RNNs are versatile! In class, we learned that this family of neural networks have many important advantages and can be used in a variety of tasks. They are commonly used in many state-of-the-art architectures for NLP.

(a) For each of the following tasks, state how you would run an RNN to do that task. In particular, specify how the RNN would be used at **test** time (not training time), and specify

1. how many outputs i.e. number of times the softmax $\hat{\mathbf{y}}^{(t)}$ is called from your RNN. If the number of outputs is not fixed, state it as *arbitrary*.
2. what each $\hat{\mathbf{y}}^{(t)}$ is a probability distribution over (e.g. distributed over all species of cats)
3. which inputs are fed at each time step to produce each output

The inputs are specified below.

i. (3 points) Named-Entity Recognition: For each word in a sentence, classify that word as either a person, organization, location, or none.
Inputs: A sentence containing n words.

ii. (3 points) Sentiment Analysis: Classify the sentiment of a sentence ranging from negative to positive (integer values from 0 to 4).
Inputs: A sentence containing n words.

iii. (3 points) Language models: generating text from a chatbot that was trained to speak like you by predicting the next word in the sequence.
Input: A single start word or token that is fed into the first time step of the RNN.

(b) You build a sentiment analysis system that feeds a sentence into a RNN, and then computes the sentiment class between 0 (very negative) and 4 (very positive), based only on the **final** hidden state of the RNN.

i. (2 points) What is one advantage that an RNN would have over a neural window-based model for this task?

ii. (2 points) You observe that your model predicts very positive sentiment for the following passage:

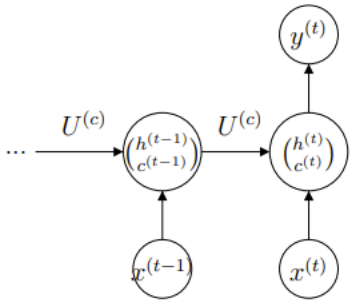
```
Yesterday turned out to be a terrible day.
I overslept my alarm clock, and to make matters worse,
my dog ate my homework. At least my dog seems happy...
```

Why might the model misclassify the appropriate sentiment for this sentence?

iii. (4 points) Your friend suggests using an LSTM instead. Recall the units of an LSTM cell are defined as

$$
\begin{aligned}
i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \\
f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \\
\widetilde{c}_t &= \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \widetilde{c}_t \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
$$

where the final output of the last lstm cell is defined by $\hat{y}_t = \text{softmax}(h_t W + b)$. The final cost function J uses the cross-entropy loss. Consider an LSTM for two time steps, t and $t-1$.



Derive the gradient $\frac{\partial J}{\partial U^{(c)}}$ in terms of the following gradients: $\frac{\partial h_t}{\partial h_{t-1}}$, $\frac{\partial h_{t-1}}{\partial U^{(c)}}$, $\frac{\partial J}{\partial h_t}$, $\frac{\partial c_t}{\partial U^{(c)}}$, $\frac{\partial c_{t-1}}{\partial U^{(c)}}$, $\frac{\partial c_t}{\partial c_{t-1}}$, $\frac{\partial h_t}{\partial c_t}$, and $\frac{\partial h_t}{\partial o_t}$. *Not all of the gradients may be used.* You can leave the answer in the form of chain rule and do not have to calculate any individual gradients in your final result.

iv. (2 points) Which part of the gradient $\frac{\partial J}{\partial U^{(c)}}$ allows LSTMs to mitigate the effect of the vanishing gradient problem? Explain in two sentences or less how this would help classify the correct sentiment for the sentence in part b).

v. (2 points) Rather than using the last hidden state to output the sentiment of a sentence, what could be a better solution to improve the performance of the sentiment analysis task?