Replication materials for "Detecting Group Mentions in Political Rhetoric: A Supervised Learning Approach"

Hauke Licht* Ronja Sczepanski†

2024-11-18

Contents

Overview **1**

Materials **2**

`data/` . . . 2

`code/` . . . 3

`results/` . . . 6

`paper/` . . . 6

Computer setup **6**

Software requirements . . . 6

Hardware requirements . . . 6

Overview

This document accompanies the replication materials for the **research article**

> Licht, H., & Sczepanski, R. (forthcoming). Who are they talking about? Detecting mentions of social groups in political texts with supervised learning. *British Journal of Political Science*.

The replication materials are available in the British Journal of Political Science (BJPolS) Harvard Dataverse[1] at: https://doi.org/10.7910/DVN/QCOQ0T

Please address any **questions** you have about the replication materials at the corresponding author Hauke Licht (hauke.licht@ubik.ac.at).

*University of Innsbruck. Corresponding author. Contact at hauke.licht@uibk.ac.at
†Scienes Po Paris
[1]https://dataverse.harvard.edu/dataverse/BJPolS

Materials

Important To display the materials in their original file structure (instead of a flat list of files), select the **tree view** in the data set files viewer:

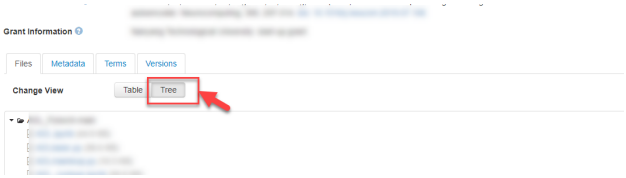

Figure 1: Enable data set files *tree view*

The replication materials are organized in the following folders:

- `code/`: Contains the code for all experiments and analyses.
- `data/`: Contains the data used all experiments and analyses.
- `results/`: Contains the results of all experiments and analyses, including fine-tuned models.
- `paper/`: Contains the figures and tables reported in the paper and the Supplementary Materials

See below for a detailed breakdown of the contents of these folders.

`data/`

The data used in this study is available in the `data/` folder. The data is organized in the following subfolders:

- `corpora/`: Contains the raw text corpora used in the study.
- `exdata/`: Contains data taken from external sources, including
  - the *Comparative Agendas Project* (`cap/`),
  - the *Comparative Manifestos Project* (`cmp/`),
  - the Dolinksy-Huber-Horne dictionary (`dhh_dictionary/`), and
  - Thau (2019) (`thau2019`).
- `annotation/`: Contains the human annotations and aggregated labels for all corpora used in our analyses and experiments.
  - `annotations/`: Contains the samples of sentences distributed for group mention annotation and the resulting annotations. Each subfolder corresponds to one annotation "job" where prefixes indicate the source corpus and jobs ending with '-review' are those where we reviewed and disambiguated annotators disagreements
  - `parsed/`: Contains the parsed coder-level annotations in a structured format
  - `labeled/`: Contains the annotation aggregation model-based labels induced from human annotators' group mention annotations
  - `exdata/`: Contains the mention-level annotations reconstructed from Thau (2019)
- `labeled/`: Contains the model-based labels at the sentence and mention levels of measurement we generated with our group mention token classifier (as well as those for the CAP policy topic classifier)
- `validation/dhh_dictionary/`: Contains the intermediat data for evaluations and validations conducted based on the Dolinksy-Huber-Horne dictionary.

code/

The code used in this study is available in the code/ folder. The code is organized in the following subfolders:

- 01-preprocessing/: Contains the code for preparing and processing the text corpora and external data sources.
- 02-annotation/: Contains the code for parsing and aggregating the group mention annotations we have collected.
- 03-finetuning/: Contains the code for fine-tuning the Transformer-based group mention token classifier and running all related experiments. In addition, it contains the code for fine-tuning and inference of the CAP policy topic classifier we use in Application I. *Note:* Script _run.slurm in this folder can be used to run all scripts in the desired order using SLURM.
- 04-descriptives/: R scripts to generate descriptive statistics of our corpora, annotations, labeled data, and the Thau (2019) data.
- 05-evaluation/: Contains the code for reporting the evaluation results of all our supervised group mention detection classifier experiments
- 06-validation/: Contains the code for validating the group mention token classifier at the document-level against Thau's. Moreover, subfolder ddh_dictonary contains all scripts to conduct our analysis of the effectiveness of the Dolinksy-Huber-Horne dictionary in group mention detection.
- 07-analyses/: Contains the R code for running the analyses presented as Application I and II in the paper
- r_utils/: Contains utility functions used in the R scripts
- utils/: Contains all custom modules and utility functions used in the Python scripts

