COL772 Autumn 2022 Major

Navya Jain

TOTAL POINTS

56.67 / 100

QUESTION 1

1 Objective Questions 2 / 11

✓ **- 1 pts** *a) wrong/missing*

✓ **- 1 pts** *b) wrong/missing*

✓ **- 1 pts** *c) wrong/missing*

✓ **- 1 pts** *d) wrong/missing*

- 1 pts e) wrong/missing

✓ **- 1 pts** *f) wrong/missing*

✓ **- 1 pts** *g) wrong/missing*

✓ **- 1 pts** *h) wrong/missing*

- 1 pts i) wrong/missing

✓ **- 1 pts** *j) wrong/missing*

✓ **- 1 pts** *k) wrong/missing*

- 0.5 pts Partial Grade for k)

- 0.5 pts Partial Grade for g)

QUESTION 2

2 SVD 4.5 / 5

- 1 pts wrong/missing calculation of embedding of Star Wars

- 1 pts wrong/missing calculation of embedding of Jill

- 0 pts Complete Correct Solution.

- 3 pts Wrong Matrix Formation for reconstructed embedding.

- 0.5 pts Partially correct embeddings for Star Wars.

✓ **- 0.5 pts** *Partially correct embedding for Jill*

- 5 pts Missing Solution/completely wrong solution.

QUESTION 3

3 Macro-Scores 2 / 5

+ 5 pts Correct

+ 1.5 pts macro-prec.

+ 1.5 pts macro-recall

+ 1 pts compute macro-f1 (any way)

+ 1 pts show other way of macro-f1

+ 0 pts incorrect

+ 2 *Point adjustment*

swapped recall with precision. where is final answer?

QUESTION 4

4 Word2Vec Tasks 0 / 7

✓ **- 2 pts** *incorrect part (a)*

- 1 pts incorrect part (a)

- 2 pts part (b): no separation of syntax and semantics

- 2 pts part (b) information about corpus present in old embeddings not getting retained much, embeddings may change a lot

✓ **- 5 pts** *part (b): no solution presented*

- 1 pts how to choose hyperparameters?

QUESTION 5

5 RNN 8.67 / 12

+ **12 pts** Correct

✓ **+ 3 pts** *a) 0.5 for each param*

✓ **+ 4 pts** *b) 0.33 for each param*

✓ **+ 5 pts** *c) 0.4 for each param*

+ **0 pts** Incorrect

- 3.33 *Point adjustment*

QUESTION 6

6 Dynamic Programming 5.5 / 12

+ **0 pts** Not attempted.

✓ **+ 1 pts** *Correct dimension of the DP table*

✓ **+ 3 pts** *Correct definition of DP[i][j]*

✓ **+ 1 pts** *Correct base cases*

+ **3 pts** Correct recursive case

+ **2 pts** Backpointer correctly defined.

✓ **+ 0.5 pts** *Space complexity*

+ **1.5 pts** Time complexity

+ **0 pts** Incorrect

QUESTION 7

7 N-gram Language Models 13 / 13

+ **0 pts** Not attempted.

✓ **+ 2 pts** *Part a*

✓ **+ 2 pts** *Part b*

✓ **+ 2 pts** *Part c*

✓ **+ 2 pts** *Part d*

✓ **+ 3 pts** *Part e*

✓ **+ 2 pts** *Part f*

+ **0 pts** Incorrect

QUESTION 8

8 Data Augmentation 3 / 3

- **1 pts** Method 1 incorrect or missing.

- **1 pts** Method 2 incorrect or missing

- **1 pts** Method 3 incorrect or missing

- **0.5 pts** Method 1 partially correct.

- **0.5 pts** Method 2 partially correct

- **0.5 pts** Method 3 partially correct

- **3 pts** Missing/wrong solution

✓ **- 0 pts** *Correct Solution*

QUESTION 9

9 BERT for Classification 2.5 / 6

+ **6 pts** Correct

✓ **+ 2 pts** *(a)*

+ **2 pts** (b)

+ **2 pts** (c)

+ **0 pts** Click here to replace this description.

+ 0.5 *Point adjustment*

QUESTION 10

10 One-short Learning with GPT3 0 / 2

- **0.5 pts** Proper Example Sample not added.

- **1 pts** Proper Instructions not added (e.g. Interpret Compound Noun).

- **0.5 pts** Proper Special Symbol or Separator not included (e.g. => , :).

✓ **- 2 pts** *Missing Answer*

+ **0 pts** Correct Answer.

QUESTION 11

11 Probabilistic CFG 9.5 / 12

✓ **+ 1 pts** *chart has right dimensionality*

+ **1 pts** final parse prob correct

✓ **+ 1 pts** *final parse correct*

+ **3 pts** unaries correct

+ **3 pts** preterminals correct

✓ **+ 3 pts** *S and NP/VP for intermediate levels correct*

+ 0 pts not attempted

+ 4.5 *Point adjustment*

unaries -- two missing (1.5/3). preterminal one missing (2.5/3). final parse prob not identified (0.5/1)

QUESTION 12

12 Pay At00tention! 0 / 6

+ 6 pts Correct

✓ **+ 0 pts** *Not attempted.*

+ 0 pts Incorrect

+ 1 pts Correct answer but incorrect explanation

+ 0 pts Click here to replace this description.

QUESTION 13

13 Constraints in NLP 4 / 4

+ 0 pts Not attempted.

✓ **+ 2 pts** *Correct first part.*

+ 0 pts Incorrect

✓ **+ 2 pts** *Correct second part*

QUESTION 14

14 Most interesting topic 2 / 2

- 1 pts (most) reasoning is not technical/question not attempted

- 1 pts (least) reasoning is not technical/question not attempted

- 0.5 pts (most) reasoning not technical enough

- 0.5 pts (least) reasoning not technical enough

✓ **- 0 pts** *OK*

1. [11 points] Objective Questions: Fill in the blanks with the best linguistic/NLP/ML term.

(a) The field of NLP that studies meaning of a sentence in the context of the sentence (e.g., a physical scene, or an image) is called _____.

(b) If a verb has two direct objects (e.g., teach in "Prof. Prasad teaches him programming languages"), it is called a ditransitive verb.

(c) homonym is the concept in linguistics where one word acquires two different meanings, where the second meaning is an attribute of the first (e.g., one word 'rose' is used in both flower and color wordsense).

