University of Glasgow | School of Computing Science

Assessed Coursework

Course Name	**Text-as-Data**			
Coursework Number	**1**			
Deadline	**Time:**	**4:30pm**	**Date:**	**12th March 2024**
% 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 to perform text processing techniques as applied to a multiple-choice question-answering system.

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.

You are tasked with building and evaluating a question-answering system that chooses the best answer among a selection of 4 options. For example:

Example: how much is 1 tablespoon of water [correct answer: (c)]

(a) In the UK, Europe and most Commonwealth countries, a tablespoon is a type of large spoon usually used for serving.
(b) In the US and parts of Canada, a tablespoon is the largest type of spoon used for eating from a bowl.
(c) This tablespoon has a capacity of about 15 mL.
(d) Measuring Spoons

For this exercise, you can assume that the options are all factually correct; the task is to identify which statement provides an answer to the question, not necessarily which provides factually correct information.

The dataset can be downloaded with the link: http://tinyurl.com/tad2024courseworkdata

Q1 - Dataset and Pre-Processing [8 marks]

Start by downloading the train, validation, and test splits of a version of the WikiQA corpus with the link above. Load the data into your Notebook and answer the following questions about it. Note that the data's accompanying README.txt file provides descriptions of the data format, the fields, and other information about its construction, which will be helpful in properly understanding the data.

Use the `text_pipeline_spacy_special` function from Lab 3 to tokenise all questions and their options. Then answer the following questions:

(1.1) How many questions and options are there in each split? **[1 mark]**

(1.2) What is the average number of tokens per question in the training set? **[1 mark]**

(1.3) What is the average number of tokens per choice in the training set? **[1 mark]**

(1.4) What is the average number of tokens per correct choice in the training set? **[1 mark]**

(1.5) Perform any additional exploration of the data that you feel would be helpful for this multiple-choice question-answering task. Briefly describe what you found. **[4 marks]**

Q2 - Set Similarity Measures [10 marks]

Use set similarity measures to calculate the similarity scores for each question against its four corresponding answers. You should use the tokenizer from Q1. For each question, pick the answer with the highest similarity score.

(2.1) Report the performance of each similarity measure (overlap coefficient, Sorensen-Dice & Jaccard) on the training and validation sets by measuring accuracy. **[6 marks]**

(2.2) For each similarity measure, how many times was the score of the most similar answer tied with another answer? When there was a tied score among the top answers, how did you choose which to select? Why? **[4 marks]**

Q3 - Cosine similarity of TF vectors [12 marks]

Generate term frequency (TF) vectors of each question as well as the four possible answers. You should use the CountVectorizer with default settings (but use the same tokenizer as in Q1 and Q2). For each question, pick the answer with the highest cosine similarity between its TF vector and the question's TF vector.

(3.1) Report the performance of the training and validation sets by measuring accuracy. Discuss how they compare with the set similarity measures from Q2. **[6 marks]**

(3.2) Propose, motivate, and evaluate **one** modification to this process to improve this method. Report the performance on the training and development sets and compare them with the unmodified version. **[6 marks]**
(Tip: You may want to inspect questions that the approach gets incorrect to motivate a modification.)

Q4 - Cosine similarity of vectors from bert-base-uncased [12 marks]

Use the feature-extraction pipeline with a `bert-based-uncased` model to create context vectors from the bert-based-uncased model for the text of each question and its four answers separately. You should use the context vector that represents the [CLS] token, which will be the first vector. For each question, pick the answer with the highest cosine similarity between its vector and the question's vector.

(4.1) Report the performance of the training and validation sets by measuring accuracy. **[8 marks]**

(4.2) What are the limitations of the set similarity and cosine similarity methods used in Q2, Q3 and Q4? **[4 marks]**

Q5 - Fine-tuning a transformer model [18 marks]

Train an AutoModelForSequenceClassification with a `bert-based-uncased` model on this dataset. This will involve transformation of the data as described below. You should train only on the training questions and use the validation set for evaluation.

Transform the dataset into a table of rows with each containing a question, an option and a label (1 or 0) if it is the correct answer. The table (referred to as the question-option pairs representation) should have four times the number of rows as questions in the original question dataset. Concatenate each question and option together with "[SEP]" text in between them. For example, the question "where is osaka japan" and the incorrect option "Osaka castle" would become "where is osaka japan[SEP]Osaka castle" with a label of 0.

In an ideal world, you would do hyperparameter tuning to identify the optimal settings. Due to computational cost, use these settings which should provide reasonable performance:

- learning_rate = 1e-5
- batch_size = 8
- epochs = 4
- weight_decay = 0

(5.1) Report the accuracy, precision, recall and F1 score of the predictions on the question-option pairs representation of the training and validation sets **[10 marks]**

(5.2) Report the accuracy for this method for selecting the correct answer on the training and validation sets of this model. Note this is different from the value in part (a). To enable this, select the option for each question with the highest output logit value for the positive class of the model. **[6 marks]**

(5.3) Why would you expect this approach to outperform the use of [CLS] vectors described in Q4? **[2 marks]**

Q6 - Test set performance [4 marks]

(6.1) Report the accuracy using your best method on the test set. Use the performance on the validation set to select the best method. **[2 marks]**

(6.2) Discuss whether the achieved accuracy would be sufficient for deployment **[2 marks]**

Ungraded Extras [0 marks]

This section is entirely optional and will not be marked.

The next approach that could be examined is an AutoModelForMultipleChoice which is a particular Hugging Face architecture of multiple choice questions. It involves careful dataset preparation into a particular format. Examination of the corresponding tutorial may help:

https://huggingface.co/docs/transformers/tasks/multiple_choice

Reasonable hyperparameter settings for this are below:

- learning_rate = 5e-5
- batch_size = 16
- epochs = 3
- weight_decay = 0.1