# CE807 – Assignment 2 - Final Practical Text Analytics and Report

School of Computer Science and Electronic Engineering - University of Essex

## Assignment Due at 11:59:59am on 25/04/2023

**Electronic Submission**

URL: https://www1.essex.ac.uk/e-learning/tools/faser2/

*Please also see your student handbook for rules regarding the late submission of assignments*

## On Plagiarism

The work you submit must be your own. Any material you use, whether it is from textbooks, classmates, the web, or any other source, must be acknowledged in your work.

All submissions are fairly and transparently checked for plagiarism. Please make sure that you provide frequent citations. But also make sure that each sentence written is originally yours, i.e., the material is read, understood, and the report is written using your own words and your own language only. Do not copy and paste and rephrase the copied text.

There are many different forms of what is considered plagiarism. For example, based on the feedback from the SAO officer, many students were unaware that, e.g., copying entire paragraphs without clearly identifying them as quotes, etc. is a form of plagiarism, etc. Thus, please check back with your scientific writing module before you submit it! Please see the university's academic integrity policy.

Further note that also plainly reusing software code or merely slightly adapting existing software code and submitting it as one's own fulfils the matter of plagiarism. Cite any code that you reuse, too.

In 2022, ~20% of the submitted reports were plagiarised. There were also multiple cases of software code plagiarism. This number is too high and shall be 0% in 2023!

**MOTIVATION:** The task of identifying offensive speech is of utter importance to both the public and companies to have a better user experience.

**OBJECTIVE:** After Assignment 1 focused on the theoretical aspects of text classification, the objective of this assignment is to get practical experience in designing, implementing, running, and scientifically evaluating your own classifier. You will train, validate and evaluate using the subset OLID dataset. During training, you could use other datasets of your choice with justification.

**Content Warning: The OLID dataset contains examples of harmful content (hate, abuse, and negative stereotypes). The assignment does not support the use of harmful language or any of the harmful representations in the dataset.**

**SUBMISSION, ASSESSMENT, AND RULES**

- This assignment counts towards 75% of the overall mark for CE807.
- The assignment is to be done individually.
- Be sure to put your registration number/student id as a comment at the top of all code and other files. Furthermore, the assessment is blind, i.e., **do not put your name on any document or provide personally identifiable information**. Your name as/in the file name is also not allowed.
- All coding needs to be done in the Google Colab. Make sure your code is properly documented and it prints the desired outputs using the "print" statement whenever required. You are allowed to use existing code with appropriate citations. All codes need to be in one Google Colab only.
- In the Google Colab for different Models, use the Text cell to explain the code cell.
- The assignment must be submitted in 5 files named ['report.pdf', 'presentation.pdf', 'code.py', 'code.ipynb', 'links.txt']. Note that you must upload these separately, not as one zip/tar file.
- You must also provide a shareable link to the Google Colab code, Google Drive and Zoom cloud link of the recorded presentation in the report. 'links.txt' will contain following 3 lines,
    - **For the sharable Google Colab link,**
    - **For the sharable Google Drive folder link where your splitted files trained models and outputs are stored,**
    - **recorded presentation zoom cloud link.**
- Your code should assume that GOOGLE_DRIVE_PATH_AFTER_MYDRIVE = **'./CE807/Assignment2/student_id/'**. **Your code should also be in the same directory**. Please see Lab 06 & 07 scripts on how to set this. This drive should also contain ['train.csv', 'valid.csv', 'test.csv'], and your code should automatically get this. You **student_id** is your 7/8 digit faser number.
- Your code must also save models in **MODEL_X_DIRECTORY = os.path.join('gdrive', 'MyDrive', GOOGLE_DRIVE_PATH_AFTER_MYDRIVE, 'models', 'X'),** where X = 1 and 2. Your code must load the model from this directory at the testing time and then test on the testing data.
- **Your output file based on the test file will be named output_test.csv and will have fields id, tweet, label and out_label. Note that, id, tweet, label come from test.csv, and out_label out_label your model's output, where out_label =[OFF,NOT]. You need to save file in the respective model folders.**
- **You must print `train`, `validation` and `test` performance measures. You must also print `train` and `validation` loss in each `epoch`, wherever you are using `epoch`, say in any deep learning algorithms.**
- **You must share './CE807/Assignment2/student_id/' Gdrive folder to ce807.essex@gmail.com**

