ALC - MIARFID
Laboratory Lesson 1
Sexism Identification in Tweets

19th February 2026

Objectives

In this laboratory session, we aim to design and implement a computational system capable of automatically detecting sexism in Tweets. This system will use text-based natural language processing techniques and employ either machine or deep learning approaches. More specifically, this session focuses on:

- **Preparing the dataset** and familiarizing with the addressed task
- Data pre-processing, exploring multiple **feature extraction** methods
- **Training and evaluating** machine/deep-learning models

The EXIST2025 Challenge

This lab session is designed to simulate the participation in the **Task 1.1: Sexism Identification in Tweets** of the EXIST 2025 Challenge[1]. This binary classification task, as detailed below, includes tweets in both Spanish and English, encouraging the development of multilingual models. Here you can see an example of the tweets we can find in the training dataset of EXIST2025:

Sexist Tweet Training Sample (`id_EXIST`: 100174)

No like.. patriarchy.. ella es literalmente una MAMASITA q está buenísima y a la q su novio maltratador [...]

[1] `https://nlp.uned.es/exist2025/`. In the new EXIST2026 call, this task was removed.

Non-Sexist Tweet Training Sample (`id_EXIST`: 200860)

🙌When your super excited today was your first meet with My Sisters Keeper girls empowerment club -you race home and design your poster #mystrongYounglady #shesgoingplaces ❤️#engage #haveavoice #makeitknown #EmpowerWomen #supportive #promote @ladybug548 @MsResource @NYCSchools https://t.co/KQovXSzflE

Note the informal and natural language, as well as the use of emojis, hashtags, user mentions, and links to web pages.

Lab Materials

All necessary data and metadata required for this laboratory session can be accessed through the subject's `PoliformaT` platform, where we can find the:

- `dataset_task1_exist2025/` directory, which contains two JSON files specifying our training and test samples of the EXIST2025 Task 1.1 challenge dataset, along with their associated tweet and metadata.
- `golds_task1_exist2025/` directory, which contains a JSON file specifying the gold standards for the training set. They are the reference ground-truth labels to evaluate the performance of your models.

An important point to note is that, as in a real challenge, **labels for the test set are not provided**. Below, we outline various implementation details, including strategies for handling model evaluation in this type of scenarios.

Implementation Details

Before starting the lab session, various aspects regarding the evaluation procedure in challenges such as EXIST2025 should be clarified.

· There is no development set!!

As mentioned above, some challenges do not provide an official development set. In such cases, two strategies are typically employed to evaluate our approaches:

- **Creating our own development set.** For example, we can use 20% of the training data for evaluation. After identifying our best-performing settings, we can train the model again on the entire training set to generate the final test predictions. For robust evaluation, it is crucial to ensure the development set is well-balanced regarding classification labels.

- **Cross Validation.** Another common strategy is to use k-fold cross-validation, typically with $k = 5$ or 10, depending on the dataset size. Our models are trained using $k - 1$ folds, while the remaining one is reserved for testing. We then ended up with k different models. This approach allows us to evaluate across the entire training dataset. Similarly, we can retrain the model again with all the data, once the hyperparameters were determined. Besides, we should ensure a well-balanced dataset splitting.

In this lab session, you can choose the strategy you prefer, as long as it ensures a consistent evaluation framework across your experiments.

> Both strategies are valid but... do you think one is better than the other? In which scenarios might each strategy be more or less adequate? Why?

· Dealing with Multiple Annotators

By inspecting the JSON files of our dataset (e.g., the training set defined in `dataset_exist2025/training.json`), we can observe how, for each sample, there are multiple metadata keys. The most relevant for us are: `id_EXIST`, `tweet`, and `label_task1`, providing the sample identification number, the tweet text message, and task labels, respectively. **However**, we do not have a unique single label, but a list reflecting the opinion of 6 different annotators[2]. You may be wondering: **What are we going to do?** We will work with *hard labels* and thereby follow a **majority voting** strategy. Please note that, for consistency in evaluation, samples with tied annotations have been excluded from this and subsequent laboratory sessions.

Inspecting the Dataset Metadata JSONs

"200875": {
——"id_EXIST": "200875",
——"lang": "en",
——"tweet": "Since men can get pregnant, then men can [...]"
——...
——"labels_task1": ["YES", "YES", "NO", "YES", "YES", "NO"],
——...
}

· Using the PyEvALL Toolkit

Regarding model evaluation, we follow the EXIST challenges guidelines and, therefore, we use the PyEvALL toolkit[3]. This toolkit will be used throughout

[2] This is particularly useful for studies on the so-called *learning with disagreements*, aimed at estimating models that integrate the subjectivity inherent in complex tasks like this one.

[3] `https://github.com/UNEDLENAR/PyEvALL/tree/main?tab=readme-ov-file`

all our lab sessions and the next EXIST2026 challenge, so it's important to familiarize yourself with some key details about its usage.

