# Programming for Computational Linguistics 2018/2019

## Final Project Checkpoint 1 — `corpus.py`

These checkpoint documents provide a detailed recommendation for how to proceed with the project. They will break the requirements into small, well-defined steps. It is not necessary to follow this plan — you can choose to implement the requirements however makes the most sense to you. However, if you are unsure of how to proceed, this should provide you a good starting point.

Start by implementing the functions `tokenize` and `detokenize`. Start simple here – for `tokenize`, maybe use something from NLTK, or use the tokenizer you wrote for lab 7. For `detokenize`, something as simple as putting a space between every two tokens could work. For now, we just want functions that produce some sort of tokens, so that we can get started on the rest of the projext.

In the future, once your language model works, you will probably want to come back and improve these functions for better results. One obvious optimization would be to lower-case all tokens in `tokenize`. This would greatly reduce the size of your language model's vocabulary, and would allow it to better learn from smaller datasets. If you do this, you could even try to re-capitalize the first token of each sentence in `detokenize`.