1. Given the following quote from Albert Einstein:

we cannot solve problems with the kind of thinking we employed when we came up with them.

(a) Perform token learning with the byte-pair encoding (only 4 iterations). For each iteration, write down: the selected most frequent pair, the dictionary, and the learned rule set. 8

Total for Question 1: 8

2. Given the corpus $D = \{d_1, d_2, d_3\}$, where:
d_1="never gonna give you up"
d_2="never gonna let you down"
d_3="never gonna run around and desert you"

(a) Write the tfidf formula, and explain the idea behind the formula: $\boxed{4}$

(b) For each document, compute the tfidf for the words "never" and "up": $\boxed{4}$

Total for Question 2: 8

Natural Language Processing - Part II

22 June 2023

3. Answer the following questions by reporting the mathematical procedure, if needed. If you have to compute the actual value, please write the procedure that leads you to the numerical values.

Write part I in a <u>different</u> sheet of paper than part II so we can <u>split</u> them when grading.

(a) When building NLP applications is important to be aware to which extent different NLP machinery can capture short or long-term dependencies. Below you find different NLP tools we encountered in the course. Connect each tool with the number of word token relationships they capture at inference time. That is, 10 means the tool captures relationships with word token 10 word apart. $\boxed{1}$

Elman RNN	LSTM	word2vec	GPT2
100	1	10	2^{10}

(b) Let's assume that you have a vocabulary V made of a million word tokens and you have a word embedding matrix $\mathbf{E} \in \mathbb{R}^{D \times |V|}$ where D is the embedding dimension. Let us also assuming that the word `salt` is at index $i = 100$ of V. Describe with linear algebra how you can get the word embedding of the word `salt` from $\mathbf{E}$. $\boxed{1}$

(c) Consider you have two corpora C_x and C_y of text of the same length: N word tokens. They share the same vocabulary V. A corpus is generated randomly selecting word tokens in V randomly with uniform distribution with replacement. The second corpus is a common dataset used to train `Continuous Bag-of-Words CBOW word2vec`. Assume you have the parameters of `word2vec` trained on this common dataset. You also know the window-size T used to train it. $\boxed{2}$

Describe a procedure to automatically discern the two corpora.

Total for Question 3: 4

4. We are given a **Elmann RNN** trained at the character level. The input sequence of character is shown in Fig. 1. The vocabulary is `['i','o','c','a','!']` and the associated char embeddings are `[-2,-1,2,1,0]`. The details of the RNN are:

- hidden layer $W_h = -1$, hidden layer input $W_x = 2$, hidden layer classification $W_y = -1$. The activation functions are all $\sigma(x) = x$. There are NO biases.
- RNN is trained with self-supervision similar to GPT2. Instead of cross-entropy, it regresses the char embedding of the next char token with a loss as $\big(f(z) - y\big)^2$ where $f(z)$ is the output of the RNN and y is the next char embedding. The final loss is $\ell = \sum_{i=1}^{4} \ell_i$

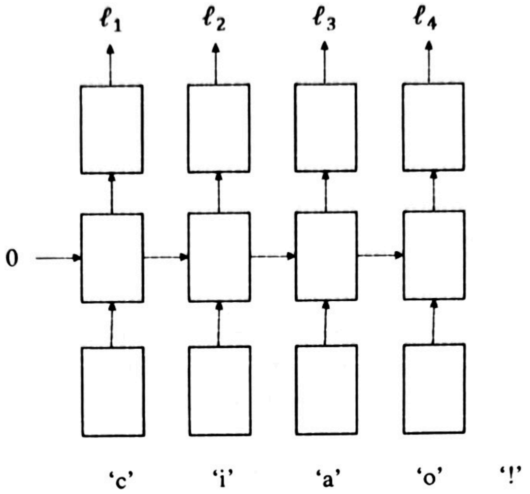

Figure 1: Elman network considered.

(a) $\boxed{4}$

- Write down the generic equation for all the state update for the model shown in Fig. 1 at a generic time step.
- Compute the numerical value of $\frac{\partial \ell}{\partial W_h}$, that is the partial derivative of the loss wrt to the hidden layer W_h, showing all the derivation with a computational graph by breaking it down to forward pass and backward pass. I strongly suggest you rebuild the graph on paper.

(b) Assume you have a vocabulary defined as `['i','o','c','a','!']`. You are at the end of classification layer of the RNN at time step n. The RNN this time models vectors and works with cross-entropy loss and softmax trained with self-supervision. You observe that the gradient of the loss at the branch n respect to the logit $\mathbf{z}$ is $\nabla_{\mathbf{z}} \ell_n(w_1, \dots, w_n; \mathbf{z}) = [0.1, 0.1, 0.2, -0.65, 0.25]$. $\boxed{1}$

- State if you can recover which is the **ground-truth char token** set as supervision to the RNN at this step. Motivate and describe the process to recover it, if possible.
- Moreover, describe how you could retrieve and compute the value of the loss at this time step, if possible.

(c) Assume you have an RNN model similar to the one in Fig. 1 that works over word tokens and is NOT a toy example. You need to train it to predict if an exam is perceived easy or difficult by student—binary classification—given textual comments of students about previous exams. $\boxed{1}$

Consider the case where you have a dataset in which cases similar to the following, or variations thereof, appear often: `''The computer science exam was insanely easy''`.

Explain what would you change in the RNN model.

Total for Question 4: 6

5. Consider a transformers network with the learnable self-attention mechanism with a single head with residual connection without layer normalization. It receives as input word embeddings $\mathbf{X} \in \mathbb{R}^{D \times N}$ where D is the word embedding dimensions and N is the number of tokens.

   (a) Write the equation of the learnable self-attention mechanism with a single head with liner algebra, in function of the given input $\mathbf{X}$, writing down all the parameters that you need to learn and explain what is their role. Pay attention to the fact that matrices size should match. $\boxed{2\frac{1}{2}}$

   (b) Based on the definition that you gave before, state if your self-attention matrix should sum up to 1 across rows or across columns or both, explaining the motivation. $\boxed{1\frac{1}{2}}$

   (c) A transformer is used also in the Contrastive Language-Image Pre-training (CLIP) model. Let's assume that you have an image **x** of a scene and need to use CLIP to make an automatic decision if a particular object `<object>` is in the image. You can assume the `<object>` size is big enough so that its visual features can be captured well by CLIP. $\boxed{2}$

   - State if you can use CLIP to solve this problem.
   - If you answered yes then briefly describe why and how can you get a prediction if the object is in the scene.
   - If you replied no, then say why CLIP cannot do it.

Total for Question 5: 6

You can use this space for writing. The summary of points is at the bottom.

Question:	1	2	3	4	5	Total
Points:	8	8	4	6	6	32
Score:						