(d) The process that creates new words as per the comic below is called orthographic ?.

(e) The word 'दयालु' (dayaalu, *kind*) is created when a/an suffix is applied to the word 'दया' (dayaa, *kindness*).

(f) The NLP task that converts a sentence/question into a logical expression (e.g., "who is the president of USA" to "select x from people where President(x, USA)" is called logical re.

(g) The NLP task that takes the passage, "I wrote the document in latex", and identifies that 'latex' is used for its software meaning and not the other meaning of 'substance used for making paint' is called contextual.

(h) A recent pre-trained language model that specializes in conversational dialogs is called T5.

(i) The neural component that re-computes the value of an input feature based on the distribution of values of the same feature in the given minibatch is called batch normalization.

Best
GPT3
T5
BART
XLNet

(j) The challenge that it is hard to represent a large number of world's languages in a pre-trained language model because of limited model capacity is called mBert 100.

(k) A prompting technique that elicits reasoning capabilities from a pre-trained language model like GPT3 is called auto prompting.

1 Objective Questions 2 / 11

✓ **- 1 pts** *a) wrong/missing*

✓ **- 1 pts** *b) wrong/missing*

✓ **- 1 pts** *c) wrong/missing*

✓ **- 1 pts** *d) wrong/missing*

- 1 pts e) wrong/missing

✓ **- 1 pts** *f) wrong/missing*

✓ **- 1 pts** *g) wrong/missing*

✓ **- 1 pts** *h) wrong/missing*

- 1 pts i) wrong/missing

✓ **- 1 pts** *j) wrong/missing*

✓ **- 1 pts** *k) wrong/missing*

- 0.5 pts Partial Grade for k)

- 0.5 pts Partial Grade for g)

Q2 contd embedding of Jill

$$\begin{bmatrix} 1.86 & -5.605 \end{bmatrix}_{7\times 2} \begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.65 \end{bmatrix}_{2\times 7}$$

Jill = [0.369, 1.2095, 0.369, 4.03, 3.81065]

2. [5 points] Singular Value Decomposition

Consider a word-document matrix consisting of 5 movies and 7 users (who rate the movies). In order to build an LSA model, an SVD of the matrix was performed as follows:

Name	Matrix	Alien	Star Wars	Casablanca	Titanic
Joe	1	1	1	0	0
Jim	3	3	3	0	0
John	4	4	4	0	0
Jack	5	5	5	0	0
→ Jill	0	2	0	4	4
Jenny	0	0	0	5	5
Jane	0	1	0	2	2

$$= \underbrace{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.01 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ -.40 & .80 & -.40 & .09 & .09 \end{bmatrix}}_{V^T}$$

Compute the reconstructed embeddings of the movie *Star Wars,* and the user *Jill* using only the two most important latent concepts (*k=2*).

After considering 2 most imp concepts,

$$U = \begin{bmatrix} 0.13 & 0.02 \\ 0.41 & 0.07 \\ 0.55 & 0.09 \\ 0.68 & 0.11 \\ 0.15 & -0.59 \\ 0.07 & -0.73 \\ 0.07 & -0.29 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}$$

$$V^T = \begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \end{bmatrix}$$

$$E' = U \Sigma V^T = \begin{bmatrix} 0.13 & 0.02 \\ 0.41 & 0.07 \\ 0.55 & 0.09 \\ 0.68 & 0.11 \\ 0.15 & -0.59 \\ 0.07 & -0.73 \\ 0.07 & -0.29 \end{bmatrix}_{7\times 2} \begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}_{2\times 2}$$

$$= \begin{bmatrix} 1.612 & 0.19 \\ 5.084 & 0.665 \\ 6.82 & 0.855 \\ 8.432 & 1.045 \\ 1.86 & -5.605 \\ 0.868 & -6.935 \\ 0.868 & -2.755 \end{bmatrix}_{7\times 2} \begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \end{bmatrix}_{2\times 5}$$

embedding of Star Wars ↳ 3rd col.

$$\begin{bmatrix} 1.612 & 0.19 \\ 5.084 & 0.665 \\ 6.82 & 0.855 \\ 8.432 & 1.045 \\ 1.86 & -5.605 \\ 0.868 & -6.935 \\ 0.868 & -2.755 \end{bmatrix}_{7\times 2} \begin{bmatrix} 0.56 \\ 0.12 \end{bmatrix}_{2\times 1} = \begin{bmatrix} 0.92552 \\ 2.9268 \\ 3.9218 \\ 4.84732 \\ 0.369 \\ -0.34612 \\ 0.15548 \end{bmatrix}$$

2 SVD **4.5 / 5**

- **1 pts** wrong/missing calculation of embedding of Star Wars
- **1 pts** wrong/missing calculation of embedding of Jill
- **0 pts** Complete Correct Solution.
- **3 pts** Wrong Matrix Formation for reconstructed embedding.
- **0.5 pts** Partially correct embeddings for Star Wars.

✓ **- 0.5 pts** *Partially correct embedding for Jill*

- **5 pts** Missing Solution/completely wrong solution.

3. [5 points] Macro-Scores

In this task, people are automatically extracted from a collection of documents and classified with a label (left, right or center) or are left unclassified. Compute macro-precision, macro-recall and macro-F scores. Can you see two alternate ways of computing macro-F scores? What are they?

Example	System Output	Answer Key
Barack Obama	Center ✓	Left
Bernie Sanders	Center ✓	Left
Bill Clinton	Right ✓	Left
Bugs Bunny	Center ✓	
Donald Trump	Right ✓	Right →
Elon Musk	Right ✓	Center
Hillary Clinton	Left ✓	Left
Howard Stern		Center
Joe Biden	Left ✓	Left
Karl Marx	Left ✓	
Mitt Romney	Right ✓	Center
Noam Chomsky	Center ✓	Center
Parag Agrawal		Left
Rush Limbaugh	Right ✓	Right
Sarah Palin	Center ✓	Right
Satya Nadella	Right ✓	
Sundar Pichai	Center ✓	Center
Yoshua Bengio	Left ✓	

actual

TP (center) → 6

actual

TP (left) → 4

actual

TP (right) → 6.