First, and most importantly, install the package via *pip* using the following command: `pip install PyEvALL`. Now, before addressing the code for evaluation, let's first review the previously commented gold standard JSON files:

Inspecting the Evaluation Gold Standard JSONs

[{
—"test_case": "EXIST2025",
—"id": "100001",
—"value": "YES"
}, {
—"test_case": "EXIST2025",
—"id": "100002",
—"value": "NO"
}, {
—...]}

This type of JSON file is the one expected by PyEvALL to perform the evaluation of your models. Consequently, we need to create a similar file with our sample-level predictions. Once we have trained our model and obtained its predictions, we can create, based on the code snippet examples described in the GitHub repository of the PyEvALL toolkit, our prediction file as follows:

```
import pandas as pd

preds_dict = {
    'test_case': ['EXIST2025','EXIST2025'],
    'id': ['100001','100002'],
    'value': ['YES','NO'],
}

preds_df = pd.DataFrame(preds_dict)
output_path = './thepath/toyour/predictions.json'
with open(output_path, 'w', encoding='utf-8') as output_file:
    output_file.write(preds_df.to_json(orient='records'))
```

Of course, the code above represents an example, assuming we only have two samples in our testing set. Please modify the code according to your needs.

For the final evaluation, PyEvALL provides a set of objects and functions that facilitate the process, as illustrated in the example code snippet below. By comparing the obtained predictions with the corresponding ground-truth labels of the gold standard, the toolkit generates the expected report, which can be printed directly or handled as a dictionary, as noted in the comments. For other

metrics and further details, please refer to the toolkit's GitHub repository.

```python
from pyevall.evaluation import PyEvALLEvaluation
from pyevall.utils.utils import PyEvALLUtils
from pyevall.metrics.metricfactory import MetricFactory

test = PyEvALLEvaluation()

preds = './thepath/toyour/predictions.json'
labels = './thepath/toyour/goldgroundtruth.json'

metrics = [
    MetricFactory.Accuracy.value,
    MetricFactory.FMeasure.value,
]

params= dict()
report = test.evaluate(preds, labels, metrics, **params)
report.print_report()
# report.report, if you want to access the dictionary :)
```

Although alternative toolkits, such as Scikit-learn (e.g., its function `classification_report(labels, preds)`), may be used for evaluation, it is important to emphasize that **the use of PyEvALL is mandatory** for this and all subsequent laboratory sessions. This requirement ensures that you become familiar with the default evaluation toolkit specified in the guidelines of the EXIST challenges.

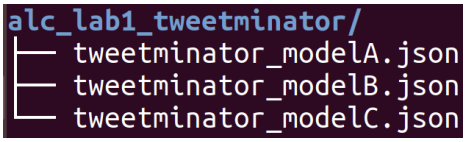
What Should You Do?

In this lab session, you should: (i) prepare the dataset; (ii) explore text-based feature representations; (iii) and estimate machine/deep-learning models to address the binary-class task of sexism detection in Tweets. All this may involve reading scientific papers related to the task in order to gather insights on which features or models might be useful.

For reference, you can consult the official EXIST2025 challenge overview paper [1]. In particular, focus on Subsection 6.1.2 (Table 7), where you will find the leaderboard with the final ranking of the task under hard-label settings. A brief description of the models involved is found in Section 5.

As you can observe, the overall F_1-score of the top-performing models is around 78%. The aim of this lab session is not to surpass the state of the art, but to learn. Once you finish your experiments, you must submit via `PoliFormaT`:

- **A ZIP file with your top-3 models.** This compressed file should thereby contain three JSON files, one for each of your top models. Each JSON file must include the model's predictions for the evaluation dataset, formatted according to the guidelines provided in this lab bulletin. *This will be part of our in-home challenge :) Who will be the winner?* Please ensure that your JSON files are well-organized, as exemplified by the image below, from which we can infer that our files should follow the naming template: `<GROUP-ID>_<MODEL-ID>.json`[4].



- **A single-page PDF report briefly describing your approach.** This can include details on the pre-processing steps, feature extraction methods, and the rationale behind your model architecture choice.

Deadline: 4th March 2026, 23h55

You can work individually or in groups of up to two people.
In the case of group submissions, only one member should upload the final work, clearly indicating the names of all participants in the corresponding task available and prepared in `PoliFormaT`.
Please register your team by clicking on this link.

References

[1] Laura Plaza, Jorge Carrillo-de Albornoz, Iván Arcos, Paolo Rosso, Damiano Spina, Enrique Amigó, Julio Gonzalo, and Roser Morante. Overview of EXIST 2025: Learning with Disagreement for Sexism Identification and Characterization in Tweets, Memes, and TikTok Videos. In *International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF)*, pages 266–289. Springer, 2025. Link to PDF.

[4] **Warning:** any model name that can be considered offensive or inappropriate may result in a penalty or, in severe cases, disqualification of the laboratory work.