Coursework (mini-project) assignment

1. Description

This mini-project is based on the material covered in lectures and the programming exercises that you are provided with during this course. The application that you will be addressing is *Sentiment Analysis*, which is concerned with the automated identification of the opinion polarity associated with a particular piece of text. Most frequently, this is defined as identifying whether a particular piece of writing (e.g., a review on a movie or a product) is positive or negative, and this is precisely what you will be doing in this project. Specifically, you are provided with a set of reviews extracted from the Internet Movie Database (IMDb) with various polarity labels, and your task is to develop an application that can detect the sentiment polarity given any input text. This is a real-world application that is popular in an academic as well as industrial context.

2. Dataset

You will be using a subset of 2,000 positive and 2,000 negative movie reviews extracted from the Large Movie Review Dataset (https://ai.stanford.edu/~amaas/data/sentiment/). The reviews were extracted based on their star rating: reviews with a score <=4 stars are considered clearly negative, while those with a score >=7 are considered clearly positive. You are provided with a set of positive reviews (in the pos/ folder) as well as negative reviews (in the neg/ folder), where you can see that the file naming conventions are as follows: id_star.txt. In other words, if you'd like to benefit from the star rating assigned to a particular review, you can extract this information from the names of the files.

For background information on the task and the datasets, take a look at:

- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*
- The dataset webpage: https://ai.stanford.edu/~amaas/data/sentiment/

3. Your assignment

Your assignment is to build a sentiment analysis application using the data provided. Specifically, you will need to apply the steps from the pipeline discussed in the course:

- *Analyse the data and the task*: Familiarise yourself with the data and define the framework for the task and approach to evaluation. Summarise your ideas in the Jupyter notebook.
- *Apply relevant data pre-processing steps*: Apply appropriate pre-processing steps that you have learned about in the course.
- *Extract relevant information*: Identify informative features for the task. Relevant techniques from the course that can be applied at this step include tokenization, PoS tagging, chunking or parsing (e.g., for the identification of phrases).
- *Apply a relevant algorithm*: Apply a sentiment analysis algorithm using the knowledge and skills acquired during the course. Note that if you experiment with different features, settings or algorithms in a machine learning framework, you need to split the dataset into training and test sets (or apply cross-validation) and run the experiments on the same splits to make different runs of your algorithm comparable.

- *Report evaluation results*: Apply relevant evaluation metrics and report the results at different steps of your implementation. Note that you will not be assessed on the basis of your evaluation results: i.e., if you implement a reasonable algorithm that attempts to solve the task but achieves low evaluation scores, you will not be penalised.

To get full marks on this project, you will need to perform a comparative experiment: i.e., implement a baseline model (e.g., the simplest or most straightforward algorithm) and experiment with at least one extension, comparing the results of your extended models to the baseline model using the same data splits. Pass mark on this project will be awarded if you write a good report on a baseline model.

4. Submission guidelines

You should submit a Jupyter notebook providing a solution to the task together with the accompanying description of the steps you applied. The description of your work in the notebook should be within 2,500 words excluding tables, graphs and images. The coursework mark will contribute 30% of the final unit mark. The deadline for submitting completed reports is Monday, 12th December 2022.

The assessment will be based on the clarity of the description and motivation of the work done, steps implemented and evaluated, demonstration of the skills and knowledge acquired during the course, and insights gained. Assessors may run your code, but you will not be assessed on the quality of your code writing, nor will you be assessed on the basis of where your system's results rank amongst the results achieved by the systems submitted by other course participants or results reported in published papers.