TP (right) → 2

TP (center) → 2

TP (left) → 2

$$\text{macro prec} = \frac{\left(\frac{2}{5} + \frac{2}{3} + \frac{2}{6}\right)}{3}$$

$$\text{macro rec} = \frac{\left(\frac{2}{6} + \frac{2}{6} + \frac{2}{4}\right)}{3}$$

$$\text{macro F1} \Rightarrow \frac{\text{macro prec} \times \text{macro rec}}{\text{macro prec} + \text{macro rec}}$$

3 Macro-Scores **2 / 5**

- **+ 5 pts** Correct
- **+ 1.5 pts** macro-prec.
- **+ 1.5 pts** macro-recall
- **+ 1 pts** compute macro-f1 (any way)
- **+ 1 pts** show other way of macro-f1
- **+ 0 pts** incorrect

+ 2 *Point adjustment*

swapped recall with precision. where is final answer?

4 Word2Vec Tasks 0 / 7

✓ **- 2 pts** *incorrect part (a)*

- 1 pts incorrect part (a)

- 2 pts part (b): no separation of syntax and semantics

- 2 pts part (b) information about corpus present in old embeddings not getting retained much, embeddings may change a lot

✓ **- 5 pts** *part (b): no solution presented*

- 1 pts how to choose hyperparameters?

5. [12 points] Recurrent Neural Networks

We define a simple GRU as follows:

$$s_j = z_j \odot s_{j-1} + (1 - z_j) \odot \tilde{s_j}$$
$$z_j = g[x_j W^{xz} + s_{j-1} W^{sz} + b^z]$$
$$r_j = g[x_j W^{xr} + s_{j-1} W^{sr} + b^r]$$
$$\tilde{s_j} = g[x_j W^{xs} + (r_j \odot s_{j-1}) W^{sg} + b^s]$$

For consistent notation let, the basic sequential Elman's RNN be $s_j = g[x_j W^{xs} + s_{j-1} W^{sg} + b^s]$. Here the g activations can be any of the following standard functions: linear, ReLU, perceptron, sigmoid or tanh. Let output $o_j = s_j W^{os} + b^o$, for both basic RNN and GRU. All the following questions concern 1 dimensional GRUs and RNNs, i.e., input, hidden state are all scalars.

(a) [3 points] Let's say we are given input sequence of 0s and 1s. Our goal is to output the number of 0s minus the number of 1s seen in the sequence so far. So, if the input is 001111010, then the output should be 1 2 1 0 -1 -2 -1 -2 -1. Let s_0=0. Provide values for $g, W^{xs}, W^{sg}, W^{os}, b^o$ and b^s such that the Elman's RNN can replicate, if possible, this behavior. Show your work.

(b) [4 points] For our simple GRU provide values for all parameters and g that would allow it to replicate, if possible, the behavior described in part (a). Show your work.

(c) [5 points] We are now given the same input sequence, and must produce an output sequence that resets the count to zero when it sees the input 1 and outputs the number of zeros from that point onwards. For example, the input sequence 0011000100 should produce 1 2 0 0 1 2 3 0 1 2. Let s_0=0. Provide values for all parameters and g that will allow the GRU to replicate, if possible, this behavior. Show your work.

a) (output number of 0s − number of 1s)

$x \rightarrow$ 0 0 1 1 1 1 0 1 0 .

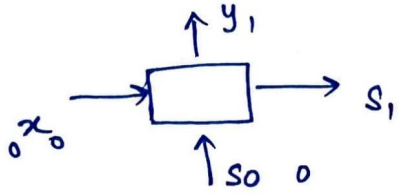
if model sees a '0' output 1 if it sees a 1 output '-1'

$$s_j = g(x_j w^{xs} + s_{j-1} w^{sg} + b^s)$$

let $s_0 = 0$.

let $b^s = 0$

$$= g(x_j + s_{j-1})$$



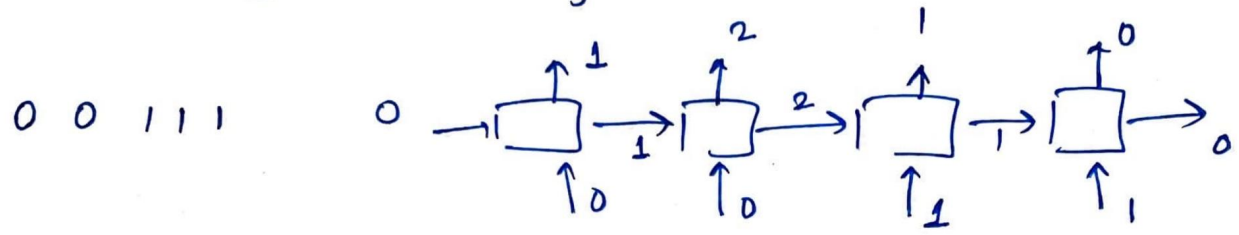
$y_1 = f(s_1)$

$y_1 = s_1$

$y_j = h(s_j)$

a) let $w_{xs} = -2$, $b = 1$ $w^{sg} = 1$ $w^{os} = 1$

$$\begin{bmatrix} s_j = -2x_j + 1 + s_{j-1} \\ o_j = s_j \end{bmatrix}.$$



b) let $g_1(x) = 1$ if $x > 0$

$= 0$ if $x < 0$.

$g_2(x) = 0$ if $x > 0$

$= 1$ if $x < 0$.

$w_1 = 0 = w_2$ $b_1 = 1 \Rightarrow z_j = 0$

$w_3 = 0 = w_4$ $b_2 = 1 \Rightarrow x_j = 1$

$\tilde{s}_j = -2x_j + s_{j-1} + 1$ i.e. $g \to$ linear

$w_3 \to -2$

$b_3^* \to 1$

$1 \otimes s_{j-1} = s_{j-1}$

$o_j = s_j$

$s_j = \tilde{s}_j = -2x_j + s_{j-1} + 1$

$o_j = s_j$ (similar structure to RNN observed)

Q5 (c) $r_j \to 0$ if $x_j \to 1$

$$q_j = x_j + s_{j-1} \times 0 + (-1)$$

$$q_j = x_j - 1$$

All other structure remains same.

$s_{j-1} \to 0$ whenever $x_j \to 1$ and hence count depends only on x_j from now.

