Comp 479/6791 # Project 1 **Fall 2022**

Due date: September 22, 2022

Goal: text preprocessing with NLTK, proofreading results

Data: Reuter's Corpus Reuters-21578
`http://www.daviddlewis.com/resources/testcollections/reuters21578/`

Note that you should always retain the original corpus. Speaking of text 'cleaning' and of stop word 'removal' is a sloppy short form for creating a clean second version of your data without stop words. Also, you may find out that you removed items that later turn out to be valuable. Thus for small collections like Reuter's, make a read only copy of the original.

Overview: Use NLTK for this project. Download the Reuter's-21578 corpus onto your computer. Use that version of the corpus, not the one available in NLTK.

Develop a pipeline of steps to

1. read the Reuter's collection and extract the raw text of each article from the corpus
2. tokenize
3. make all text lowercase
4. apply Porter stemmer
5. given a list of stop words, remove those stop words from text. Note that your code has to accept the stop word list as a parameter, do not hardcode a particular list

A pipeline means that every step can be executed in stand-alone fashion with the appropriate input and will generate output suitable as input for the next module. Manually proofread every step in your pipeline.

Description: Each step in your pipeline has specific additional requirements in order to be considered satisfactory. It is important that you do not limit yourself to the requirements, but think beyond the minimum requirements for your solution. For all modules, create your own test cases for more general solutions.

Deliverables: a folder named "Deliverables" to be submitted in Moodle before September 22, 2022 must include

- (6pts) your pipeline
- (2pts) output files returned from each of the five modules in the pipeline for the first five documents in the collection (= 25 small files)
- (1pt) Report: a .pdf document of no more than 5 pages that explains your work and submitted modules. Make it readable.
- (1pt) Demo file: a .pdf document of no more than 10 pages in which you walk through a demo of your system. Make sure you showcase strengths of your code. Make sure mention shortcomings of your system in the Report and the Demo file. The marker is free to ask some students (or all students) to demo their systems in the labs.