Assignment 4:

Content warning: this assignment contains an analysis of offensive language examples.

Learning goals:

- Applying and comparing state-of-the-art models for hate speech detection
- Getting insights into generalizability issues in the area of hate speech detection by conducting cross-domain experiments

Aim of the assignment:

- Develop and execute different types of models for hate speech detection (transformers, conventional machine learning approaches, lexicon-based approaches, others)
- Evaluate their performance both in the in-domain and cross-domain experimental setups
- Perform a quantitative error analysis both in the in-domain and cross-domain setups
- Discuss the difference in performance in the in-domain and cross-domain setups
- Submit the notebook and write a report about your findings

1. Data

In this assignment, you will be working with the **OLIDv1 dataset**, which contains 13,240 annotated messages (tweets) for offensive language detection. The detailed description of the dataset collection and annotation procedures can be found here. This dataset was used in the SemEval 2019 shared task on offensive language detection (OffensEval 2019).

This assignment focuses on Subtask A (identify whether a tweet is offensive or not). We preprocessed the dataset so that label ‘1’ corresponds to offensive messages (‘OFF’ in the dataset description paper) and ‘0’ to non-offensive messages (‘NOT’ in the dataset description paper) and selected a subset of the OLIDv1 train dataset of the same size and with the same label distribution as the HASOC train dataset (described below): olid-train-small.csv. The test set is the same as used in the SemEval 2019 shared task: olid-test.csv.

The second dataset you will be using in this assignment is the **HASOC dataset** (only the training set of the English dataset composed of tweets and Facebook messages). The dataset was preprocessed in the same way as the OLIDv1 dataset: label ‘1’ corresponds to hateful/offensive messages and ‘0’ to non-hateful/non-offensive messages.

The preprocessed training (olid-train-small.csv) and test (olid-test.csv) partitions of the OLIDv1 dataset and the HASOC train dataset (hasoc-train.csv) can be found here.

Three datasets:

OLID-train-small
OLID-test
HASOC-train

2. Experimental setup

2.1. In-domain experiments

Train the models described below on the OLIDv1 train set (olid-train-small.csv) and evaluate on the OLIDv1 test set (olid_test.csv).

2.2. Cross-domain experiments

Train the models described below on the HASOC train dataset (hasoc-train.csv) and evaluate on the OLIDv1 test dataset (olid-test.csv, same as in 2.1) .

3. Methods

3.1 Transformer-based models (use at least two models)

Run your notebook on colab, which has (limited) free access to GPUs.

You need to enable GPUs for the notebook:

- navigate to Edit → Notebook Settings
- select GPU from the Hardware Accelerator drop-down

1. Install the simpletransformers library: *!pip install simpletransformers*
   (you will have to restart your runtime after the installation)
2. Follow the documentation to load at least two pre-trained language models, for example, BERT (e.g., ClassificationModel('bert', 'bert-base-cased')), RoBERTa, XLNet.
   Alternatively, you can use transformer models re-trained for hate speech detection, e.g., HateBERT (Caselli et al., 2021) and/or fBERT (Sarkar et al., 2021).
   * Select at least two transformer models from step 2 (general and/or re-trained).
   Choosing more than 2 models can be useful for assignment 5.
3. Fine-tune the models on olid-train-small for the in-domain experiments and on the hasoc-train for the cross-domain experiments, and make predictions on the OLIDv1 test set (as also described in the Experimental setup part).

   Note: you can use models with the default hyperparameters.
   Note: if you optimize the hyperparameters, set aside a subset of the training data as your evaluation (or development) set.
   Note: do not forget to save your model, so that you do not need to fine-tune the model each time you make predictions.

3.2 Other models (use at least one model)

Use at least one of the following approaches (you may use approaches from assignment 3 and/or explore other models):

- A conventional machine learning model (e.g., SVM, Logistic Regression, Naive Bayes)
- A lexicon-based approach: either lexicon lookup method or a machine learning approach with (additional) features extracted from a lexicon
- CNNs
- LSTMs
- BiLSTMs

Run the model(s) both in the in-domain and cross-domain setups.

Optional: you can also explore the performance of recent generative Large Language Models (LLMs) such as ChatGPT (in addition to 3.1 and 3.2).

4. Analysis

Quantitative analysis of the models performance and discussion:

- How do the examined models perform in the in-domain and cross-domain setups in terms of macro-averaged precision, recall, and F1-score (also provide results for each class and confusion matrix)?
  - Which approach performs best in the in-domain setup?
  - Why do you think one of the approaches performs best?
  - Is the model that shows the best results in the in-domain setup also provides the best results in the cross-domain setup? Why is this (not) the case, in your opinion?
- Is there a drop in performance in the cross-domain setup (as compared to the in-domain setup)? Quantify the drop in performance in the cross-domain setup if observed.
- Discuss potential explanations for the difference in performance in the in-domain and cross-domain setups, for instance, can the difference be partially explained by the different length of messages in the training and test data? Can it be explained by different topical focuses of these datasets? What other characteristics of the datasets can be the reason for a cross-domain drop?

5. Write an academic report with the following sections: data, models, experiments, results and analysis. Do not forget to refer to literature.