5 RNN **8.67 / 12**

+ 12 pts Correct

✓ **+ 3 pts** *a) 0.5 for each param*

✓ **+ 4 pts** *b) 0.33 for each param*

✓ **+ 5 pts** *c) 0.4 for each param*

+ 0 pts Incorrect

- 3.33 *Point adjustment*

6. [12 points] Dynamic Programming

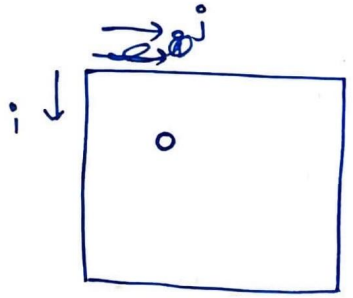
You are working on creating data for semantically similar sentences. You ask crowdworkers to replace words or phrases from input sentence to create a semantically similar sentence. You apply this replacements multiple times through different crowd workers to create a linguistically different but semantically similar sentence. For example this process, starting from sentence (s) "Several boys and girls are playing the game of cricket", may generate a sentence (t) "Children like to play sports such as cricket".

You believe that you can train a good model to identify semantically similar sentences if you first align phrases in s with those in t. Let the original sentence be composed of words ($s_1, s_2, \ldots, s_N$), and the target sentence be ($t_1, t_2, \ldots, t_M$). Say you pretrain a basic neural model (that has *tanh* as its last operation) that outputs a score $m(i,j,i',j')$ to reflect how well the phrase $s[i,j]$ matches with phrase $t[i',j']$. You define the joint score of aligner as the sum of scores of every aligned pair of phrases. Our goal is to write a dynamic program that finds the maximum scoring alignment that satisfies the following properties:

A) Every word in s is aligned to exactly one phrase in t
B) Every word in t is aligned to exactly one phrase in s
C) The alignment cannot have cross edges and maintains order, i.e., if $s[i1,j1]$ matched with $t[i1',j1']$ and $s[i2,j2]$ matched with $t[i2',j2']$ then $i1 < i2 \rightarrow j1' < i2'$. Moreover, $i1 < i2 \rightarrow j1 < i2$. These also imply that $i1' < i2'$.

For our running example, the best match will be ("Several boys and girls", "Children"), ("are playing", "like to play"), ("the game of", "sports such as"), ("cricket", "cricket").

(a) Specify a dynamic programming table for finding the best alignment. Indicate its dimensions and define an entry in a cell. Specify the cell that contains the score of the best alignment.
(b) Specify the base condition and the recursive case.
(c) Specify the equation for the backpointer.
(d) What is the time and space complexities of your dynamic program as a function of N and M.



s[i, j] t[i', j']

a) let dp[i,j] denote the max. score (aligner) of aligned pair of phrases in s[1...i] and t[1...j].

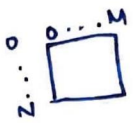
b) $dp[i,j] = \max \begin{cases} dp[i-1, j-1] + m(1, i, 1, j) & \text{ith word mapped to jth word.} \\ dp[i-1, j] + [m(1, i, 1, j) - m(1, i, 1, j-1)] & \text{ith word mapped to jth but not the first to map.} \\ dp[i, j-1] & \text{word i not mapped to word j} \end{cases}$

(subtract prev phrase score)

2) a) $dp[i,j]$ → max. alignment score of aligned pair of phrases in $s[1,\ldots i]$ and $t[1,\ldots j]$

dimensions → $(M+1)(N+1)$

best score → $dp[N\ M]$



b) contd.

base case $dp[0,0] = 0$

$dp[i,0] = 0$

$dp[0,j] = 0$

c) backpointer

if $\max(dp(i-1,j-1) + m(1,i,1,j),\ dp(i-1,j) + (m(1,i,1,j) - m(1,i,1,j-1)),\ dp(i,j-1)) =$

① $= dp[(i-1),j-1] + m(1,i,1,j)$ ↖

② $= dp[i-1,j] + m(1,i,1,j) - m(1,i,1,j-1))$ ↑

③ $dp(i,j-1))$ ←

d) Space → $O(NM)$

time → $O(NM)$

(given if score calculation takes linear time)

6 Dynamic Programming 5.5 / 12

+ **0 pts** Not attempted.

✓ **+ 1 pts** *Correct dimension of the DP table*

✓ **+ 3 pts** *Correct definition of DP[i][j]*

✓ **+ 1 pts** *Correct base cases*

+ **3 pts** Correct recursive case

+ **2 pts** Backpointer correctly defined.

✓ **+ 0.5 pts** *Space complexity*

+ **1.5 pts** Time complexity

+ **0 pts** Incorrect

7. [13 points] N-gram Language Models

You are given the following corpus:

<s> Life is good </s> — 5
<s> Life is always good </s> — 6
<s> Life is good as life always is vibrant </s> — 10
<s> Good is life </s> — 5
<s> As life always is vibrant life is good </s> — 10

You are trying to implement a bigram language model. Your vocabulary is all the words in the corpus (ignore capitalization). Do not worry about OOV words. *In your calculations, include <s> and </s> in your counts just like any other token.*

(a) [2 points] What is P(is | always) for an unsmoothed maximum-likelihood language model?
(b) [2 points] What is P(is | always) if we use absolute discounting of 0.5? Show your work.
(c) [2 points] What is P(is | always) if we use linear interpolation of unigram and bigram models with $\lambda_1 = \lambda_2 = 0.5$. Show your work.
(d) [2 points[What is P(is | always) if we use absolute discounting of 0.5 with linear interpolation of unigram and bigram models with $\lambda_1 = \lambda_2 = 0.5$. Show your work.
(e) [3 points] What is P(is | always) using Katz Backoff such that anything appearing two or more times is in the "seen" set and everything else is in the unseen set. Show your work.
(f) [2 points] What will be the best word to extend the sentence "<s> Life was" if only $P_{CONTINUATION}$ is used to pick the best word?

a) $P(\text{is} \mid \text{always}) = \frac{c(\text{always, is})}{c(\text{always})} = \frac{2}{3}$

b) $P^*(\text{is} \mid \text{always}) = \frac{c(\text{always, is}) - 0.5}{c(\text{always})} = \frac{2-0.5}{3} = 0.5$

c) $P(\text{is} \mid \text{always}) = \lambda_1 P(\text{is} \mid \text{always}) + \lambda_2 P(\text{is})$
$= (0.5)\left(\frac{2}{3}\right) + 0.5\left(\frac{7}{36}\right) = \frac{31}{72} = 0.43$

d) $P(\text{is} \mid \text{always}) = \lambda_1 P^*(\text{is} \mid \text{always}) + \lambda_2 P(\text{is})$
$= (0.5)(0.5) + 0.5\left(\frac{7}{36}\right) = \frac{25}{72} = 0.347$

e) seen set → $k \geq 2$ $A(v) := \{w : c(v,w) \geq k\}$ seen set.

