Comp 479/6791 **Project 1** **Fall 2023**

Due date: September 20, 2023

Goal: text preprocessing with NLTK, proofreading results

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

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 Reuter's news item (these are your documents) 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 on different sample input.

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 gzip file named "Deliverables.zip" to be submitted in Moodle no later than September 20, 2023 must include

- (2.5pts) code for your NLTK pipeline in a file called 'Pipeline'
- (1.5pts) output returned from each of the five modules in your pipeline for the first five Reuter's news items in the corpus (it is the assignment to students to find out what that is) in the collection (= 25 **small** output files for 5 news items x 5 pipeline modules). Call the files for the output of the respective module 'Tokenizer-output', 'Lowercased-output', 'Stemmed-output', 'No-stopword-output'. Enumerate the stopword list you used in a separate file called 'Stopwords-used-for-output'
- (.5pt) Report: a file 'Report.pdf' of no more than 5 pages that explains your work and submitted modules. Expected is a report including design decisions and detailing problems and clever ideas. The report is not a log file

- (.5pt) Demo: a file 'Demo.pdf' 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 that you 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