Radboud University

# Authorship Attribution

**Assignment 44-AA**

**Text and Multimedia Mining**
**LET-REMA-LCEX06-2023-2024**

This is a group assignment: you will build a simple program for authorship attribution and write a report.

After this assignment:

- You can design a simple program to attribute authorship.
- You learn how to extract features from text.
- You understand the importance of feature selection.
- You are able to analyze the informativeness of individual features.
- You are able to analyze strong and weak points of an algorithm, given certain features.
- You can outline a full experiment: choose the appropriate method, the experimental setup, evaluate results and reflect on outcomes.

## Dataset

The data for this assignment was sourced from the PAN2020 dataset[1]. The dataset was designed for text analysis tasks such as author profiling and authorship attribution and originally created for the PAN 2020 Authorship Verification competition (part of the CLEF Conference). For this assignment, we are using a sample from the 'small' versions of the datasets. The dataset contains fanfiction texts by 20 different authors. We provide you with a training set, development set and test set in CSV format. Each line in these datasets contains a unique snippet number ID, the textual content of the snippet and lastly the numerical Author ID code. Each of these three sets contains document snippets written by the 20 authors. The task is to predict which author of the 20 authors wrote each snippet in the test set.

Fanfiction can often contain explicit content. The TA's have cleaned the dataset, but it is possible that there is still explicit and inappropriate content in the texts. Please be aware if you are going to read the texts (which you don't need to do).

[1] https://pan.webis.de/clef20/pan20-web/author-identification.html

## The application

Build a simple program for authorship attribution, make sure that your program at least consists of:

1. Loading of the data.
2. Extracting at least 50 features. This might seem a lot, but in reality using over 200 features is not uncommon in text mining. *(Hint: features should be both lexical and syntactical. You might want to take a look at the Sockpuppet paper[2] [optional reading] and the authorship attribution lecture slides if you get stuck.)*
3. Develop and tune a classifier that predicts the author based on the text using the training and development set.
4. Evaluating the performance of your classifier. You should be able to get an F-score of at least 0.7[3].
5. Performing an ablation analysis in the same style as is shown in the example Figure 1 (Solorio et al., 2014, p. 1357), where you check the performance of the classifier when you leave one of the features out. Which feature is the most informative for your classifier? (You do *not* need to use the exact same features as shown in the figure, but use this style of visualization of your ablation analysis.)
6. As a final step, run your classifier on the unseen test set.

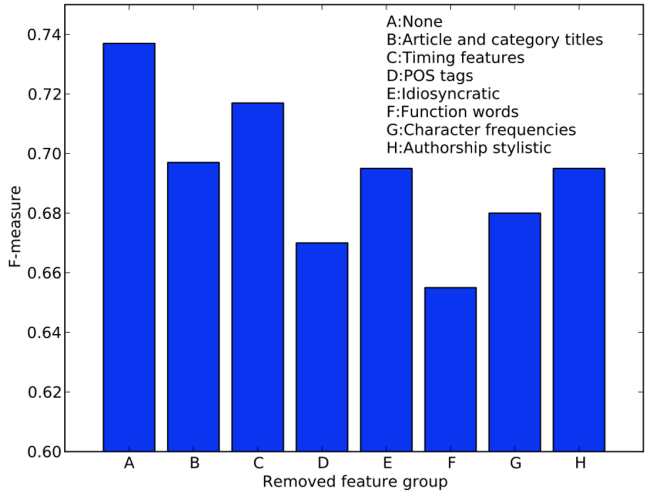


*Figure copied from: Sockpuppet Detection in Wikipedia: Solorio, T., Hasan, R., & Mizan, M. A Corpus of Real-World Deceptive Writing for Linking Identities, Proceedings LREC 2014*

[2] Solorio, T., Hasan, R., & Mizan, M. (2013). A Case Study of Sockpuppet Detection in Wikipedia http://www.cs.cornell.edu/~cristian/LASM_2013_files/LASM/pdf/LASM07.pdf.

[3] During test driving this assignment we achieved an average F1-score of 0.87 with 85 features on the dev dataset.

## Report

Provide a self-contained report of 3 pages in which you:

- Describe the design decisions you have made:
    - For each group (i.e., type) of features, provide a short description of the group, and 1-2 sentences of motivation for using that group (why did you choose it? What do you think it expresses?)
    - Evaluate the effect of individual feature groups (as mentioned above).
- Evaluate your classifier:
    - Evaluate the effectiveness of your classifier.
    - Mention whether it would be useful in your particular case to use feature selection?[4]
- Write a concise analysis of the results and reflect on the performance of the dev- and test sets.
- Make sure that your report answers these questions:
    a. What problems did you encounter? Do you see systematic patterns in the mistakes that are made by your best performing feature set (failure analysis)?
    b. What problems could you solve with 10 hours extra time?
    c. What problems do you think are real challenges in authorship attribution?

Feel free to add other issues you want to reflect upon.

**Submit your report as a PDF file to the respective folder, naming it "Txmm_A4-AA_groupnumber.pdf". Submit your programming code as a zip file to the other submission folder. If you use a notebook, please run the notebook before you hand in the assignment. Only one of the group members has to hand in the assignment on Brightspace.**

# Practical information

Note that this group assignment will be graded with a Pass/Fail system. We only grade the report. We will only look at your code to support grading your analysis if something is unclear.

You are free to use any programming language that your prefer but the TA's can only support Python related issues.

Whenever you have any questions or if you have trouble getting started with this assignment, feel free to ask us in the open lunch hours on Mondays! You can also contact the TAs, Ellen Jansen and Heleen Visserman, through discord or by sending a mail to Heleen (heleen.visserman@ru.nl).

We would appreciate it if you do not contact us via WhatsApp for non-urgent matters, so we can keep our TA work and private life (somewhat) separate.

[4] Examples of feature selection in scikit-learn: http://scikit-learn.org/stable/modules/feature_selection.html