SNLP assignment 7: detecting offensive language (Germeval 2018)

Deadline: Aug 8, 2018 @ 10:00 CEST

Why am I doing this?

- Experiment with text classification
- Work on a real-world NLP problem
- (optionally) gain experience with describing your systems and reporting your results

In this assignment you will work with a current, real-world text classification problem as defined by the Germeval 2018 shared task.[1] The task aims to detect offensive language in social media posts in German.

The team formation rules for this assignment is less strict than the earlier assignments. You can form teams of up to three members with whom you may have worked on earlier assignments.

The shared task deadline aligns with the assignment deadline well. Although it is not required for the assignment, you are encouraged to participate in the shared task. However, the shared task organizers only accept a small number of participating teams per site. As a result, **if you intend to participate in the shared task, please contact the instructor latest on July 2, 2018**.[2] Note that the shared task participation requires writing a system description paper as well as submitting predictions on a test set to be released later.

A copy of the training data is placed in your homework repository as a tab-separated file, with the name `germeval2018.training.txt`.[3] Please see the shared task site for further description of the data.

Exercise 1. *The task*

Create, train (and tune) a classifier for Germeval 2018 'task 1' (binary classification).[4] You can use any machine learning method, (or set of methods) of your choice. Update your `README.md` file with the following information.

- A brief description of your system, e.g., preprocessing, classifier(s), the features, feature selection/weighting methods, use of external data ...
- Explanation of how to
  - *train* the system using an input file with a set of hyperparameters[5]
  - How to *tune* the hyperparameters of your model[6]
  - How to obtain the predictions on a test file (possibly training the model using a set of hyperparameters, or using a model saved after tuning)
- The best hyperparameter setting, along with an explanation of your method of tuning (e.g., through a random search using a range of hyperparameters)
- The macro-averaged precision/recall/f-score values obtained on the development set or via k-fold cross validation

Unlike earlier assignments, the amount of work you may put on this task is open. Your score and/or the amount of effort will determine half of the grade you get from this assignment.

Do not forget to push all your source code to your repository as well as updating the `README` file.

You should be able to get reasonable results with rather modest computational power. If your personal computing equipment is a limiting factor, you may try the department server `urobe.sfs.uni-tuebingen.de`.

[1] `https://projects.fzai.h-da.de/iggsa/`.

[2] We will organize a single participating team, possibly using combined results from multiple assignment teams.

[3] There is no designated development set. You should use either k-fold cross validation, or a reasonable development set split for tuning your model(s).

[4] For the class assignment, you are only required to work on 'task 1'. However, if you intend to participate in the shared task, you can choose to work on only one of the tasks, or both.

[5] You are highly recommended to use command line arguments for specifying input files, or other options (e.g., hyperparameters of your model).

[6] Tuning the parameters manually is acceptable for the assignment. Just indicate it if you did it so.