CITI
INNOVATION
LAB
TLV

Shalom,

We are delighted that you are considering joining our NLP team. Our tasks encompass not only proof-of-concept for the solutions we develop, but also launching them into production. Therefore, we all need skills that are required to create production-ready code that can be then integrated into the overall workflow, and successfully deployed by our engineering team. The following is a short exercise that provides you with an opportunity to demonstrate your preparedness to come up with the production-ready solutions.

Please, be aware that the focus of this exercise is the code, and not the NLP solutions. We in our NLP group follow a coding-for-production "manual of style" that we developed jointly with our engineering team. It stipulates that the notebook-based experiments must be eventually converted into production code that uses Python classes and functions, multiple Python modules, is well documented (so that it can be maintained and reused in the future), and is optimized for performance and robustness (and of course, follows PEP 8 conventions). The procedures we follow stipulate that individual R&D branches are merged into master by the engineers who scrutinize the code prior to the merging. – If they deem the code insufficiently production-ready, they reject it, and it must be refactored. The following exercise provides you the opportunity to demonstrate that this will not be required once you join the team.

Data Preparation

Choose a category and download the corresponding data from here: http://deepyeti.ucsd.edu/jianmo/amazon/index.html. Write a Python module that reads the downloaded file, and outputs a shuffled dataset required for BERT classification. Demonstrate that BERT can be fine-tuned using your dataset. Document your code with in-line documentation.
Download data from here: https://www.kaggle.com/alaakhaled/conll003-englishversion. Write a Python module that reads the downloaded file, and outputs a shuffled dataset required for BERT classification. Demonstrate that BERT can be fine-tuned using your dataset. Document your code with in-line documentation.
In the documentation explain how these two datasets are different, and what the differences in the resulting tensors are.
Please, submit your code as one or more text files with the extension ".py" (and NOT as a Jupyter notebook).

Good luck!