- Further, instruction for report format, code structure, dataset path, etc., is provided separately. You must follow that.
    - Code Instructions: https://colab.research.google.com/drive/1YDoRcSlP6irZR7vhdSxLc4XKzHTrilOo?usp=sharing
        - You must have train_method1, test_method1 for method 1 and train_method2, test_method2 for method 2 to perform full training and testing. This will be evaluated automatically, without this your code will fail and no marked.
    - Report Instructions: https://drive.google.com/file/d/1-VEZo13sMji2bXagaLln9lfKH915fMBJ/view?usp=share_link
    - Report Overleaf: https://www.overleaf.com/read/gyqzsvdhnjgp , look at 'report_final.tex', will provide you with sample examples on how to insert figures, tables, etc.
    - Output Directory Structure: https://drive.google.com/drive/folders/1okgSzgGiwPYYFp7NScEt9MNVolOlld1d?usp=share_link
- **You code and data will be evaluated automatically, so if you don't follow these instructions, you will not get marks because it would give error.**
- **Initialize all seeds to your student_id i.e. torch.seed(student_id), np.random.seed(student_id)** This will ensure that all codes are reproducible and each student have different set of model initialization and data split.

**Important Note**: You are free to use any software you like for this assignment. The final code must use Google Colab. You start with using a small portion of the dataset. In most cases, the Google Colab should be sufficient. However, if you think you need more computing, you could use the HPC of the university. For details, see: https://hpc.essex.ac.uk/

**Furthermore, you are encouraged to explore and exploit additional datasets for hate/offensive speech. https://hatespeechdata.com/ provides an excellent resource for that**

## Task Description: Offensive Speech Classification

Text classification is one of the standard tasks in text analytics. The assignment's objective is to classify offensive speech using the subset of the OLID dataset. You will use the subset of the OLID dataset provided to you as train/valid/test splits. Note you have to use this version of the dataset only. For a given data point, you will have to decide whether the data point is offensive or not. Please read the paper for more details, note that you only have to do **Level A: Offensive language Detection** (see Sec 2.1 in the paper).

## _____ TASKS _____

**TASK 1: Model Selection**

Whenever you develop a new classifier or other text analytics software, you must select the best possible model considering the resource and show that your system outperforms state of the art on certain conditions. This part of the assignment aims to explore the landscape of offensive speech classification. In Assignment 1, you have already identified some methods, now, explore further to finalize **two methods**.

Note that you can't select exactly the same classifier discussed in the lectures and labs. **Also, you can't select both models using pre-trained large language models (LLM) like BERT, RoBeRTa, GPT-2, hate-BERT etc.** However, with modification is allowed like LLM is one part of your actual method/model.

---

**TASK 2: Devising and training your own classifier**

This task will only be marked if you have completed Task 1.

For this task, you will develop your own offensive speech classifier models selected in Task 1. Doing this will involve:

- Identifying the approaches and features you want to extract;
- Developing code to extract these features from the text (and weigh them if you want to use more than simply binary features);
- Train a classifier that uses these features;
- Evaluate the performance of the classifier using a scientifically sound methodology.
- You must use Google Colab's GPU for the deep learning methods, which will make your training faster.
- Initially, start with a subset of the large datasets. Subsequently, scale the classifier to include more and more data until you use the entire dataset or very large parts. You can request access to the HPC CERES with your student account, see: https://hpc.essex.ac.uk/

---

**TASK 3: Data Size Effect**

This task will only be marked if you have completed Tasks 1 & 2.

For this task, you will use models to train on different data sizes and testing on it. Doing this will involve:

- Splitting training data into 4 sub-sets of [25%, 50%, 75%, and 100%] of the original dataset
- Train all your models on these 4 sub-sets and report the results
- You will have to report both validation and testing set performance as a plot with X-axis as data size and Y-axis as a performance measure. You can take inspiration from the "Data hunger of the model" part in https://peerj.com/articles/cs-559/ see Figures 2, 3 & 4.

---

**TASK 4: Report and Presentation (4-6 pages in ACL style plus references)**

This task will only be marked if you have completed Tasks 1, 2, and 3.

Finally, you will write a report documenting what you did in previous Tasks and comparing and contrasting your approach. You should explain why you decided on the algorithms and features you used and how this compares to state-of-the-art. You should discuss the performance of your approach and reflect on what you have learned. **You must include 5 interesting and diverse examples by comparing different models' outputs and explain why you think it is interesting**. **Note that you must provide 5 examples from test set fot both Task 2 & 3.** See Table 7 for inspiration at https://peerj.com/articles/cs-559/, you can't use these examples. Your report must be **4-6 pages** excluding references in the ACL format.

