1. Report: please list at least one correct and one incorrect prediction from your improved network and give a proposed explanation for why the model might have gotten it wrong. Did the pooling network get these examples right?

   Incorrect: ('tantalizing', 'VERB')

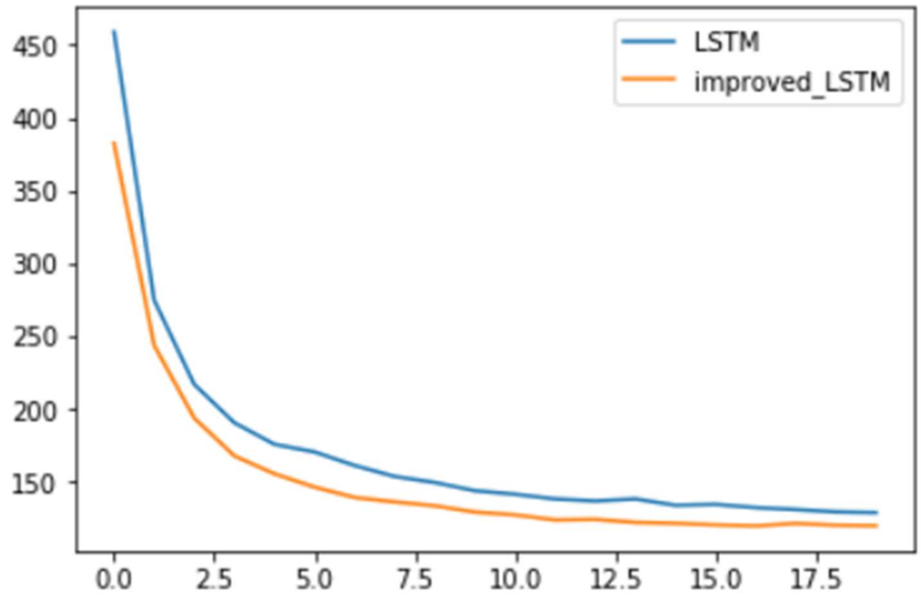
   Correct: ('avoiding', 'VERB')

   The model probably recognized tantalizing as a verb based on its -ing structure. In most cases, word ending in -ing is verb in present participle tense, but in this rare case, it is used as an adj. Thus, in this case, it would require the model to consider longer context to determine these words.

   Pooling network also get them wrong.

2. Report: please describe the modifications you made to your LSTM and its corresponding perplexity. Include (1) a concise and precise description of the extension that you tried, (2) a motivation for why you believed this approach might improve your model, (3) a discussion of whether the extension was effective and/or an analysis of the results, and (4) a bottom-line summary of your results comparing validation perplexities of your improvement to the original LSTM. This should involve some combination of tables, learning curves, etc. and be at least half a page in length.
   Method 1 (all-layer l2):
   (1) Added an L2 regularization term for all weights, with lambda = 0.0001.
   (2) Because our dataset is relatively small, regularization could tune the loss function by adding a penalty term; thus, prevents fluctuation of the coefficients. Thereby, reduces the chances of overfitting.
   (3) The modification is successful as the perplexity reaches 119

(4) Throughout the training process, the validation perplexity of improved_LSTM is lower than the normal LSTM for about 20-50 perplexity.



Method 2 (combined all layer l2 and lr scheduler):

(1) Added an L2 regularization term for all weights, with lambda = 0.0001. Used Step Scheduler which reduces the learning rate by half after 10 epochs.

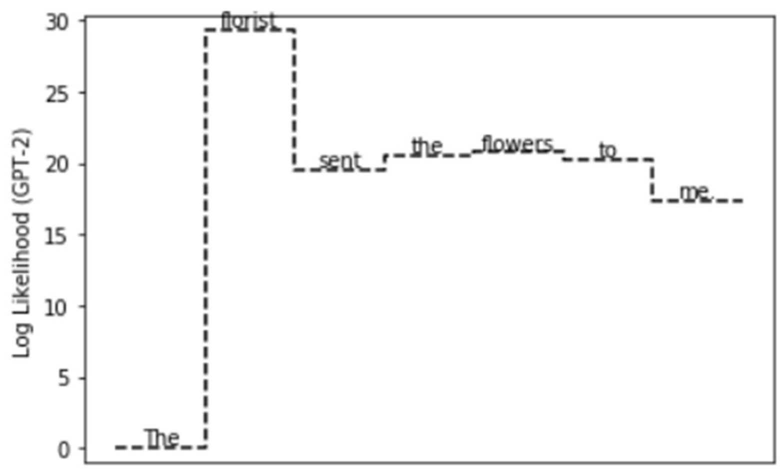
(2) Because our dataset is relatively small, regularization could tune the loss function by adding a penalty term; thus, prevents fluctuation of the coefficients. Thereby, reduces the chances of overfitting. Moreover, after 10 epochs, the validation score does not improve and vibrate. Therefore, I reduce the learning rate by half to make the model step slower to the right target.

(3) The modification is successful as the perplexity reaches 118.

(4) Throughout the training process, the validation perplexity of improved_LSTM is lower than the normal LSTM for about 20-50 perplexity.

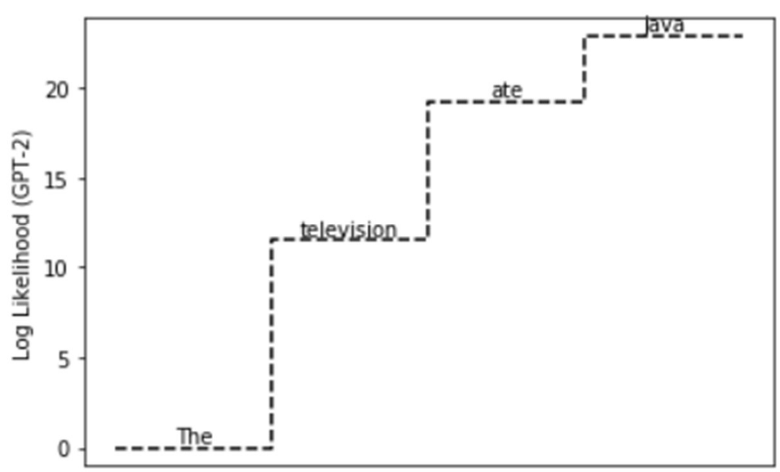
(b) The first sentence shows that for most part of the sentence, the speaker is using quite uniform surprise words. On the other hand, the other sentence “the television ate java” is never used in reality. Thus, its surprise rate show large variation.

</>




```
[0.0, 11.542124406515327, 19.250289286338116, 22.940240456294397]
[ 0.          0.          11.54212441 19.25028929 22.94024046]
```

</>



(c) Report: what prompt did you use for the final part of this assignment? How did you choose it, and did you make any other modifications to the prediction function?

```
"Input: The film is awful tawdry crummy trashy trash shit. Output: no.",
"Input: The movie is terrible cheap sleazy trashy trash shit. Output: no.",
"Input: Beautiful, awesome, and amazing film. Output: yes.",
"Input: The movie is terrific fantastic. Output: yes.",
"Input: I love this incredible marvelous wonderful movie. Output: yes."
```

I start with a simple structure and add synonyms of positive or negative words to it.

I choose the probability distribution of the last word and get “yes” and “no” probabilities from it. The prompts are concatenated by /n so that they are each on a new line.