seen set $A(\text{always}) = \{w : c(\text{always}, w) \geq 2\} = \{\text{is}\}$

unseen set $B(\text{always}) = \{w : c(\text{always}, w) < 2\} = \{\text{<s>, life, good, </s>, always, as, vibrant}\}$.

$$P(w_i|w_{i-1}) = \begin{cases} P^*(w_i|w_{i-1}) & \text{if seen set} \\ \alpha(w_{i-1})\, P(w_i) & \text{if unseen set} \end{cases}$$

$P(\text{is}|\text{always}) \Rightarrow$ seen set

$=$ { $P^*(w_i|w_{i-1}) \longrightarrow$ any prob discounted.

$\Rightarrow$ let us take abs discounting with $k' = 0.5$

$$\Rightarrow \quad P^*(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) - 0.5}{c(w_{i-1})} = \frac{2 - 0.5}{3}$$

$$= \underline{\underline{0.5}}$$

$$P^*(w_i, w_{i-1}) < \frac{2k}{3} \quad \text{where} \quad k \ll 1$$

f) uniq(*, <s>) = 0

uniq(*, life) = { <s>, as, is, vibrant } = 4

uniq(*, is) = { life, always, good } = 3

uniq(*, good) = { is, always, <s> } = 3

uniq(*, </s>) = { good, vibrant, life } = 3

uniq(*, always) = { is, life } = 2

uniq(*, as) = { good, <s> } = 2

uniq(*, vibrant) = { is } = 1

max occurrence as a word.

best word to extend $\longrightarrow$ life

LIFE.

7 N-gram Language Models 13 / 13

+ 0 pts Not attempted.

✓ **+ 2 pts** *Part a*

✓ **+ 2 pts** *Part b*

✓ **+ 2 pts** *Part c*

✓ **+ 2 pts** *Part d*

✓ **+ 3 pts** *Part e*

✓ **+ 2 pts** *Part f*

+ 0 pts Incorrect

8. [3 points] Data Augmentation

You wish to train a neural model for the fake news detection task, in which given a news article and background passage, the goal of the model is to verify if the news article is accurate (as per the passage) or not. The training data for this task is relatively small, and you wish to use data augmentation techniques to supplement the training data. Suggest any three techniques with brief explanations on how you will augment data.

① Back translation :- Translation text to some other language (say, hindi / french) and translate back.

② Paraphrase article :- (label won't change), ⟷ at sentence level synonym replacement, random insertion, deletion or swap. of words at token level

③ word replacement by language modelling :- mask a part/word of sentence, pass through BERT, predict word.

~~can also do a~~

9. [6 points] BERT for Classification

② BERT → context embeddings

We are given a primitive, which takes a sequence of strings s, runs BERT on it, to create a sequence of contextual embeddings BERT(s) of the same size as the original input. We wish to use BERT for the following tasks. For each of the tasks, suggest the input you will provide to BERT, and the how will you decode the output.

(a) Sentiment mining task where input is a sentence and output is one of the two classes: positive and negative.

~~[CLS] can~~ get embedding of string s → x.

concat

[CLS] x → run through BERT

~~CLS~~ run CLS through MLP to classify +ve & -ve.

get prob distribution → do a softmax.

8 Data Augmentation **3 / 3**

- **- 1 pts** Method 1 incorrect or missing.
- **- 1 pts** Method 2 incorrect or missing
- **- 1 pts** Method 3 incorrect or missing
- **- 0.5 pts** Method 1 partially correct.
- **- 0.5 pts** Method 2 partially correct
- **- 0.5 pts** Method 3 partially correct
- **- 3 pts** Missing/wrong solution

✓ **- 0 pts** *Correct Solution*

8. [3 points] Data Augmentation

You wish to train a neural model for the fake news detection task, in which given a news article and background passage, the goal of the model is to verify if the news article is accurate (as per the passage) or not. The training data for this task is relatively small, and you wish to use data augmentation techniques to supplement the training data. Suggest any three techniques with brief explanations on how you will augment data.

① Back translation :- Translation text to some other language (say, hindi / french) and translate back.

② Paraphrase article :- (label won't change), ⟷ at sentence level synonym replacement, random insertion, deletion or swap. of words at token level

③ word replacement by language modelling :- mask a part/word of sentence, pass through BERT, predict word.

~~can also do a~~

9. [6 points] BERT for Classification

② BERT → context embeddings

We are given a primitive, which takes a sequence of strings s, runs BERT on it, to create a sequence of contextual embeddings BERT(s) of the same size as the original input. We wish to use BERT for the following tasks. For each of the tasks, suggest the input you will provide to BERT, and the how will you decode the output.

(a) Sentiment mining task where input is a sentence and output is one of the two classes: positive and negative.

~~[CLS] can~~ get embedding of string s → x.

concat

[CLS] x → run through BERT

~~CLS~~ run CLS through MLP to classify +ve & -ve.

get prob distribution → do a softmax.

(b) Multiple choice question answering task, where input is a question and four choices, and the output is one of the four choices.
(c) Question answering task, where input is a passage, and a question, and the output is a span in the passage, which is the answer.

(b) [CLS$_q$] [question embedding] [SEP] [CLS$_2$] [choice 1] [SEP$_2$] [choice 2] ...

Input to BERT ..... train CLS, SEP$_1$, SEP$_2$, SEP$_3$, SEP$_4$,

CLS$_1$, CLS$_2$, CLS$_3$, CLS$_4$.

similarity b/w Question CLS and options CLS.

(c) Input [[Q] : Quest embedding [Sep] passage embedding]

⇓

output → attention b/w question embedding as query and passage embeddings to get most useful answer.

10. [2 points] One-shot learning with GPT3
We wish to use GPT3 for noun compound interpretation. We are given one training data point (olive oil, oil made from olives). Suggest a sample input to GPT3 that uses in-context learning and prompting to get an interpretation for "coffee mug".

