Assignment 2: sequence labelling

Text mining course

This is a **hand-in assignment for groups of two students**. Send in via Brightspace **before or on Tuesday November 12**:

- Submit your report as PDF and your python code as separate file. **Don't upload a zip file containing the PDF** (the Python code might be zipped if it consists of multiple files).
- Your report should **not be longer than 3 pages** (being concise is an important lesson!)
- Do not copy text from external sources (other groups, web pages, generative models such as chatGPT). Turnitin is enabled and a large overlap will be reported to the Board of Examiners. Reuse of code is no problem (and intended because we build on existing packages and tutorials).

Goals of this assignment

- You can pre-process existing annotated text data into the data structure that you need for classifier learning.
- You can perform a sequence labelling task with annotated data in Huggingface.
- You can perform hyperparameter optimization.
- You can evaluate sequence labelling with the suitable evaluation metrics.

Preliminaries

- You have followed the Huggingface tutorial on token classification https://huggingface.co/learn/nlp-course/chapter7/2 and its preliminaries (**exercises week 6 and 8**).
- You have all the required Python packages installed.

We are going to train an NER classifier for six entity types in the archaeological domain. The data can be downloaded from https://github.com/alexbrandsen/Archaeo-NER-data-English/tree/main/5-folds-test-train-val-split/fold1 (the data was split in five different ways for 5-fold-cross validation; we only use fold1).

Tasks

1. Download the data from fold1 of the data. There are 3 IOB files: `train.txt`, `val.txt`, `test.txt`.
2. Convert the IOB data to the correct data structure for token classification in Huggingface (words and labels like the conll2023 data in the tutorial) and align the labels with the tokens. Note that since you are working with a custom dataset, the data conversion is a necessary step for using the Huggingface training function.

3. Fine-tune a model with the default hyperparameter settings on the **train** set and evaluate the model on the **test** set. These are your baseline results.
4. Set up hyperparameter optimization (HPO), use the **val** set as validation. Optimize at least three hyperparameters (learning rate, batch size and weight decay). You can choose your own way to implement this and select your own grid. After the model has been optimized, evaluate the result on the **test** set.
5. Extend the evaluation function so that it shows the Precision, Recall and F-score for each of the entity types (location, artefact, etc.) on the **test** set. Include the metrics for the B-label of the entity type, the I-label, and the full entities.
6. Look up the definitions of *macro- and micro-average scores* and compute the macro- and micro average F1 scores over all entities.

Write a report of at most 3 pages in which you:

- describe the task and the data (give a few statistics. What is the distribution over the entity types?) and briefly describe two challenges of the task and the data.
- show your results:
    - a results table with both the baseline results and the results after hyperparameter optimization (do not report results on the val set, only on the **test** set): a table with Precision, Recall, F-score for both settings.
    - a table with the results on the test set after hyperparameter optimization for the different entity types (Precision, Recall, F-score for B, I, and the full entities), and the macro- and micro F1 scores.
- write brief conclusions. Address the following questions:
    - What is the effect of hyperparameter optimization on the quality of the model?
    - What does the difference between scores for different entity types tell you?
    - Where does the difference between macro- and micro-averaged F1 scores come from?

Grading

Maximum 2 points for each of the following criteria:

- General: length correct (2-3 pages) and proper writing + formatting
- Description of the task and the data, with description of 2 challenges
- Baseline results with default hyperparameter settings and results with optimized hyperparameter settings
- Results for the different entity types (after HPO if you completed task 4, otherwise without HPO)
- Sensible conclusions, briefly addressing the questions listed above.