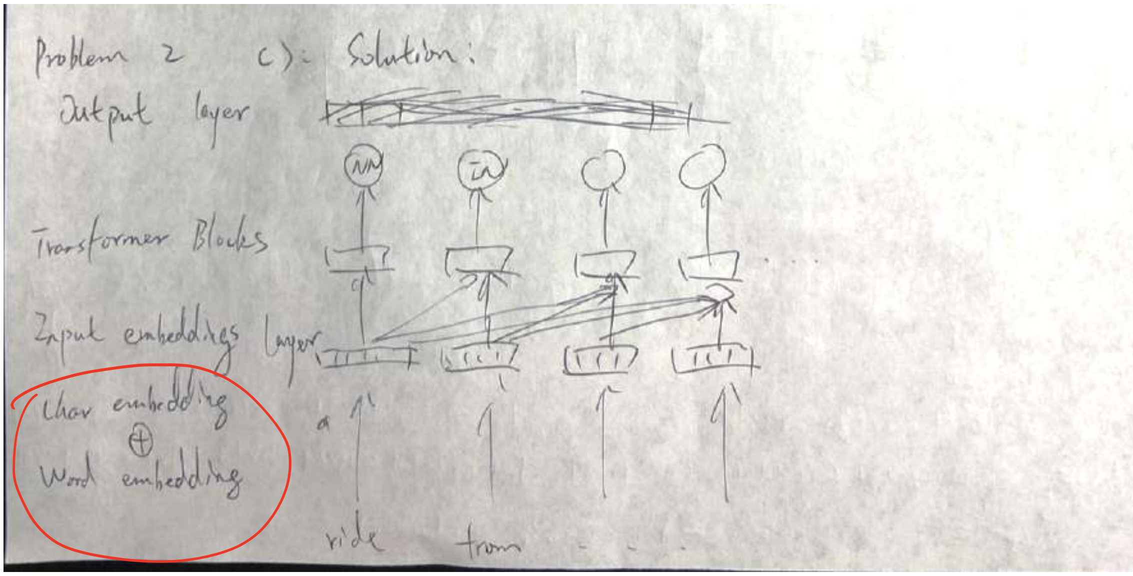
=) What are the parameters? How do you learn them?

c) Propose a transformer-based architecture for performing POS tagging. Draw the architecture, give the equations and discuss how the self-attention is computed (**10 points**). How do you implement and represent the input to your neural architecture?
(5 **points**). 0

6/10

6/15

incorrect



Equations and input solution:

[connect to Problem 2 c]

$Q = W^Q X$

$K = W^K X$

$V = W^V X$

$\mathrm{SelfAttention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$

Use char embedding and word embedding to firstly transform word into ~~hot vectors~~ embedding vectors for input embedding layer.

incorrect

Problem 3 (Affect) (**20 points**)

The SentProp Algorithm has computed the following Positive and Negative scores for the polarity of words:

LEXICON: **{** (change: w^{pos}=0.66 w^{neg}=0.34); (hate: w^{pos}=0.22, w^{neg}=0.78), (love: w^{pos}=0.75, w^{neg}=0.25); (opposite: w^{pos}=0.27, w^{neg}=0.63), (suffer: w^{pos}=0.25, w^{neg}=0.75); (thrive: w^{pos}=0.88, w^{neg}=0.12), (want: w^{pos}=0.69, w^{neg}=0.31), (wish: w^{pos}=0.55, w^{neg}=0.45)**}**.

a) Given this lexicon, determine the sentiment of the following text and explain providing details how you obtained the resulting sentiment, considering that the threshold you will consider in your computation is T=0.55 [**10 points**]

TEXT: *Love and hate are similar in being directed toward another person because of who he or she is. Despite this similarity, the two seem like polar opposites. Very often when we love someone, we want them to thrive. When we hate someone, we are more likely to wish they would suffer — or at least change who they are. If she loves you, it doesn't mean she is not capable of hate. If you hate him now and suffer because of it, it does not mean you never loved him.*

Solution:

Problem 3 a): Solution

	love
count	4
w^{pos}	0.75
w^{neg}	0.25

change	hate	love	opposite	suffer	thrive	want	wish
1	4	4	1	2	1	1	1

$f^+ = 0.66\times1 + 0.22\times4 + 0.75\times4 + 0.27\times1 + 0.25\times2 + 0.88\times1 + 0.69\times1 + 0.55\times1 = 7.43$

$f^- = 0.34\times1 + 0.78\times4 + 0.25\times4 + 0.63\times1 + 0.75\times2 + 0.12\times1 + 0.31\times1 + 0.45\times1 = 7.47$

~~sentiment~~ $\frac{f^+}{f^-} = 0.994645 > 0.55 = T$

∴ sentiment = +

how about ~~a~~ $\frac{f^-}{f^+}$?

0.55

Positive

b) Whenever any word from the lexicon is within the scope of a negation (e.g. "<u>never</u> loved"), swap the values of the positive and negative weights and recompute the sentiment of the text. Show the details of the computation. [**10 points**]

<u>Solution:</u>

b) = Solution

	change	hate	love	opposite	suffer	thrive	want	wish
remain same	1	3	3	1	2	1	1	1
Swap value	0	1	1	0	0	0	0	0

$t^+ = 0.66\times1 + 0.22\times3 + 0.78\times1 + 1.75\times3 + 0.25\times1 + 0.27\times1 + 0.25\times2 + 0.88\times1 + 0.69\times1 + 0.55\times1 = 7.49$

$t^- = 0.34\times1 + 0.78\times3 + 0.22\times1 + 0.25\times3 + 0.75\times1 + 0.63\times1 + 0.75\times2 + 0.12\times1 + 0.31\times1 + 0.45\times1 = 7.41$

$t^+/t^- = 1.010796 > 0.55 = T$ $\quad\therefore$ sentiment $= +$

Positive