{ , what what is in olive oil? or

9 BERT for Classification **2.5 / 6**

+ 6 pts Correct

✓ **+ 2 pts** *(a)*

+ 2 pts (b)

+ 2 pts (c)

+ 0 pts Click here to replace this description.

+ 0.5 *Point adjustment*

(b) Multiple choice question answering task, where input is a question and four choices, and the output is one of the four choices.
(c) Question answering task, where input is a passage, and a question, and the output is a span in the passage, which is the answer.

(b) [CLS$_q$] [question embedding] [SEP] [CLS$_2$] [choice 1] [SEP$_2$] [choice 2] ...

Input to BERT ..... train CLS, SEP$_1$, SEP$_2$, SEP$_3$, SEP$_4$,

CLS$_1$, CLS$_2$, CLS$_3$, CLS$_4$.

similarity b/w Question CLS and options CLS.

(c) Input [[Q] : Quest embedding [Sep] passage embedding]

⇓

output → attention b/w question embedding as query and passage embeddings to get most useful answer.

10. [2 points] One-shot learning with GPT3
We wish to use GPT3 for noun compound interpretation. We are given one training data point (olive oil, oil made from olives). Suggest a sample input to GPT3 that uses in-context learning and prompting to get an interpretation for "coffee mug".

{ , what what is in olive oil? or

10 One-short Learning with GPT3 **0 / 2**

- **0.5 pts** Proper Example Sample not added.
- **1 pts** Proper Instructions not added (e.g. Interpret Compound Noun).
- **0.5 pts** Proper Special Symbol or Separator not included (e.g. => , :).

✓ **- 2 pts** *Missing Answer*

- **0 pts** Correct Answer.

11. [12 points] Probabilistic Context Free Grammar

Consider the following grammar

S → NP VP [1.0]	VP → V ADVP [0.3]	DT → the [0.9]	V → rain [0.01]
NP → N [0.2]	VP → V [0.7]	N → rain [0.1]	V → rains [0.1]
NP → DT N [0.8]	ADVP → ADV [1.0]	N → rains [0.05]	ADV → down [0.1]

Consider the sentence "the rain rains down". Parse the sentence and output the most likely parse tree, as well as, the probability of parse tree. Show your work: the final Extended CKY chart, and all calculations done for each cell. (some preterminal rules, which are not relevant, are not shown)

0 the 1 rain 2 rains 3 down 4

5.04×10^{-3}

0.00504

the . 1.4×10^{-3}.

1-3 = 1-2 23 [V,VP]

V V ∝
N V ∝
NP V ∝

V VP ∝
N VP ∝
NP VP →

	01	02	03	04
DT →				
		12	13	14
			23	24
				34

0-2 = 01 + 12

→ DT V ∝
DT N →
DT NP

	01	02	03	04
DT → the (0.9)	NP → DT N (0.072)		S → NP VP (0.005)	S → NP VP (0.000 216)
	V → rain (0.01) N → rain (0.1) NP → N → rain (0.02)	12	S → NP VP (0.0014)	S → NP VP (6×10⁻⁵)
	VP		V → rains (0.1) VP → V → rain (0.07)	VP → V ADVP (0.003)
			N NP	ADV → down (0.1) ADVP → ADV (0.1)
				34

14 12 13 24

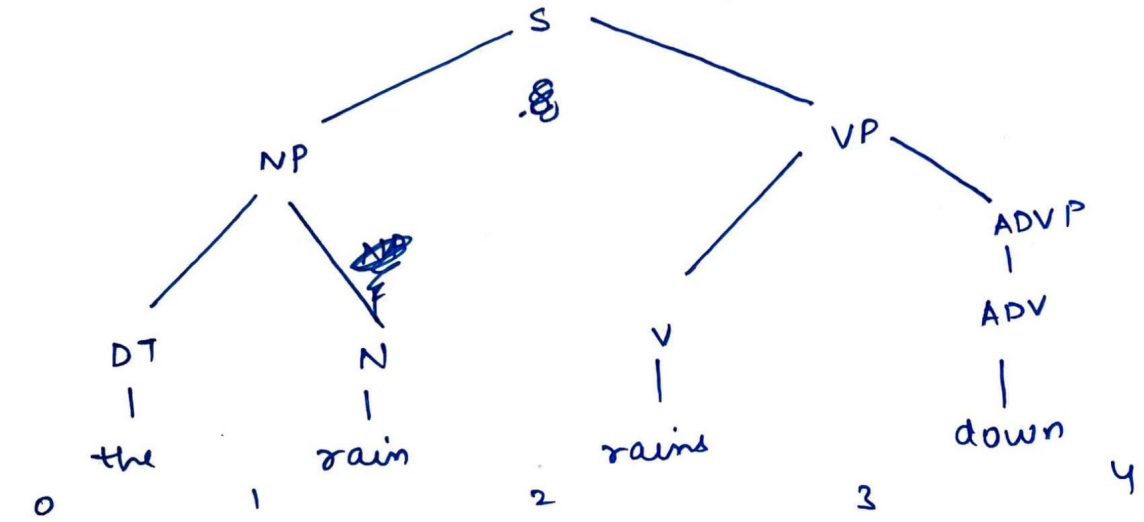
0-3
01 13
02 23.

DT S.
NP V 02 23
NP VP.

24
2-3 34.
V ADV
VP ADVP

1-4				
1-2	2-4		1-3	3-4
V	VP		S	
N	(0.003)			
NP				
(0.02)				

0-4							
0 1	1 4		0 2	2 4		0 3	3 4
DT	S		NP	VP		S	ADV
							ADVP



11 Probabilistic CFG **9.5 / 12**

✓ **+ 1 pts** *chart has right dimensionality*

+ 1 pts final parse prob correct

✓ **+ 1 pts** *final parse correct*

+ 3 pts unaries correct

+ 3 pts preterminals correct

✓ **+ 3 pts** *S and NP/VP for intermediate levels correct*

+ 0 pts not attempted

+ 4.5 *Point adjustment*

unaries -- two missing (1.5/3). preterminal one missing (2.5/3). final parse prob not identified (0.5/1)

12 Pay At00tention! **0 / 6**

+ 6 pts Correct

✓ **+ 0 pts** *Not attempted.*

+ 0 pts Incorrect

+ 1 pts Correct answer but incorrect explanation

+ 0 pts Click here to replace this description.

