University of Glasgow | School of Computing Science

Assessed Coursework

Course Name	**Text-as-Data (H/M)**			
Coursework Number	**1**			
Deadline	**Time:**	**4:30pm**	**Date:**	**13th March 2023**
% Contribution to final course mark	**20**			
Solo or Group ✔	**Solo**	**✔**	**Group**	
Anticipated Hours	**30 hours**			
Submission Instructions	**As per specification below.**			
Please Note: This Coursework cannot be Re-Assessed				

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below.

The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline
   a. the work will be assessed in the usual way;
   b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

Penalty for non-adherence to Submission Instructions is 2 bands

You must complete an "Own Work" form via https://studentltc.dcs.gla.ac.uk/ for all coursework

Text-as-Data Coursework

Introduction

The TaD coursework aims to assess your abilities relating to techniques discussed in the course. The objective is to assess your ability in text processing techniques, and applications to text classification.

Your work will be submitted through Moodle and will be assessed primarily on your **PDF report**. Your code is submitted as one or more supporting **Jupyter/Colab notebooks** (as separate .ipynb file(s)). This is an **individual exercise**, and you should work independently. If you have questions concerning this document, you are encouraged to contact the course lecturers as soon as possible.

The page limit of the report is 8 pages including figures. You are encouraged to write concisely. The marking scheme does roughly translate from 1 mark to 1 discussion point / idea which may take a few sentences to communicate.

Dataset cleanliness is an important practice in machine learning. Using your test dataset too much when building a system can damage its ability to give you accurate metrics on the performance of your system. This coursework follows good practice to use your validation dataset for the various experiments when building a classifier (in Q3-5) and only using the test set for the final evaluation in Q6.

(Masters Only) Remember that there is a second coursework which has a separate specification and submission page on the course Moodle.

Q1 - Dataset [8 marks]

Your first step is to choose a **text classification** dataset for this coursework. You will explore it, build and evaluate a classifier on it. The dataset should contain a number of documents (which could be short like a tweet or long like a research article). It should also have some labels (more than two) that you want to be able to predict automatically. Those labels may be categories, topic information, sentiments, authors, or other. Find something that appeals to you.

Below are some examples, but we encourage you to explore and propose alternative datasets too. You must choose your dataset and submit its information through Moodle for approval (the Choose your dataset Questionnaire) **no later than 10 February**. There are questions which require processing your data. Don't leave this to the last minute.

- Datasets from Reddit:
    - You may use the Reddit dataset that we use in the labs (or create your own with the Reddit API)
- Amazon Food Reviews
    - Could try to predict positive, neutral or negative reviews
    - Link: https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews
- Movie Dialog
    - Could try to predict the genre of the movie from snippets of dialog
    - Link: http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html
- Or, a dataset of your choosing! Here are some possible sources:
    - https://github.com/niderhoff/nlp-datasets
    - https://github.com/awesomedata/awesome-public-datasets#naturallanguage

There are some requirements for the chosen dataset:

- It must be publicly accessible
- It must contain between 1000 and 10,000 samples (>= 1000 and <= 10000)

    - If your chosen dataset is too large, you should use random sampling to reduce the number of samples to 10,000
- It must have between three and ten labels (>= 3 and <= 10). This means that it cannot be a binary classification dataset, and must be a multiclass **(not multilabel)** classification dataset.
    - The labels can be created from another field (e.g. by thresholding a numerical score into high, medium and low score labels).
    - For this coursework, if there are more than ten labels, you can either:
        - Keep the nine most frequent labels and combine the less popular labels into an “Other” label
        - Keep the ten most frequent labels and remove any samples that are labelled with any other label

Answer the following questions in your report about the dataset you selected. Note that you may need to do some pre-processing of the data to answer these questions.

(a) What is your dataset, and why did you select it? How might automatic classification/labelling of your dataset be used in practice? **[2 marks]**

(b) Provide a summary of the labels and input text to be used. What labels are to be predicted? Did you need to do any preprocessing to create the labels or to make sure the number of labels was >= 3 and <= 10? What is the text that will be used for classification? **[4 marks]**

(c) Is the dataset already split into a training, validation and test set? If not, use a 60/20/20% split to create the training, validation and test set. Provide a table with the label counts for each split of the dataset and comment on the distribution across labels and across data splits. Be sure you use the same splits throughout the entire report. **[2 marks]**

Q2 - Clustering [14 marks]

Your next step is to perform **k-means clustering** over your data.

Write your **own** implementation of k-means clustering over sparse TF-IDF vectors of your dataset. **[6 marks]**

- Recall that k-means consists of the following steps. Be sure that each of these steps are clearly indicated in your notebook submission.
    - Step 0: Vectorise text
    - Step 1: Pick k random "centroids"
    - Step 2: Assign each vector to its closest centroid
    - Step 3: Recalculate the centroids based on the closest vectors
    - [Repeat Steps 2 and 3 until the model converges]
- If your dataset has multiple text fields (e.g., title and body), choose the field (or combination of fields) that you think will yield the best clusters.
- Consider the model converged when no vectors change their assigned cluster between iterations.
- **Important note: You may NOT use the implementation of k-means provided by Scikit-Learn or any other package.** You may, however, use packages to provide tokenisation, similarity calculation, etc., including the functions that you built in lab.

