Durham University

# COMP42415: Text Mining and Language Analytics coursework

| | |
|---|---|
| **Module/Lecture Course:** | Text Mining and Language Analytics |
| **Deadline for submission:** | Friday, 18 March 2022, 14:00 GMT |
| **Submission instructions:** | Submit all files via Blackboard Ultra |
| **Submission file type(s) required:** | Zip file named "[CIS USERNAME].zip" containing the Word/PDF document for the individual report, Jupyter notebook(s) and related supplementary files. **Do not include the provided dataset in the submitted files.** |
| **Format:** | Report as a Word or PDF document. Python software implementation as a Jupyter notebook. Supplementary files as needed. |
| **Contribution:** | The coursework contributes 100% to the final mark for the module. |

In accordance with University procedures, submissions that are up to 5 working days late will be subject to a cap of the module pass mark, and later submissions will receive a mark of zero.

## Content and skills covered by the assignment

- Understand advanced concepts of Natural Language Processing (NLP).
- Have a critical appreciation of the main strengths and weaknesses of a range of NLP methods and understand how to use them.
- Have a critical appreciation of how to prepare textual datasets for analysis.
- Understand how to manipulate potentially large datasets in an efficient manner.
- Be able to write computer programs for NLP in Python using industry-standard packages.
- Be able to select appropriate data structures for modelling various NLP scenarios.
- Be able to select the appropriate algorithms for a given NLP problem.
- Be able to prepare, train, evaluate and deploy machine learning models for NLP.
- Effective written communication.
- Planning, organising and time-management.
- Problem solving and analysis.

## Requirements

Students are expected to work on the coursework **individually**.

This assignment requires you to design, explain and justify your proposed solution for a Natural Language Processing (NLP) scenario. The solution should be implemented using the Python (3.x) programming language and also requires a written report to demonstrate how and why you designed the proposed solution, as well as a thorough performance evaluation of it.

Durham University

## Scenario

In this imaginary scenario, you are a data scientist for a marketing company. Your company has asked you to create machine learning models for sentiment analysis of internet posts, like for example tweets, movie reviews, product reviews, etc. The goal of this analysis is to determine whether a post's author expresses positive, neutral or negative sentiment via their posts, thus allowing your employer to monitor the public opinion about potential customers. To succeed in your assignment, you have to use the provided tweets and movie review datasets and create suitable machine learning models for sentiment analysis, as described below. The deliverables for your assignment consist of a **Python implementation** for training the proposed models and using them for predictions, as well as a **written report** that justifies your decisions and provides a performance evaluation of your proposed solutions.

## Datasets

### Rotten Tomatoes Movie Reviews

The dataset is comprised of tab-separated files with phrases from the Rotten Tomatoes movie reviews dataset. Each phrase is annotated with a sentiment value, as follows: 0 – negative, 1 - somewhat negative, 2 – neutral, 3 - somewhat positive, 4 – positive.

| File | Description |
|---|---|
| rtmr_info.txt | Text file that contains information about the dataset |
| rtmr_train.tsv | Tab-separated file that contains the phrases to be used for training |
| rtmr_test.tsv | Tab-separated file that contains the phrases to be used for testing |

### Sentiment140

The dataset is comprised of comma-separated files with posts from Twitter (tweets). Each tweet is annotated with a sentiment value, as follows: 0 - negative, 2 - neutral, 4 - positive.

| File | Description |
|---|---|
| s140_info.txt | Text file that contains information about the dataset |
| s140_train.csv | Comma-separated file that contains the tweets to be used for training |
| s140_test.csv | Comma-separated file that contains the tweets to be used for testing |

## Python implementation (70% of Total Marks)

You are asked to use the provided datasets in order to develop and evaluate machine learning models for sentiment analysis from text. The data from the "*_train.*" files should be used for training the machine learning models, while their performance should be evaluated (tested) on the data from the "*_test.*" files. The required tasks are the following:

1. Load and prepare the available data as needed. **(5%)**
2. Transform the input text using a suitable representation for each model. **(10%)**
3. Implement **three** Naïve Bayes models for sentiment analysis: **(a)** The first will be trained on the Rotten Tomatoes Movie Reviews (RTMR) dataset and tested separately on the RTMR and the Sentiment140 (S140) datasets. **(b)** The second will be trained on S140 and tested separately on RTMR and S140. **(c)** The third will be trained on the combination of RTMR and S140 and tested separately on RTMR and S140. **(12%)**

4. Implement a Convolutional Neural Network (CNN) model for sentiment analysis. Train your model on the RTMR dataset and test it separately on the RTMR and S140 datasets. **(11.5%)**
5. Implement a Recurrent Neural Network (RNN) or a Long Short Term Memory (LSTM) model for sentiment analysis. Train your model on the RTMR dataset and test it separately on the RTMR and S140 datasets. **(11.5%)**
6. Compute the confusion matrix, accuracy, F1-score, precision and recall for each model and training/test configuration. **(10%)**
7. Store the **five** trained models in files and implement a function "predict_post(text, model)" that given a text string ("text") and model filename ("model"), it will load the pre-trained model, and predict the sentiment class of the input text. **(10%)**

## Written Report (30% of Total Marks, 1500 words max)

You are asked to provide a written report about the developed NLP solution. The report should be divided in the following sections: i) Dataset, ii) Data preparation, iii) Text representation, iv) Machine learning models, v) Experimental results, vi) Discussion. The following are required:

1. Critical discussion about the datasets (suitability, problems, class balance, etc.). **(5%)**
2. Description and justification of the data preparation steps used. **(5%)**
3. Description and justification of the text representation method(s) used. **(5%)**
4. Description and commentary on the machine learning architectures used. **(5%)**
5. Detailed performance evaluation of the developed machine learning models. **(5%)**
6. Critical discussion on the achieved results, including potential limitations and usage instructions/suggestions. **(5%)**

## Examiners expectations

What the examiners expect from your software implementation:

- Your program must be runnable – a program that partially works or does not run at all will receive no mark.
- You are asked to use Python 3.x.
- Your source code should be documented with comments, making it to be as easily followed as possible.

What the examiners expect from the report:

- Your report needs to be professional and the language should be scientific.
- Your report should provide justification for the design decisions you made in your solution.

## Word Limit policy

Tables and figures are excluded from the word limit. Examiners will stop reading once the word limit has been reached, and work beyond this point will not be assessed. Checks of word counts may be carried out on submitted work. Checks may take place manually and/or with the aid of the word count provided via electronic submission.

## Plagiarism and collusion

Your assignment will be put through the plagiarism detection service on Blackboard Ultra and the submitted Python code will be checked using a programming plagiarism detection tool.

Students suspected of plagiarism, either of published work or work from unpublished sources, including the work of other students, or of collusion will be dealt with according to the Computer Science Department and University guidelines.