case ② :- Verb set = {V} ⇒ A
Prec. set = {NN, NNS, NNP, RB, IN} ⇒ B

$L(w,\lambda) = L(w) + \lambda \left[\sum_i \left[\min\left(0, \max\left(0, \max_{\forall l \in B} P_{w_i}(l) - \max_{\forall l \in P-B} P_{w_i}(l)\right)\right) * \min\left(0, \max\left(0, \max_{\forall l \in A} P_{w_i}(l) - \max_{\forall l \in P-A} P_{w_i}(l)\right)\right)\right]$

⇒ if 'i' is verb set, loss forces preceding to be from B set else

13. [4 points] Constraints in NLP

You are training a sequence labeling model for POS tagging using a Transformer. The input is the sentence $s=w_{1..N}$ and the output is a probability distribution $p_i(l)$ that represents the probability that POS for word w_i is l. Let the gold label for w_i be y_i. Moreover, you wish to add two additional (soft) constraints: (1) a sentence always has an equal number of opening and closing bracket/parentheses symbols (POS tag: OB and CB, respectively), and (2) a verb is always preceded by either a noun (NN, NNS or NNP), or an adverb (RB), or a preposition (IN). Devise a complete loss function for the task. If you introduce any hyperparameters, mention which ones.

$L(w)$ = Binary cross entropy loss function.

$L(w,\lambda) = L(w) + \lambda f_k$.

Case ① : equal no. of opening & closing parentheses. (count all openings & closings, subtract, make the one with the most equal, if not equal, less diff in prob)

let $P \in \{OB, CB\}$ → set of parentheses
$V \in$ {all tags}.

$P_1 = \{OB\}$
$P_2 = \{CB\}$ required tag

let $h(x) = 1$ if $x > 0$
$= 0$ if $x < 0$.

define $g(w) = h\left[\left(\max_{\forall l \in P_1} P_w(l) - \max_{\forall l \in P-V} P_w(l)\right)\right]$

for each w, g(w) tells (1 or 0) if the word is

$L(w,\lambda) = L(w) + \lambda \sum_i \max\left(0, \left| \sum_{w_i} \left(\max_{\forall l \in P_1} - \max_{\forall l \in V-P_2}\right)\right.\right.$

$\max\left(0, \left| \sum_{w_i} \max\left(0, \min\left(\max_{\forall l \in P_1} P_w(l)\right) - \max_{\forall l \in V-P} (P(w)\, l) - \sum_{w_i} \max\left(0, \min\left(\max_{\forall l \in P_2} (P_w(l)\right) - \max_{\forall l \in V-P} (P_w(l)\right.\right.\right.\right.$

14. [2 points] Which topic of the course was the most interesting to you and why? Please provide a technical reason – do not give reasons like we taught it well, or it is very popular, or easy to code. What about the least interesting topic and its reasons?

→ Attention was the most interesting topic for me, just one idea changed how word embeddings are perceived and gave a new direction for contextual embeddings.

→ Dependency Parsing was a bit uninteresting for me since it was based upon a lot of rules and did not add much value to NLP tasks.

(13) case (2) :- Verb set = {V} ⇒ A
Prec. set = {NN, NNS, NNP, RB, IN} ⇒ B

$L(w,\lambda) = L(w) + \lambda \left[\sum_{w_i} \left[\min\left(0, \max\left(0, \max_{\forall l \in B} P_{w_{i-1}}(l) - \max_{\forall l \in P-B} P_{w_i}(l)\right)\right) * \min\left(0, \max\left(0, \max_{\forall l \in A} P_{w_i}(l) - \max_{\forall l \in P-A} P_{w_i}(l)\right)\right)\right]\right]$

⇒ if 'i' is verb set, loss forces preceding to be from B set else

13. [4 points] Constraints in NLP

You are training a sequence labeling model for POS tagging using a Transformer. The input is the sentence $s=w_{1..N}$ and the output is a probability distribution $p_i(l)$ that represents the probability that POS for word w_i is l. Let the gold label for w_i be y_i. Moreover, you wish to add two additional (soft) constraints: (1) a sentence always has an equal number of opening and closing bracket/parentheses symbols (POS tag: OB and CB, respectively), and (2) a verb is always preceded by either a noun (NN, NNS or NNP), or an adverb (RB), or a preposition (IN). Devise a complete loss function for the task. If you introduce any hyperparameters, mention which ones.

L(w) = Binary cross entropy loss function.

$L(w,\lambda) = L(w) + \lambda f_k$.

Case (1) : equal no. of opening & closing parentheses. (count all opening & closing, subtract, if not equal, make the one with the most less diff in prob near to required tag)

Let $P \in \{OB, CB\}$ → set of parentheses
$V \in$ {all tags}.

$P_1 = \{OB\}$
$P_2 = \{CB\}$

Let $h(x) = 1$ if $x > 0$
$= 0$ if $x < 0$.

define $g(w) = h\left[\left(\max_{\forall l \in P_1} P_w(l) - \max_{\forall l \in P-V} P_w(l)\right)\right]$

for each w, g(w) tells (1 or 0) if the word is

$L(w,\lambda) = L(w) + \lambda \max\left(0, \left| \sum_{w_i} \left(\max_{\forall l \in P_1} - \max_{\forall l \in V-P_2}\right)\right.\right.$

$\max\left(0, \left| \sum_{w_i} \max\left(0, \min\left(\max_{\forall l \in P_1}(P_w(l)) - \max_{\forall l \in V-P}(P(w)\, l\right) - \sum_{w_i} \max\left(0, \min\left(\max_{\forall l \in P_2}(P_w(l)) - \max_{\forall l \in V-P}(P_w(l)\right)\right.\right.\right.$

14. [2 points] Which topic of the course was the most interesting to you and why? Please provide a technical reason – do not give reasons like we taught it well, or it is very popular, or easy to code. What about the least interesting topic and its reasons?

→ Attention was the most interesting topic for me, just one idea changed how word embeddings are perceived and gave a new direction for contextual embeddings.

→ Dependency Parsing was a bit uninteresting for me since it was based upon a set of rules and did not add much value to NLP tasks.

13 Constraints in NLP **4 / 4**

+ 0 pts Not attempted.

