Introduction to NLP: Assignment on Text Classification and Sequence Labeling

This assignment relates to the Text classification and Sequence labeling themes of the introduction to NLP (courses Deskriptiv analytik / Machine learning for descriptive problems), and will focus on gaining some practical, hands-on experience in building and training simple models for these tasks.

The assignment is handed in as a Jupyter notebook (or a PDF render thereof) containing the code used to solve the problem, output presenting the results, and, most importantly, notes that present the students' conclusions and answer questions posed in the assignment. The assignment is solved individually or in groups of 2-3 students, and **a description of how each member has contributed is required** (e.g., in case responsibilities have been divided, specify how). The report is **due on February 27**, submitted through the course platform.

In case you don't want to work on your own computer, instructions how to use CSC Notebooks can be found here: https://github.com/TurkuNLP/intro-to-nlp/blob/master/instructions-demo-environment.pdf

Assignment steps/questions:

1. Test sklearn's TfidfVectorizer in place of CountVectorizer on the IMDB data. Do you see any difference in the classification results or the optimal C value?

2. Test different lengths of n-grams in the CountVectorizer on the IMDB data. Do you see any difference in the classification results or the optimal C value? Do these n-grams show up also in the list of most significant positive/negative features?

3. In the data package for the course (http://dl.turkunlp.org/intro-to-nlp.tar.gz), the directory language_identification contains data for 5 languages. Based on this data, train an SVM classifier for language recognition between these 5 languages.

4. If you completed (3), toy around with features, especially the ngram_range and analyzer parameters, which allow you to test classification based on character ngrams of various lengths (not only word n-grams). Gain some insight into the accuracy of the classifier with different features, and try to identify misclassified documents - why do you think they were misclassified?

5. [OPTIONAL FOR BSC STUDENTS] On the address universaldependencies.org, you will find datasets for a bunch of languages. These come in an easy-to-parse, well-documented format. Pick one language that interests you, and one treebank for that language, and try to build a POS tagger for this language. You can use the 4th column "UPOS" https://universaldependencies.org/format.html Report on your findings. If you have extra time, try to experiment with various features and see if you can make your accuracy go up. You can check here https://universaldependencies.org/conll18/results-upos.html what the state of the art roughly is for your selected language and treebank. Did you come close?