You will also prepare a presentation and use zoom to record the presentation for **8-10min** (not more than 10min). You must use the university's zoom, save your presentation on the university cloud, and provide a shareable link to the recorded video.

**Note that all writing and presentation must only be done in Task 4 and will have equal weightage.**

---

*Tasks and Submission Pointers*

- In Task 1, you will select 2 methods, write a short summary, and explain why you think that method is most suitable for the offensive language and OLID dataset. **Note that you can't select exactly the same naïve classifiers discussed in the lectures and labs.** You should have some modification to the pipeline.
- In Task 2, you will **devise and train your Offensive speech classifier using a suitable set of features extracted from the given dataset based on the** methods identified in Task 1. You can use any tools you like for extracting the features, e.g., any of the tools you have come across in the labs, such as NLTK, ScikitLearn, Pytorch, etc. You can also use codes from the web with proper citations, **but can't use your classmates code**. **Please note that simply rerunning the code is not sufficient. You need to provide the justification for why you selected any method and how that method works.**
- In Task 3, you will train your selected models on varying data sizes. You will divide **only the training** dataset into 4 equal subsets **and save it in the GDrive**. That is 100% dataset will be divided into sizes of [25%, 50%, 75%, and 100%] such that all previous data points are included in the next one. The division should be such that it maintains both class distributions of the original dataset. You should have a look at the train_test method with stratify. In practice, use train_test to recursively divide data into 4 splits, each with 25% data, say call them train_1, train_2, train_3, and train_4. Now 25% train = train_1, 50% train = train_1 + train_2, 75% train = train_1 + train_2 + train_3, and 100% train = train_1 + train_2 + train_3 + train_4. **This is a very important step, you need to get this right**. Remember, due to different seeds it is highly unlikely that two students will have same data splits.
- Your code and report must follow the structure provided. **You can't change the report's section/sub-section names and order. Your code must be properly commented on and follow consistent indentation.** You show used Google Colab's text and code cell effectively so that anyone can get an overview of the code. **Your code should always print at different places, like showing data statistics, data samples, model training progress (epoch train/valid loss for Neural Network), train/valid/test performance, and displaying the plots inline**.
- In Task 4, you will also **record the presentation** explaining your report and findings clearly and concisely. Your presentation should be **8-10 min**. **You will record yourself presenting using Zoom. Both your slides and face must be visible during the presentation**. Please ensure you are presentable in a quiet place with a good microphone and an internet connection for saving the recording to the zoom cloud. **You must ensure that anyone with the shared link is correct video is playable on the web, and it is not deleted**. A sample guide for the zoom recording. You have to submit the same presentation slide in pdf format.
- In Task 4, you will **write a summary scientific report** explaining what you did in previous Tasks together with some motivation (why you did it) and reflection (which alternatives did you consider, lessons learned, etc.). You need to comment on why and how one method is better compared to the other. You must comment on which method is better when less amount of data is available and why. You will compare and contrast your results with what you found and discuss and reflect on results in a

scientifically sound manner. We expect this to be between **4-6 pages** using ACL style plus references, which do not count toward the page limit.

- **Remember, you have to select best model based on the validation set. For a model, you must test only once using the testing dataset.**

### Organisation and content of the report shall follow a scientific paper

As a template for your report, please use the ACL style for conferences, preferably using LaTeX. There is even an Overleaf version of it available. Please check the report format provided.

You can find the template here: https://github.com/acl-org/acl-style-files

**Important Note: It is mandatory to use the ACL style to format the results for comparability of the different reports being submitted.**

## MARKING BREAKDOWN (out of 100%)

**Task 1. Model Selection (20%)**

- Summary of 2 selected Models **(up to 10%)**
- Critical discussion and justification of model selection **(up to 10%)**

**Task 2. Design and implementation of classifiers (30%) - Task 1 required**

- Training of classifier and features using selected models **(up to 10%)**
- High-quality code including comments and **printing required measures etc** **(up to 10%)**
- Achieving state-of-the-art performance and **examples** **discussion (up to 10%)**

**Task 3. Data Size Effect (20%) - Task 1 & 2 required**

- Training of classifiers on different data size **(up to 10%)**
- Critical Discussion of performance**/examples** with respect to data size **(up to 10%)**

**Task 4. Report and Presentation (30%) - Task 1, 2 & 3 required**

- Discussion of work carried out **(up to 10%)**
- Lessons learned **(up to 10%)**
- Material submitted in appropriate format **(up to 10%)**