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 Monday November 14**:

- 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

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 hyperparameter optimization
- You can perform a sequence labelling task with annotated data in CRFsuite
- You understand the effect of using different features and context sizes
- You can evaluate sequence labelling with the suitable evaluation metrics

Preliminaries

- You have followed the CRFsuite tutorial https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html
- You have all the required Python packages installed
- You might need to change your sklearn version to 0.22.2 to make it compatible to CRFsuite:

```
pip install scikit-learn==0.22.2 --user
```

We are going to train an NER classifier for the task "Emerging and Rare entity recognition" from the Workshop on Noisy User-generated Text (W-NUT). The description of the task can be found at https://noisy-text.github.io/2017/emerging-rare-entities.html (I put the data itself on Brightspace)

Tasks

1. Download `W-NUT_data.zip` from the Brightspace assignment and unzip the directory. It contains 3 IOB files: `wnut17train.conll` (train), `emerging.dev.conll` (dev), `emerging.test.annotated` (test)
2. The IOB files do not contain POS tags yet. Add a function to your CRFsuite script that reads the IOB files and adds POS tags (using an existing package for linguistic processing such as Spacy or NLTK). The data needs to be stored in the same way as the benchmark data from the tutorial (an array of triples `(word,pos,biotag)`).

3. Run a baseline run (train -> test) with the features directly copied from the tutorial.
4. Set up hyperparameter optimization using the dev set and evaluate the result on the test set.
5. Extend the features: add a larger context (-2 .. +2 or more) and engineer a few other features that might be relevant for this task. Have a look at the train/dev data to get inspiration on potentially relevant papers.
6. Experiment with the effect of different feature sets on the quality of the labelling.

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

- describe the task and the data (give a few statistics. What are the entity types?)
- describe the features;
- show your results (Precision, Recall, F-score for the B and I tags):
    - a results table with both the baseline results and the results after hyperparameter optimization (do not report results on the dev set, only on the test set);
    - show a results table with a number of experimental results from your changes in the feature set.
- write brief conclusions.

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
- Description of the adapted features
- Baseline run with features from tutorial & experimental runs with adapted features (show results in table: Precision, Recall, F-score for the B and I tags)
- Sensible conclusions