Assignment 2 FAQs

How is the running time limit calculated?
The time limit for part 2 is 5 minutes to complete execution of all three commands (train, test, and evaluate) in sequence. The time limit is calculated using the Linux `timeout` command. To get an accurate running time, you can prefix your command with `timeout 300` when running your code in the SoC cluster (e.g., `timeout 300 python3 a2part2.py --train …`).

My code runs perfectly on my laptop but produces errors in CodeCrunch
All grading will be done on the SOC cluster, so you should try running your code on the SoC cluster before submitting it to CodeCrunch.

My code runs faster on my laptop than on CodeCrunch or SoC cluster.
When you are running your code on CodeCrunch, GPU will be used as the accelerator instead of CPU. GPU's strength is in parallelization instead of iteration (loop). Thus, it is advisable to minimize loops by processing the data in batch using PyTorch functions (please refer to the documentation). Whenever you see a loop, there is a high chance that it can be replaced with a single-call PyTorch function. You should also make sure that your operations inside the model's forward function are, as much as possible, in PyTorch tensor types. You do not need data types other than PyTorch tensors. Data transfer between CPU and GPU should be minimized, and changing data type from tensors to others and vice versa will require data transfer that makes the code run slower.

My code produces different results on CodeCrunch
With torch.manual_seed(0) (line 14 in the skeletal code), all of your PyTorch operations should be deterministic across different runs on the same device, unless you are using advanced modules that are not needed for this assignment. However, it is not guaranteed to be the same on a different device. What you need to do is to make sure you do not have any randomness in other parts of your code, so that you can get the same result on CodeCrunch on your subsequent submissions.

Randomness can be introduced through the usage of unordered data structures such as set and dictionary, sorting lists that have items with the same precedence, and others. Inspect your code carefully and look for ways to remove randomness from your code completely. If your code does not have any external randomness, your code will have the exact same loss value and accuracy across multiple runs on the same device. Note that your code will only be run once on the blind test set and then will be graded as is.

Is the blind training set similar to the given training set?
There is no blind training set. During grading, the model will be trained using the released training data and tested on both the released test set and the blind test set.

A bigram is represented by a d-dimension vector. How do we decide on the value of d? Is it based on the size of the vocab?
d is a hyper-parameter whose value is to be set by you. d does not have to be equal to |V| (size of the vocabulary).

How to fix the error: “Expected all tensors to be on the same device, but found at least two devices, cuda:0 and cpu!”
You should avoid creating new tensors inside the forward function as much as you can. If you need to do so, set the device of the new tensor to be the same as the other tensors (e.g., the model input) in the forward function. The TA discussed setting/moving the tensor's device in his presentation on PyTorch.

What is the format of the vocab?
You're free to implement vocab in a way most convenient to you.

What is the return of the collator function?
The return of the collator function should be just a tuple of tensors (texts and labels). You can set the labels to None (or other values) when the collator function is used on the test data.

Are we allowed to have a validation dataset? (by splitting the training data)
You are allowed to split the training data and create a held-out validation dataset during your own development. However, when your code is executed in CodeCrunch, your model will be trained on the whole training data set and tested on the released test set.

Do you have some recommendations on where to learn PyTorch?
The following materials are useful for learning PyTorch:

- https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
- https://pytorch.org/tutorials/beginner/basics/intro.html
- Complete documentation: https://pytorch.org/docs/stable/index.html

The following e-book is available from the NUS Digital Library:
Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning
By Delip Rao & Brian McMahan. O'Reilly Media, Inc.