Description of all scripts and their running order

- 01-preprocessing/
  - download_and_clean_cap_data.py: downloads and cleans Comparative Agendas Project (CAP) data
  - thau2019/
    - match_thau_group_appeals_to_manifesto_texts.R: Match group appeal annotations in Thau (2019) data to manifesto sentence texts
    - parse_thau2019_matched_spans_to_token_labels.ipynb: Parse matched group appeal annotations to token-level labels
- 02-annotation/
  - parse_uk-manifesto_annotations.ipynb: Parse UK manifesto annotations
  - parse_de-manifesto_annotations.ipynb: Parse German manifesto annotations
  - parse_uk-commons_annotations.ipynb: Parse UK House of Commons annotations
  - induce_labels_uk-manifestos.ipynb: Induce token-level labels from UK manifesto annotations using bayesian sequence aggregation model
  - induce_labels_de-manifestos.ipynb: Induce token-level labels from German manifesto annotations using bayesian sequence aggregation model
  - induce_labels_uk-commons.ipynb: Induce token-level labels from UK House of Commons annotations using bayesian sequence aggregation model
- 03-finetuning/
  - _download_transformer_model_checkpoints.py: downloads transformer model checkpoints
  - _run_experiments.slurm: runs the following files:
    - run_model_comparison_experiment.py: Run hyperparameter search for different transformer models
    - run_5x5_crossval_experiment.py: Run 5x5 cross-validation experiment with DeBERTa model on UK manifesto annotations

    - * `run_training-size_experiment.py`: Run training-size experiment with RoBERTa model on UK manifesto annotations
    - * `run_cross-party-transfer_experiment.py`: Run cross-party transfer experiment with DeBERTa model on UK manifesto annotations
    - * `run_cross-domain-transfer_experiment.py`: Run cross-domain transfer experiment with DeBERTa model on UK manifesto and House of Commons annotations
    - * `run_cross-lingual-transfer_experiment.py`: Run cross-lingual transfer experiment with XLM-RoBERTa model on UK and German manifesto annotations
  - `_finetune_token_classifier.slurm`: runs the following files:
    - * `finetune_token_classifier.py`: Finetune RoBERTa token classifier on UK manifesto annotations
    - * `inference_token_classifier.py`: Applies finetuned token classifier to unlabeled UK manifesto sentences
  - `_finetune_cap_topic_classifier.slurm`: runs the following files:
    - * `finetune_cap_classifier.py`: Finetune RoBERTa sentence classifier on UK manifesto CAP labels
    - * `inference_cap_classifier.py`: Applies finetuned CAP classifier to unlabeled UK manifesto sentences
- `04-descriptives/`
  - `describe_dataset.R`: Describe the UK manifesto, German manifesto, and UK House of Commons datasets
  - `describe_annotations.R`: Describe the UK manifesto, German manifesto, and UK House of Commons annotations
  - `describe_labeled_data.R`: Describe the UK manifesto, German manifesto, and UK House of Commons annotation aggregation model-based labels
  - `describe_thau2019_data.R`: Describe the Thau (2019) group appeal annotations
- `05-evaluation/`
  - `report_classifier_performances.R`: Report token classifier performances
  - `evaluate_token_classifier_thau2019-manifestos.py`: Evaluate the finetuned RoBERTa token classifier in the Thau (2019) annotations
  - `evaluate_uk-manifesto_token_classifier_error_analysis.py`: Perform error analysis of the finetuned RoBERTa token classifier in the test set
- `06-validation/`
  - `validate_uk_manifesto_document-level_measurements.R`: Validate manifestos-level measurements in UK manifesto obtained with our finetuned RoBERTa token classifier, Thau's human annotators, and the Dolinsky-Huber-Horne (DHH) dictionary
  - `dhh_dictionary/`
    - * `estimate_ddh_dictionary_expansion_effort.ipynb`: Estimate the effort required to extend the keyword list in the the DHH dictionary using human-in-the-loop word embedding-based dictionary expansion
    - * `apply_dhh_dictionary_uk-manifestos.py`: Apply the Dolinsky-Huber-Horne (DHH) dictionary to the UK manifesto sentences
    - * `apply_dhh_dictionary_thau2019-manifestos.py`: Apply the Dolinsky-Huber-Horne (DHH) dictionary to the Thau (2019) manifestos annotations
    - * `evaluate_dhh_dictionary_thau2019-manifestos.py`: Evaluate the Dolinsky-Huber-Horne (DHH) dictionary in the Thau (2019) annotations
    - * `evaluate_dhh_dictionary_uk-manifestos.py`: Evaluate the Dolinsky-Huber-Horne (DHH) dictionary in the UK manifesto sentences
    - * `evaluate_expanded_dhh_dictionaries_thau2019-manifestos.py`: Evaluate the Dolinsky-Huber-Horne (DHH) dictionary in the Thau (2019) annotations after expanding the dictionary with human-in-the-loop word embedding-based dictionary expansion

