COMP34812 Natural Language Understanding Coursework

Changelog

Please check the following list of updates to this document to ensure that you are aware of revisions since the release of its first version on the 1st March 2024.

***v1.1 (released 11th March)** - Removed references to EvalAI being the submission platform for predictions on the test data, due to the marking team's decision to use Blackboard for ALL deliverables. The affected document sections are:*

- *Section IV, Item 3 - removed mention of an "online platform that we set up specifically for the NLU shared task"*
- *Section V, Deliverable 1 - replaced EvalAI with Blackboard as the means for submitting predictions. It is important to note that now all four deliverables should be compressed into one zip file, which should then be uploaded to the relevant Blackboard submission site.*
- *Section VI, Launch of evaluation platform - reworded to indicate that EvalAI is just a supporting tool during development, rather than an official submission site. A link to a Blackboard Wiki page with instructions for using EvalAI has also been provided.*
- *Section VI, Deadline: Submission of all deliverables - reworded to indicate that all deliverables should be submitted via Blackboard*

I. Introduction

For the COMP34812 coursework, your group (with two members) will participate in what we call a *shared task.* In the natural language processing (NLP) community, shared tasks are often organised to assess state-of-the-art performance on a particular task. There is a shared task for almost any NLP problem, but some of the more recently organised ones include the SemEval 2023 task on Multilingual Complex Named Entity Recognition[1], the BioNLP 2024 task on Lay Summarization of Biomedical Research Articles[2] and the Multilingual Lexical Simplification task[3], to mention a few examples.

A dataset is (almost always) provided to participants in a shared task, often consisting of training and development subsets. Tasks can be either *open* or *closed*; the former means that participants are allowed to exploit any resources/datasets outside of what was officially provided by the task organisers, while the latter constrains the participants to using only the dataset provided as part of the shared task.

A shared task's official evaluation (test) data is often not made available to participants until much later, when enough time has passed such that the participants are likely to have completed the development of their solution(s). Once the test data is released, participants are typically given just a short time to run their solution(s) on the test data in order to generate outputs, i.e., predictions, according to an agreed format. The participants then submit the predictions produced by their solution(s), which are evaluated against the hidden, gold standard annotations (labels) in the test set, using a number of metrics. The performance of the submitted solution(s) is finally reported in terms of those metrics.

Sometimes, a shared task consists of multiple *tracks*, each of which is focussed on a specific problem, domain or subtask. In the NLU coursework, we are organising a shared task on **pairwise sequence classification** with three different tracks. Your group will choose the one track that they wish to participate in.

[1] https://multiconer.github.io/
[2] https://biolaysumm.org/
[3] https://sites.google.com/view/mlsp-sharedtask-2024/home

II. Intended Learning Outcomes

- To build and implement two different solutions to a pairwise sequence classification problem
- To describe and present your solution in the form of a flash presentation
- To act as a responsible project partner, collaborate on project development/writing up, and contribute to planning/organisation

III. The Shared Task and its Tracks

The NLU shared task is focussed on pairwise sequence classification, the task of categorising a pair of sequences according to a predefined set of classes that capture the relationship between the two sequences. The three tracks that we are organising are:

A. Natural Language Inference (NLI)

   Given a *premise* and a *hypothesis*, determine if the hypothesis is true based on the premise. You will be given more than 26K premise-hypothesis pairs as training data, and more than 6K pairs as validation data.

B. Evidence Detection (ED)

   Given a *claim* and a piece of *evidence*, determine if the evidence is relevant to the claim. You will be provided with more than 23K claim-evidence pairs as training data, and almost 6K pairs as validation data.

C. Authorship Verification (AV)

   Given two sequences, determine if both sequences were written by the same author. You will be provided with around 30K sequence pairs as training data, and 6K sequence pairs as validation data.

It is worth noting that we are running the shared task in **closed** mode, hence, it is not permissible to use any existing datasets outside of those that we are providing, in developing your solutions.

IV. Instructions

This section provides an overview of what needs to be done as part of the coursework. Details pertaining to deadlines, output formats and submission sites are provided in succeeding sections.

