Applied Natural Language Processing: Assessed Coursework

Submission format: You should submit one file that should either be a Python notebook or a zip file containing a Python notebook and any other files (e.g., images or Python files) that you want to include in the notebook.

Due date: Your work should be submitted on the module's Canvas site before 4pm on Thursday 11th November. This is Thursday of week 7. The standard late penalties apply.

Return date: Marks and feedback will be recorded on Canvas by the end of week 10 (Friday 3rd December).

Weighting This assessment contributes 30% of the mark for the module.

Overview

For this assignment you are asked to complete a python notebook ('NLassignment2021.ipynb') which is provided with these guidelines. It is based on activities that you have already completed in labs during weeks 1-4 of the module. Any code you have developed during the labs can be submitted as part of your answers to the questions in the assignment. To score highly on this assignment you will need to demonstrate that you:

- understand the theory and your code;
- can write and document high quality python code;
- can develop code further to solve related problems;
- can carry out experiments and display results in a coherent way;
- can analyse and interpret results; and
- can draw conclusions and understand limitations of the technology.

For this report you should submit a single Python notebook containing all of your answers to all of the questions in 'NLassignment2021.ipynb'. You may import from standard libraries. If you wish to import any other code, it must be included in a zip file with your notebook. It **must** be possible for the assessors to run your Python notebook.

Marking Criteria and Requirements

Your submission will be marked out of 40. The assignment is broken down into 5 questions – all with multiple parts. All parts should be answered. The breakdown of marks between questions is given here and also specified in the notebook. General and part specific criteria are given below. Please read these guidelines carefully and ask if you have any questions.

General: 4 marks available (10%)

4 marks (10%) are available for the overall quality of your assignment. When awarding these marks the following general guidelines will be considered.

- In order to avoid misconduct, you should not talk about these coursework questions with your peers. If you are not sure what a question is asking you to do or have any other questions, please ask me or one of the Teaching Assistants.
- Your report should be no more than 2000 words in length excluding code and the content of graphs, tables and any references.
- You should specify the length of your report. 2000 is a strict limit.
- You should use a formal writing style.
- All graphs should have a title and have each axis clearly labelled.
- In all parts, marks will be awarded for the quality of your written answers as well as your code.
- Written / textual answers **MUST** be included in Markdown cells. Otherwise, you may score 0 for these answers.
- Code on its own does not count as an explanation or a discussion. Nor do code comments. Code should be commented but explanation and discussion MUST be given as text in Markdown cells (see previous point!).
- Do not add external text (e.g. code, output) as images.
- Your code must be applied to and your explanations must refer to the unique set of examples generated by entering your candidate number at the top of the notebook. This must be your own candidate number. Otherwise you may score 0.
- You should submit your notebook with the code having been run (i.e., with the output displayed rather than cleared)
- It **must** be possible for the assessors to run your Python notebook.

Question 1: 8 marks available (20%)

1) a) **Generate** a list of 10 content words which are representative of the positive reviews in your training data.

b) **Generate** a list of 10 content words which are representative of the negative reviews in your training data.

c) **Explain** what you have done and why

The following breakdown of marks will be applied

- Clear and effective use of code [6 marks]
- Explanation [2 marks]

Question 2: 5 marks available (12.5%)

2) a) **Use** the lists generated in Q1 to build a **word list classifier** which will classify reviews as being positive or negative.

b) **Explain** what you have done.

The following breakdown of marks will be applied

- Clear and effective use of code [3 marks]
- Explanation [2 marks]

Question 3: 8 marks available (20%)

3) 1) **Calculate** the accuracy, precision, recall and F1 score of your classifier.

b) Is it reasonable to evaluate the classifier in terms of its accuracy? **Explain** your answer and give a counter-example (a scenario where it would / would not be reasonable to evaluate the classifier in terms of its accuracy).

The following breakdown of marks will be applied

- Clear and effective use of code for calculations [6 marks]
- Argument / explanation in part b [2 marks]

Question 4: 5 marks available (12.5%)

4) a) **Construct** a Naive Bayes classifier (e.g., from NLTK).

b) **Compare** the performance of your word list classifier with the Naive Bayes classifier. **Discuss** your results.

The following breakdown of marks will be applied

- Clear and effective use of code [3 marks]
- Discussion [2 marks]

Question 5: 10 marks available (25%)

5) a) Design and **carry out an experiment** into the impact of the **length of the wordlists** on the wordlist classifier. Make sure you **describe** design decisions in your experiment, include a **graph** of your results and **discuss** your conclusions.

b) Would you **recommend** a wordlist classifier or a Naive Bayes classifier for future work in this area? **Justify** your answer.

The following breakdown of marks will be applied

- Clear and effective use of code to carry out experiment [3 marks]
- Explanation of experimental design [2 marks]
- Graph of results [2 marks]
- Conclusions, recommendation and justification [3 marks]