NLP Coursework Specs (2023)

Summary: Your task will be to develop a binary classification model to predict whether a text contains patronising or condescending language, writing a report based on your findings. The task was task 4 (subtask 1) in the SemEval 2022 competition, with more information available from the task paper.

Your task: Implement a transformer-based model that outperforms the task's RoBERTa-base baseline (provided by the task organisers) in F1 score. The RoBERTa-baseline achieved 0.48 on the official dev dataset, and 0.49 on the test-set.

The coursework will be marked based on your report rather than your final model performance. While you are expected to improve on the performance of the RoBERTa-base baseline, your report will be assessed on how well you answer the questions in the *Marking Scheme* section below rather than on your raw model performance.

Deadline: The coursework deadline is the **7th March (7pm)**

Coursework submission: We require you to submit a PDF of your report, in addition to a SHA1 key for your GitLab repository, corresponding to the commit you want to be submitted. In this repository, you should have completed and pushed:
– Dev set predictions as `dev.txt`
– Test set predictions as `test.txt`

The dev set evaluation will be a public test on LabTS, so please check that this is working as expected. The test set evaluation will be a private test, and you will see your results when your report is marked. Examples of these files are available in your group repos.

Your GitLab repo should also contain the code corresponding to your dev and test predictions, but this code will not be marked.

Data and evaluation: To obtain the task data you will need to submit this form (all group members should complete the form). After completing the form, the task organisers will email you the data - you will be using the **dontpatronizeme_pcl.tsv** file. An allocation of this data into train and the official dev set is also provided here. You will need to create your own internal dev set from within the training data for your hyper-parameter testing. As the test data is held out, creating an additional internal dev set allows you to use the official dev set as your own test set. We ask you to submit the predictions of your best performing model on LabTS, for both the official dev set and the official test set. The test set can be found here. We have been given labels for the test set, so we will share your final test performance when marking your report.

Note, here the task repo also contains a breakdown of the type of PCL language detected for each example (broken down into seven categories). You are welcome to use this additional label information if it is helpful.

Results should be measured using the F1 of the positive class (PCL examples).

Teams*:* You should form groups of up to 3 students. Submissions with fewer students are possible, but will *not* be assessed differently.

If you have issues with a team member not contributing you should inform us well in advance of the deadline (please email marek.rei@imperial.ac.uk).

Code: You are required to upload your code to your GitLab (if you are using a Colab notebook, please make sure this is uploaded to your repo). The code does not need to contain all your results and model configurations. Instead, this code should align with your final test submission. You will not be assessed on your code.

Report: Your report should be a maximum of 5 pages using this template. You may also include any additional appendices, however the markers will not be required to look at these.

Your report should contain the names and email addresses for each person in the group, but does not need to contain your affiliation (Imperial). You are not allowed to change the template, for example you cannot change the font size or the size of the margins.

Marking Scheme

1) Data analysis of the training data (15 marks):

For a written description of the training data. This should include:

1. **5 marks** – Analysis of the class labels: how frequent these are and how they correlate with any feature of the data, e.g. input length.
2. **10 marks** - Qualitative assessment of the dataset, considering either how hard or how subjective the task is, providing examples in your report.

2) Modelling (40 marks):

For the successful implementation of a transformer model:

1. **10 marks**: Successful implementation of a transformer model (train and produce predictions which outperform the F1 score for the RoBERTa-base baseline provided).
    - 7 marks for outperforming the baseline model on the dev set - 0.48
    - 3 marks for outperforming the baseline model on the test set - 0.49
2. **5 marks**: Choice of model hyper-parameters and description of your model setup. This should include choosing an appropriate learning rate and checking whether implementing a learning schedule improves performance. Also consider whether your model is cased or uncased. You should mention how many epochs you train the model for, whether you are using any early-stopping, and how you are using the training labels.
3. **10 marks**: Further model improvements (beyond using a bigger transformer model), for example pre-processing, data sampling, data augmentation, ensembling, etc.
    - Two main improvements, with a third less explored improvement is sufficient.
    - For example: try several different data sampling approaches, try several data augmentation strategies by perturbing observations in different ways, and then see if incorporating one of the categorical columns improves performance.
4. **10 marks**: Compare your model performance to two simple baselines (e.g. a BoW model). Share some of the features that one of your baseline models used, and highlight an example misclassified with a suggestion of why the baseline may have made the misclassification.
5. **5 marks**: Description of the model results and your hyper-parameter tuning (some evidence of this is required in your report). Your results should show how the different strategies you have tried impacted the model performance. For any results presented in your paper, you should be clear if these are from your own internal dev set or the official dev set.

3) Analysis (15 marks):

Analysis questions to be answered (these questions can be answered without training any additional models):

Your report should state the analysis questions so that this can be read as a self-contained report, rather than referring to 'analysis question 1' etc.

1. ***5 marks**:* To what extent is the model better at predicting examples with a higher level of patronising content? Justify your answer.
2. ***5 marks**:* How does the length of the input sequence impact the model performance? If there is any difference, speculate why.
3. ***5 marks**:* To what extent does model performance depend on the data categories? E.g. Observations for homeless vs poor-families, etc.

4) Written report (30 marks):

Marks are awarded for the quality of your written report:

1. **5 marks**: Introduction, with an explanation of the task and the dataset. You may want to read/cite the task paper (and any other paper of your choosing).
2. **10 marks**: Readability of the report (language, coherence, clarity of results etc.).
3. **10 marks**: Good use of graphs or results tables that address the analysis questions. Make sure any text on your graphs is clearly readable.
4. **5 marks**: Conclusion, with a summary of your results, and your key findings from the analysis questions. You should suggest at least one further experiment as a next step.