✓ **+ 2 pts** *Correct first part.*

+ 0 pts Incorrect

✓ **+ 2 pts** *Correct second part*

case ② :- Verb set = {V} ⇒ A
Prec. set = {NN, NNS, NNP, RB, IN} ⇒ B

$L(w,\lambda) = L(w) + \lambda \left[\sum_i \left[\min\left(0, \max\left(0, \max_{\forall l \in B} P_{w_i}(l) - \max_{\forall l \in P-B} P_{w_i}(l)\right)\right) * \min\left(0, \max\left(0, \max_{\forall l \in A} P_{w_i}(l) - \max_{\forall l \in P-A} P_{w_i}(l)\right)\right)\right]$

⇒ if 'i' is verb set, loss forces preceding to be from B set else

13. [4 points] Constraints in NLP

You are training a sequence labeling model for POS tagging using a Transformer. The input is the sentence $s=w_{1..N}$ and the output is a probability distribution $p_i(l)$ that represents the probability that POS for word w_i is l. Let the gold label for w_i be y_i. Moreover, you wish to add two additional (soft) constraints: (1) a sentence always has an equal number of opening and closing bracket/parentheses symbols (POS tag: OB and CB, respectively), and (2) a verb is always preceded by either a noun (NN, NNS or NNP), or an adverb (RB), or a preposition (IN). Devise a complete loss function for the task. If you introduce any hyperparameters, mention which ones.

$L(w)$ = Binary cross entropy loss function.

$L(w,\lambda) = L(w) + \lambda f_k$.

Case ① : equal no. of opening & closing parentheses. (count all openings & closings, subtract, make the one with the most equal, if not equal, less diff in prob)

let $P \in \{OB, CB\}$ → set of parentheses
$V \in$ {all tags}.

$P_1 = \{OB\}$
$P_2 = \{CB\}$ required tag

let $h(x) = 1$ if $x > 0$
$= 0$ if $x < 0$.

define $g(w) = h\left[\left(\max_{\forall l \in P_1} P_w(l) - \max_{\forall l \in P-V} P_w(l)\right)\right]$

for each w, g(w) tells (1 or 0) if the word is

$L(w,\lambda) = L(w) + \lambda \sum_i \max\left(0, \left| \sum_{w_i} \left(\max_{\forall l \in P_1} - \max_{\forall l \in V-P_2}\right)\right.\right.$

$\max\left(0, \left| \sum_{w_i} \max\left(0, \min\left(\max_{\forall l \in P_1} P_w(l)\right) - \max_{\forall l \in V-P} (P(w)\, l) - \sum_{w_i} \max\left(0, \min\left(\max_{\forall l \in P_2} (P_w(l)\right) - \max_{\forall l \in V-P} (P_w(l)\right.\right.\right.\right.$

14. [2 points] Which topic of the course was the most interesting to you and why? Please provide a technical reason – do not give reasons like we taught it well, or it is very popular, or easy to code. What about the least interesting topic and its reasons?

→ Attention was the most interesting topic for me, just one idea changed how word embeddings are perceived and gave a new direction for contextual embeddings.

→ Dependency Parsing was a bit uninteresting for me since it was based upon a lot of rules and did not add much value to NLP tasks.

(13) case (2) :- Verb set = {V} ⇒ A
Prec. set = {NN, NNS, NNP, RB, IN} ⇒ B

$L(w,\lambda) = L(w) + \lambda \left[\sum_{w_i} \left[\min\left(0, \max\left(0, \max_{\forall l \in B} P_{w_{i-1}}(l) - \max_{\forall l \in P-B} P_{w_i}(l)\right)\right) * \min\left(0, \max\left(0, \max_{\forall l \in A} P_{w_i}(l) - \max_{\forall l \in P-A} P_{w_i}(l)\right)\right)\right]\right]$

⇒ if 'i' is verb set, loss forces preceding to be from B set else

13. [4 points] Constraints in NLP

You are training a sequence labeling model for POS tagging using a Transformer. The input is the sentence $s=w_{1..N}$ and the output is a probability distribution $p_i(l)$ that represents the probability that POS for word w_i is l. Let the gold label for w_i be y_i. Moreover, you wish to add two additional (soft) constraints: (1) a sentence always has an equal number of opening and closing bracket/parentheses symbols (POS tag: OB and CB, respectively), and (2) a verb is always preceded by either a noun (NN, NNS or NNP), or an adverb (RB), or a preposition (IN). Devise a complete loss function for the task. If you introduce any hyperparameters, mention which ones.

L(w) = Binary cross entropy loss function.

$L(w,\lambda) = L(w) + \lambda f_k$.

Case (1) : equal no. of opening & closing parentheses. (count all opening & closing, subtract, if not equal, make the one with the most less diff in prob near to required tag)

Let $P \in \{OB, CB\}$ → set of parentheses
$V \in$ {all tags}.

$P_1 = \{OB\}$
$P_2 = \{CB\}$

Let $h(x) = 1$ if $x > 0$
$= 0$ if $x < 0$.

define $g(w) = h\left[\left(\max_{\forall l \in P_1} P_w(l) - \max_{\forall l \in P-V} P_w(l)\right)\right]$

for each w, g(w) tells (1 or 0) if the word is

$L(w,\lambda) = L(w) + \lambda \max\left(0, \left| \sum_{w_i} \left(\max_{\forall l \in P_1} - \max_{\forall l \in V-P_2}\right)\right.\right.$

$\max\left(0, \left| \sum_{w_i} \max\left(0, \min\left(\max_{\forall l \in P_1}(P_w(l)) - \max_{\forall l \in V-P}(P(w)\, l\right) - \sum_{w_i} \max\left(0, \min\left(\max_{\forall l \in P_2}(P_w(l)) - \max_{\forall l \in V-P}(P_w(l)\right)\right.\right.\right.$

14. [2 points] Which topic of the course was the most interesting to you and why? Please provide a technical reason – do not give reasons like we taught it well, or it is very popular, or easy to code. What about the least interesting topic and its reasons?

→ Attention was the most interesting topic for me, just one idea changed how word embeddings are perceived and gave a new direction for contextual embeddings.

→ Dependency Parsing was a bit uninteresting for me since it was based upon a set of rules and did not add much value to NLP tasks.

14 Most interesting topic **2 / 2**

- **1 pts** (most) reasoning is not technical/question not attempted
- **1 pts** (least) reasoning is not technical/question not attempted
- **0.5 pts** (most) reasoning not technical enough
- **0.5 pts** (least) reasoning not technical enough

✓ **- 0 pts** *OK*