ECS7001P - NN & NLP
Assignment 2: Social Media, Information Extraction and Dialogue

14th March, 2023

In this assignment, you will gain practice with one of the most popular basic methods in today's NLP: the use of large pre-trained language models. You will then apply the techniques you've learned to three new important domains: information extraction, coreference resolution and dialogue systems.

You will reach these goals by four stages:

- In Part A (Lab 7, week 7) you will implement regression model to assess humour in social media posts. [20 marks].
- In Part B (Lab 8, week 8) you will start working on Information Extraction - in particular, Named Entity Recognition (NER) [20 marks].
- In Part C (Lab 9, week 9) you will continue working on Information Extraction, developing a coreference resolver [20 marks].
- In Part D (Lab 10, week 10), you will move on to dialogue data, building flat and hierarchical dialogue act taggers [20 marks].
- In Part E (Lab 11, week 11), you will turn this into a sequence-to-sequence model to build a basic dialogue system [20 marks].

When all parts of the assignment are completed, you will have to submit two things:

- Your completed Python code;
- Your descriptions of what you did, with answers to any specific questions (instructions below).

The deadline for returning all completed parts of the assignment (Parts A, B, C and D) is **10:00:00 Monday 24th April 2023**.

Part A: Social Media Processing [20 marks]

For this part of the assignment, worth 20 marks in total, you must carry out the steps specified by the Lab script ("Lab 7: Social Media Processing" - separately provided). These marks are broken down among the different parts of the assignment as follows (please refer to the lab script):

1. Build and evaluate two regression models for humour rating prediction (a) without special pre-processing and (b) with special pre-processing applied to the training data (Tasks 1 and 2). [3 marks]

   In your report, include the output of the `model.summary()` command to show your model structure and training epochs (with loss values), as well as the mean squared error (MSE) scores over the test set for both models. Plot the histograms of the predicted and true humour ratings for the first model (a) and comment on the results. Describe the differences in performance between two models (a and b) and discuss why they occur.

2. Augment the training data twice for the regression model (Task 1 above) with the two following methods (a) synonym replacement from Wordnet and (b) deletion of random words (Task 3). [3 marks].
   In your report, include the output of the `model.summary()` command to show your model structure and training epochs (with loss values), as well as the MSE scores over the test set for both models. Describe the differences in performance between the models (a) and (b) and hypothesise why they occur.

3. One way to improve the performance is to perform ensembling of three regressors. Try this (Task 4) and see if it improves the performance of the model from Task 1. Use non-augmented data. [6 marks]

   In your report, include the output of the `model.summary()` command to show your model structure and training epochs (with loss values), as well as the MSE scores over the test set. Describe the difference in performance to the model from Task 1.

4. Build and evaluate the two following regression models with half of the size of the original training data: (a) multi-task learning simultaneously with the second regression task of offense rating and (b) single-task regressor from Task 1. [8 marks]

In your report, include the output of the `model.summary()` *command to show your model structure and training epochs (with loss values), as well as the MSE scores over the test set for both models. Describe the differences in performance between two models (a and b) and discuss why they occur.*

Part B - Information Extraction 1: Training a Named Entity Resolver [20 marks]

For this part of the assignment, you must carry out the steps specified by the Lab script ("Lab 8 - A named-entity recognizer using Keras" - separately provided). These marks are broken down among the several parts of the assignment as follows (please refer to the lab script):

1. Task 1: Create a bidirectional GRU and Multi-layer FFNN [10 marks].
   In this task, you need to complete the `build()` *method. For full marks, your report must include the code and your explanation.*
2. Task 2: Form the predicted named entities [10 marks].
   Again, you must include the code and your explanation. In addition, you should provide the F1 score achieved by your model on the test set, and the output your script produces (model summary, training, and test set result) - you are given the output of a model solution for sanity check. DO NOT WORRY IF YOUR RESULTS ARE SLIGHTLY DIFFERENT FROM THOSE PROVIDED!

Part C - Information Extraction 2: A Coreference Resolver [20 marks]

For this part of the assignment, you must carry out the steps specified in the Colab script ("Lab 9 - Coreference Resolution" - separately provided). These marks are broken down among the several parts of the assignment as follows (please refer to the lab script):

1. Task 1: Preprocessing (See section 4.4) [6 marks].
   In this task, you need to complete the `get_data()` method. to produce mention and mention pair representations. For full marks, your report must include the code and your explanation.

2. Task 2: Building the model [6 marks].
   In this task, you need to write code for the `build_model()` function to assign embeddings, run a biLSTM over those, and then a FFN to compute the mention representation. Again, you must include the code and your explanation.

3. Task 3: Coreference evaluation (Section 6.2) [5 marks].
   In this task, you need to write code for the `eval_coref()` function. Again, you must include the code and your explanation.

4. Task 4: Some questions (Section 8)[3 marks].
   In this task, you need to answer three questions about alternative solutions.

Part D - Dialogue 1: Dialogue Act Tagging [20 marks]

For this part of the assignment, you must carry out the steps specified by the Lab script ("Lab 10 - Dialogue Act Tagging" - separately provided). These marks are broken down among the several parts of the assignment as follows (please refer to the lab script):

1. Task 1: Implementing an utterance-based tagger, using standard text classification methods from lectures [5 marks].
   Your report must include the code, accuracy figures and explanation as specified in the script.

2. Task 2: Minority DA tag class analysis and utterance-based tagger with re-balanced weighted cost function [5 marks].
   Your report must include the code, accuracy figures and explanation as specified in the script.

3. Task 3: Implementing a hierarchical utterance+DA-context-based tagger [10 marks].
   Your report must include the code, accuracy figures and explanation, including analysis of the effect on minority DA classes and an example of how the model outputs have changed, as specified in the script.

Part E - Dialogue 2: A Conversational Dialogue System [20 marks]

For this part of the assignment, you must carry out the steps and answer the questions as specified by the Lab script ("Lab 11 - Creating an End-To-End Dialogue System" - separately provided). The marks are broken down among the different parts of the assignment as follows (please refer to the lab script):

1. Task 1: Implementing the encoder [2 marks].
   You must include the code as specified in the notebook, together with an explanation of the architecture you have specified and why.

2. Task 2: Implementing the decoder with attention [3 marks].
   You must include the code as specified in the notebook, together with an explanation of the architecture you have specified and why.

3. Task 3: Investigating the behaviour and the properties of the encoder-decoder network [15 marks].
   You must include your answers to the questions in the notebook, giving examples and/or explaining the evidence for your answers.

Submission

Please submit one zip file with all your answers to Parts A-E together.

As well as code, you should include text explanations, descriptions and answers to specific questions as necessary and as specified above. Code should be in Python; explanatory text can be either as a separate report in PDF format (not Word, please), or included together with the code as a Jupyter/Colab notebook.

For each section, marks will be awarded for correctness of code and classifier performance, but also for clarity of explanations and justifications.