# COM3110 : Text Processing (2020/2021)

## Assignment: Sentiment Analysis of Movie Reviews

## 1 Project Description

The aim of this project is to implement a machine learning model based on Naive Bayes for a sentiment analysis task using the Rotten Tomatoes movie review dataset. Obstacles like sentence negation, sarcasm, terseness, language ambiguity, and many others make this task very challenging.

## 2 Submission

Submit your assignment work electronically via Blackboard. Precise instructions for what files to submit are given later in this document. Please check you have access to the relevant Blackboard unit and contact the module lecturer if not.

**SUBMISSION DEADLINE: 15:00, Friday week 11 (11 December, 2020)**

Penalties: standard departmental penalties apply for late hand-in and use of unfair means

## 3 Data Description

The dataset is a corpus of movie reviews originally collected by Pang and Lee. This dataset contains tab-separated files with phrases from the Rotten Tomatoes dataset. The data are split into **train/dev/test** sets and the sentences are shuffled from their original order.

- Each sentence has a **SentenceId**.
- They all have been **tokenized** already.

The training, dev and test set contain respectively 7529, 1000 and 3310 sentences. The sentences are labelled on a scale of five values:

0. negative
1. somewhat negative
2. neutral
3. somewhat positive
4. positive

In the following table you can find several sentences and their **sentiment score**.

| SentenceId | Phrase | Sentiment |
|---|---|---|
| 1292 | The Sweetest Thing leaves a bitter taste . | 0 |
| 343 | It labours as storytelling | 1 |
| 999 | There 's plenty to enjoy – in no small part thanks to Lau . | 3 |
| 1227 | Compellingly watchable . | 4 |

# 4 Evaluation

Systems are evaluated on classification accuracy (the percentage of labels that are predicted correctly) for every sentence on the dev and test set.

# 5 Project Roadmap

1. Implement some preprocessing steps:
   - You are free to add any preprocessing step (e.g. lowercasing) before training your models. Explain what you did in your report.
   - Implement a function to map the 5-value sentiment scale to a 3-value sentiment scale. Namely, the labels "negative" (value 0) and "somewhat negative" (value 1) are merged into label "negative" (value 0). "Neutral" (value 3) will be mapped to "neutral" (value 1). And finally, "somewhat positive" (value 3) and "positive" (value 4) will be mapped to the label "positive" (value 2).
2. Implement a Naive Bayes classifier **from scratch**.
   - You may **NOT** re-use already implemented classes/functions (e.g. scikit-learn)
3. For each set of labels (5-value and 3-value scales), train **at least two different models** (at least 4 models in total):
   - One considering all the words in the training set as features.
   - One with a set of **features** of your choice determined by your experience (you will explain how you selected the features in your short report).
4. Compute and display confusion matrices on the development set for each developed models. Compare the results.
5. Process the test data with your best performing model.
6. Write a report (see below for details)

# 6 What to Submit

Your assignment work is to be submitted electronically using MOLE, and should include:

1. **Your Python code**.
   It can be either a python file or a python notebook.

2. **Four files with the predictions on the development and test corpora**.
   The format is tab separated as follows : **sentence_id[tab]sentiment_value**
   An example file named "SampleSubmission_test_predictions_5classes_John_DOE.tsv" is provided with the data.
   Those files **MUST BE NAMED** respectively:

   - dev_predictions_3classes_Firstname_LASTNAME.tsv
   - test_predictions_3classes_Firstname_LASTNAME.tsv
   - dev_predictions_5classes_Firstname_LASTNAME.tsv
   - test_predictions_5classes_Firstname_LASTNAME.tsv

   where **Firstname is your firstname and LASTNAME is your lastname**.

3. **A short report (as a pdf file)**.
   It should **NOT EXCEED 2 PAGES IN LENGTH**. The report should include a brief description of the extent of the implementation achieved, and should present the performance results you have collected under different configurations, and any conclusions you draw from your analysis of these results. Graphs/tables may be used in presenting your results, to aid exposition.

# 7 Assessment Criteria

A total of 25 marks are available for the assignment and will be assigned based on the following criteria.

**Implementation and Code Style (15 marks)**
Have appropriate Python constructs been used? Is the code comprehensible and clearly commented? How many different models have been tested? How do you choose which is the best model?

**Report (10 marks)**
Is the report a clear and accurate description of the implementation? How complete and accurate is the discussion of the performance of the different systems under a range of configurations? What is the most important aspect to be taken into account to get the best results with a Naive Bayes approach?

# 8 Notes and Comments

- Consider using the **Pandas** library to load the data `https://pandas.pydata.org/`.
- Consider using **Seaborn heatmap** to render the confusion matrices `https://seaborn.pydata.org/`.
- You may search internet for lists of English punctuation and/or stopwords (also called function words) that you may use in your assignment.