# E4577 – Assignment 3: Pre-Processing library

In this assignment you will be tasked with writing a Python library to perform preprocessing for a sentiment analysis task with a CNN + Embedding model.

As explained in class, you code should take a string of raw text as input and produce an array of int as an output.

You code will have to perform the following tasks, which should also be the main methods in your code

- *clean_text*
  - should be cleaning the raw text to remove URLs and any other tokens that you deem unnecessary
- *tokenize_text*
  - should be converting a string into an array of tokens.
  - you will be using the *TweetTokenizer* from *nltk* for that operation
  - *nltk* requires artifacts to be downloaded before being operational
  - your code should be including these artifacts in the cleanest way possible to not require download at boot time
- *replace_token_with_index*
  - should be replacing each token in a list of tokens by their corresponding index in an embedding dictionary and producing a list of indexes
  - you will be using the twitter *GloVe* embedding dictionary for that task
  - you should load the dictionary in the most memory efficient way possible until a given number of words (*max_length_dictionary*)
- *pad_sequence*
  - should be padding a list of indices with 0 until a maximum length (*max_length_tweet*), like explained in class

Your end to end pipeline should be taking two optional arguments: *max_length_tweet* and *max_length_dictionary*

Through this assignment, your codebase will have the following requirements:

- it should be unit tested
- the code coverage should be at 80%
- it should have a perfect linting score (10/10)

To report coverage:

- install pytest and pytest-cov
- run pytest –cov ./<path_to_your_source_dir>

To report linting:

- install pylint
- run pylint ./<path_to_your_source_dir>

The linter report will provide you with a list of suggestions on how to improve your code. You can have it integrated with Sublime using the plugins defined in assignment #1.

Finally, your repository should be integrated with TravisCI and Travis should be running your unit tests with the code coverage and pylint