1. Identify a project partner. As a group, consider the three available tracks and decide which track you wish to participate in. You might find it useful to have a look at a small trial dataset that contains examples of sequence pairs and their corresponding labels.

Once you have made a choice, register your group and your chosen track by completing an online form. Note that the online form is set up in such a way that you can edit your responses until the deadline (see Section VI); however, after this deadline, you will not be able to switch to another track.

2. Develop **two** different solutions to the problem that you have chosen, where any two of the following categories of approaches should be represented:

a. Unsupervised or traditional machine learning-based approaches
b. Deep learning-based approaches that do not employ transformer architectures
c. Deep learning-based approaches underpinned by transformer architectures

IMPORTANT NOTES:

- For each of the three tracks, we (the marking team) have developed baseline methods, one from each of the three categories of approaches above. You will likely obtain a passing mark even if your solution is unable to outperform the baseline performance, but higher marks will be given to solutions that obtain improved performance (in comparison with the baseline methods).
- You are free to choose any method/model that you wish to develop, and will likely obtain a passing mark even with just standard approaches (e.g., fine-tuning a BERT-based model). However, as can be seen in the Marking Rubric (Section VIII), higher marks will be given to more creative/inspired solutions.

3. Once the test data is released, run each of your developed solutions on it and produce predictions. Note that the test data will follow the same format as the files in the trial dataset, except that the **label** column will not be provided.

Submit the **predictions** in the required format by the coursework deadline. Your predictions will be evaluated against the (hidden) gold standard labels in the test set, using standard metrics such as F1-score. See Deliverable 1 in Section V.

4. Prepare the code for your solutions by the coursework deadline. Specifically, there should be code that can be run in "inference mode", henceforth referred to as **demo code**: given an input file containing test data, the code should load either of the solutions and generate the required predictions file. Overall, the code should be well-documented so as to help the marking team in identifying the code that corresponds to the training/building of each of your chosen approaches. See Deliverable 2 in Section V.

IMPORTANT NOTE: The demo code should be in the form of a **Python notebook** that contains lines for installing any required packages, to allow the marking team to run it on their own.

5. Create a **model card** for each of your solutions. A model card[4] is a document, typically a markdown file, that concisely describes a model. When sharing models with people outside of your own group, it is good practice to provide some key information that will help make your work reusable. To help you in creating your model cards, a model card template and a Python notebook for generating your customised model cards have been provided. Your generated model cards should be included in your code submission. See Deliverable 3 in Section V.

6. Prepare a **poster** that describes your solutions and how you developed them. This poster will be the basis of your flash presentation to the marking team during Workshop 4 (see Section VI), that both you and your partner are required to attend. Additionally, during the flash presentation you will showcase each of your solutions in real time using your demo code (see Item 4 above).

V. Deliverables and Submission Sites

Deliverable 1. The predictions produced by each of your two solutions on the test data. These will be submitted via Blackboard.

Deliverable 2. Your code for developing/training models and evaluating them (e.g., on the development set), as well as demo code, i.e., runnable code and models for generating predictions for a given input data file, in the form of a Python notebook. It is expected that your submission is organised or structured in such a way that:

- any code that was written to train models should be separated from the demo code; make sure that your models have been saved and can be readily loaded by the demo code;
- any code for evaluation (on the development set) is included; this should be separated from your code for training models

[4] https://www.kaggle.com/code/var0101/model-cards

- someone else outside of your team can follow the code; in other words, provide some documentation, i.e., in-line comments and a README explaining the code structure and how it is run.

Importantly, your README should provide: (1) attribution to any data sources you used or code bases you reused, and (2) links to any models that you yourself trained and stored on the cloud (i.e., OneDrive).

IMPORTANT NOTE: If any of your resources (e.g., models) are more than 10MB, please do not include them in your Blackboard submission and instead store them on the cloud.

Deliverable 3. A model card for each of your solutions, in markdown format.

Deliverable 4. A poster in PDF format. Please use landscape orientation, as the posters will be displayed on a screen. The size can be flexible but 36 x 48 inches is recommended.

IMPORTANT NOTE: All deliverables (Deliverables 1, 2, 3 and 4) should be compressed into **one zip file**, which should then be uploaded to the relevant Blackboard submission site.

