SUPERVISOR/S: Tony C Smith and Panos Patros

PROJECT TITLE: Random-rip Text Classification

FIELD: computer science

DIVISION/SCHOOL: HECS - School of Computing and Mathematical Sciences

PROJECT LOCATION: Hamilton

PROJECT ABSTRACT:
It is known that large portions of documents are not only unhelpful for classification, but detrimental ... in that much of a text can be irrelevant to its class. Current techniques rely on sophisticated feature extraction methods or language models to generate useful representations of documents prior to classification—attempting to find the most useful bits of text. This summer project will borrow from the idea of Random Forests and compression-based classification by leveraging repeated random sampling of subsequences. The idea is to extract a random substring of a target document, and another equal length substring from a rated document; then concatenate the former to the latter and compress the entire sequence with bzip. The same target sequence is then concatenated to one from a sample document rated as being in another class, then compressed with bzip. Notionally, the target sequence will compress better when combined with a sequence from the same class; thus compression size indicates class. To avoid poorly chosen subsequences leading to misclassification, the process is repeated many many times for a single document (i.e. many different randomly chosen subsequences) and the results are averaged; hopefully leading to accurate classification via massive sampling. Bzip is fast and requires little memory, meaning that any success from this approach at all could lead to new and useful text classification algorithm.

STUDENT SKILLS:
- strong pass in COMPX301
- prefer some experience with document classification, or at least some form of ml-based classification
- interest in classification and text compression
- familiarity with Linux (although this can be learned)

PROJECT TASKS:
- Research current state of compression-based text classification, and summarise state-of-the-art classification performance results for MIMIC-III (and similar) collections
- Design pseudocode algorithm for random-rip classification
- Implement algorithm as a shell script for testing
- Test, debug, assess performance and scalability
- Run large scale tests on well-known collections (e.g. MIMIC) for binary classification
- Repeat comparative tests using conventional bag-of-words, word2vec and doc2vec classification methods
- Evaluate results and hopefully draft a paper on the outcomes
- If time permits, further evaluate for multi-label classification

EXPECTED OUTCOMES:
- Student’s Research Poster (as per clause 6 of the Scholarship regulations)
- Success or failure, but with a publishable result either way
- Draft of a paper, and ideally submission thereof