In your report, answer the following questions:

(a) When using k=5 clusters, give a few examples of the documents assigned to each cluster, and the top 5 tokens with the highest magnitude in the corresponding centroid. **[2 marks]**

(b) Based on (a), do the clusters make sense? Are there certain topics that appear in some but not others? **[2 marks]**

(c) Construct a confusion matrix between the k=5 clusters and your target labels. **[2 marks]**

(d) What trends, if any, do you notice from the confusion matrix? Does it look like some clusters are able to pick up on a single label? When a cluster includes multiple labels, are they related? **[2 marks]**

Q3 - Comparing Classifiers [10 marks]

Use the text in your dataset to train baseline classification models with the Scikit Learn package. Conduct experiments using the following combinations of classifier models and feature representations:

- Dummy Classifier with strategy="most_frequent"
- Dummy Classifier with strategy="stratified"
- LogisticRegression with One-hot vectorization
- LogisticRegression with TF-IDF vectorization **(default settings)**
- SVC Classifier with One-hot vectorization (SVM with RBF kernel, default settings)

(a) Implement the five baseline classifiers above, train them on the training set and evaluate on the validation set. Discuss the classifier performance in comparison to the others and preprocessing techniques **[7 marks]**

> For the above classifiers report the classifier accuracy as well as macro-averaged precision, recall, and F1 (to three decimal places). Show the evaluation metrics[1] obtained by the classifiers on the training and validation sets in one table, and highlight the best performance. For the best-performing classifier (by macro F1 in validation set), include a bar chart graph with the F1 score for each class - (classes on x-axis, F1 score on Y axis).
>
> Analyse and discuss the effectiveness of the classifiers. Your discussion should include how the models perform relative to the baselines and each other. It should discuss the classifiers' behaviours with respect to:
>
> 1) Appropriate model "fit" (how well is the model fit to the training/test dataset),
>
> 2) Dataset considerations (e.g. how are labels distributed, any other dataset issues?)
>
> 3) Classifier models (and their key parameters).

(b) Choose your own classifier/tokenisation/normalisation approach from Scikit Learn, and report on its performance with respect to the five baselines from above on the test set. **[3 marks]**

> You should describe your selected classifier and vectorisation approach including a justification for its appropriateness.

Q4 - Parameter Tuning [5 marks]

In this task you will improve the effectiveness of the **LogisticRegression with TF-IDF vectorisation** from Q3.

Parameter tuning - Tune the parameters for both the vectorizer and classifier on the *validation* set. **[5 marks]**

1. Classifier - **Regularisation** C value (typical values might be powers of 10 (from 10^-3 to 10^5)
2. Vectorizer - Parameters: **`sublinear_tf`** and **`max_features`** (vocabulary size) (in a range None to 50k)
3. Select another parameter of your choice from the classifier or vectorizer

> Your search does **not** need to be exhaustive. Changing all parameters at once is expensive and slow (a full sweep is exponential in the number of parameters). Consider selecting the best parameters sequentially. The

[1] Accuracy and macro precision / recall / F1

resulting tuned model should improve over the baseline TF-IDF model. Report the results in a table with the accuracy, macro-averaged precision, recall, and F1 on the **validation data**. Discuss the parameters and values you tried, what helped and what did not and ***explain why*** **this may be the case.**

Q5 - Context vectors using BERT [12 marks]

Now you will explore whether a deep learning-based approach can improve the performance compared to the earlier more traditional approaches. **Note:** This will be computationally expensive and time-consuming. You should use a GPU where possible for this, which can be enabled in Google Colab under "Change Runtime Type" and through providing an appropriate 'device' to HuggingFace.

(a) Encode the text of your documents using the 'feature-extraction' pipeline from the HuggingFace library with the 'roberta_base' model. Use only the first context vector for each document (which should represent the [CLS] token). Pass the context vectors (without any other previous features) into a LogisticRegression classifier from scikit-learn and train using the training set. Report the evaluation metrics on the validation set. **[3 marks]**

(b) Train an end-to-end classifier using the 'trainer' function from the HuggingFace library, again using the 'roberta_base' model. Use a learning rate = 1e-4, epochs = 1, batch_size = 16 and no weight decay. Report the evaluation metrics on the validation set. **[3 marks]**

(c) Try different values for the model, learning_rate, epochs and batch_size. Normally, you would do some form of systematic search across these values, but due to computational costs, you should not do that. Pick three different sets of these hyperparameters and describe your motivation for these choices. Retrain the models from scratch on the training set and report the evaluation metrics on the validation set for those three settings in a table along with the hyperparameter settings from (b). **[3 marks]**

(d) Which performed best: the approach in part (a) using context vectors from the pipeline approach or using an end-to-end model in parts (b-c). How do these approaches differ? What is the likely reason for any performance difference? **[3 marks]**

Q6 - Conclusions and Future Work [14 marks]

You will now take your best model from Q3/Q4/Q5 and evaluate it on the test set. You will then explore the performance of your classifier and discuss the strengths and weaknesses of your machine learning pipeline.

(a) Take the best approach from the prior questions (which should be trained on the training set) and evaluate it with the test set. Report the evaluation metrics. Provide a single confusion matrix of the classifications on the test set. **[2 marks]**

(b) Manually examine the predictions on the test set. Analyse the results for patterns and trends using the confusion matrix. Hypothesise why common classification errors are made. Report on your error analysis process and summarise your findings. Be sure to give specific examples of the types of errors you observe. **[5 marks]**

(c) Discuss whether the final performance is sufficiently high for the stated purpose of this classifier. What impact would false positives and false negatives have on the deployment of this machine learning pipeline? Refer to the confusion matrix from (a) to support your claims. **[3 marks]**

(d) Could the deployment of this system have any negative societal effects? **[2 marks]**

(e) Propose further steps that could be taken to improve the classification effectiveness of the system. **[2 marks]**

(f) How long did you spend on this coursework (in hours, an approximation is okay). **[0 marks]**

Report Quality [2 marks]

Quality of the report (organisation, correct spelling, presentation, use of appropriate diagrams, evaluation metrics, confusion matrices, etc): **[2 marks]**