VI. Release of Data, Timetabled Activities and Deadlines

Highlighted in red font below are critical deadlines/activities that every group should definitely pay attention to.

Event	Description	Site	Relevant Date
Release of coursework specifications and trial data	Trial data for all tracks published	Blackboard (look for "Trial Data for Shared Task")	01 March 18:00
Workshop 1	Q&A regarding coursework specifications; groups should be finalised	In-person	05 March 14:00-15:00
Deadline: Registration of groups	Groups register their members and their chosen track	Microsoft Form	08 March 18:00
Release of training data	Training data for all tracks published	Blackboard (look for "Training Data for Shared Task")	11 March 18:00
Launch of evaluation platform	EvalAI released as an optional supporting tool for benchmarking	See Blackboard Wiki page entitled "Optional Tool: EvalAI"	11 March 18:00
Workshop 2	Overview of the EvalAI platform; Q&A	In-person	12 March 14:00-15:00
Workshop 3	Advice on preparing posters and other deliverables, including model cards, for submission	In-person	09 April 14:00-15:00

Release of test data	Test data for all tracks published	Blackboard (look for "Test Data for Shared Task")	17 April 18:00
Deadline: Submission of all deliverables	Groups upload their system predictions, code, model cards and poster	Blackboard for all deliverables: system predictions, code, model cards and poster (look for "34812-cwk-S-Project")	19 April 18:00
Workshop 4	Flash presentations	In-person	30 April 15:00-17:00

VII. Teamwork

Each team member will be given the same mark for the coursework. One of the Intended Learning Outcomes of this coursework is focussed on acting as a responsible team member (see Section I), hence it is each team member's duty to ensure that tasks are delegated fairly and that there are equal contributions.

VIII. Marking Rubric

The coursework is worth 40 marks overall. The table below outlines the criteria comprising the marking rubric and the distribution of marks across these criteria.

System Predictions		
Competitive performance (Solution 1)	The solution obtains performance improvement performance that is statistically significant, in comparison with a baseline method for the same type of approach.	2
Competitive performance (Solution 2)	The solution obtains performance improvement that is statistically significant, in comparison with a baseline method for the same type of approach.	2
Implementation of Solutions		
Organisation and documentation	The code is well-documented and structured in a way that even developers outside of the team can follow it.	2
Completeness and Reproducibility	All necessary resources, including trained models and demo code, were provided for both solutions, allowing for reproducibility.	2
Soundness (Solution 1)	The design of the solution is technically sound; there were no obvious technical details that were ignored/misunderstood.	2
Soundness (Solution 2)	The design of the solution is technically sound; there were no obvious technical details that were ignored/misunderstood.	2
Creativity (Solution 1)	The design of the solution draws from literature on state-of-the-art approaches; the group was adventurous in that they went beyond the typical, standard approaches to the problem.	2

Creativity (Solution 2)	The design of the solution draws from literature on state-of-the-art approaches; the group was adventurous in that they went beyond the typical, standard approaches to the problem.	2
Evaluation	There is evidence that the group put effort into evaluating their solutions, outside of the supporting benchmarking tool (EvalAI).	2
Model Cards		
Formatting	Both model cards were prepared in the correct format	2
Informativeness (Solution 1)	The model card sufficiently describes the model, allowing any potential users to reuse/build upon the model.	2
Informativeness (Solution 2)	The model card sufficiently describes the model, allowing any potential users to reuse/build upon the model.	2
Accurate representation	The model cards accurately represent the implemented solutions.	2
Flash Presentation		
Live demo (Solution 1)	The demo code works out of the box.	2
Live demo (Solution 2)	The demo code works out of the box.	2
Poster content	The poster is informative; it is stand-alone, in that viewers can understand what the group implemented even without the group explaining it.	2
Poster aesthetics	The group put effort into making the poster visually appealing; there is good use of visuals.	2
Poster presentation	The poster presentation is engaging; the group was able to explain details of their implementations in a concise and compelling way.	2
Q&A (Solution 1)	The group was able to answer questions at a satisfactory level.	2
Q&A (Solution 2)	The group was able to answer questions at a satisfactory level.	2