    * `report_dhh_dictionary_performances.R`: Report the performance of the Dolinsky-Huber-Horne (DHH) dictionary in the UK manifesto sentences and Thau (2019) annotations (before and after dictionary expansion)

- `07-analyses/`
  - `01-analyze_uk-manifestos_cap.R`: Perform analysis reported in Application I
  - `02-analyze_group_affect_relation.R`: Perform analysis reported in Application II

Where to find the code for generating the figures and tables in the main paper and supplementary materials

Paper

- Figure 1: `code/06-validation/validate_uk_manifesto_document-level_measurements.R`
- Table 2: `code/05-evaluation/report_classifier_performances.R`
- Figure 3: `code/06-validation/validate_uk_manifesto_document-level_measurements.R`
- Figure 5: `code/07-analyses/01-analyze_uk-manifestos_cap.R`
- Figure 6: `code/07-analyses/01-analyze_uk-manifestos_cap.R`
- Figure 7: `code/07-analyses/02-analyze_group_affect_relation.R`

Supplementary Materials

- Table A1: `code/04-descriptives/describe_dataset.R`
- Table A2: `code/04-descriptives/describe_dataset.R`
- Table A3: `code/04-descriptives/describe_dataset.R`
- Table B1: `code/04-descriptives/describe_annotations.R`
- Table B2: `code/04-descriptives/describe_annotations.R`
- Table B4: `code/04-descriptives/describe_annotations.R`
- Table B5: `code/04-descriptives/describe_annotations.R`
- Table B6: `code/04-descriptives/describe_annotations.R`
- Table B7: `code/04-descriptives/describe_labeled_data.R`
- Table B8: `code/04-descriptives/describe_labeled_data.R`
- Table E1: `code/05-evaluation/report_classifier_performances.R`
- Table E2: `code/05-evaluation/report_classifier_performances.R`
- Table E3: `code/05-evaluation/report_classifier_performances.R`
- Figure E1: `code/05-evaluation/report_classifier_performances.R`
- Table F1: `code/04-descriptives/describe_thau2019_data.R`
- Table F2: `code/04-descriptives/describe_thau2019_data.R`
- Table F3: `code/05-evaluation/report_classifier_performances.R`
- Figure F1: `code/05-evaluation/report_classifier_performances.R`
- Figure F2: `code/07-analyses/01-analyze_uk-manifestos_cap.R`
- Table G1: `code/06-validation/dhh_dictionary/report_dhh_dictionary_performances.R`
- Table G2: `code/06-validation/dhh_dictionary/report_dhh_dictionary_performances.R`
- Figure G1: `code/06-validation/validate_uk_manifesto_document-level_measurements.R`
- Table G3: `code/06-validation/dhh_dictionary/report_dhh_dictionary_performances.R`
- Table G4: `code/06-validation/dhh_dictionary/report_dhh_dictionary_performances.R`
- Figure G2: `code/06-validation/validate_uk_manifesto_document-level_measurements.R`
- Figure G3: `code/06-validation/dhh_dictionary/report_dhh_dictionary_performances.R`
- Table H1: `code/07-analyses/02-analyze_group_affect_relation.R`
- Table H2: `code/07-analyses/02-analyze_group_affect_relation.R`
- Table H3: `code/07-analyses/02-analyze_group_affect_relation.R`
- Table H4: `code/07-analyses/02-analyze_group_affect_relation.R`
- Table H5: `code/07-analyses/02-analyze_group_affect_relation.R`

`results/`

- `classifiers/`: Fine-tuned group mention detection and CAP policy topic classifiers underpinning our ananylses.
- `experiments/`: Results from classifier fine-tuning experiments reported in paper section 4 and Supplementary Material E.
- 'validation/": Results from the evaluating the Dolinksy-Huber-Horne group mention dictionary.

`paper/`

- `figures/`: All figures reported in the paper and Supplementary Materials
- `tables/`: All tables reported in the paper and Supplementary Materials

Computer setup

Software requirements

The code was written in R and Python.

The **python version** used in the analysis is 3.10. The **python environment** is managed by `conda`. The required python libraries and versions are documented in the file `python_requirements.txt`. To replicate the python setup, run the following command in the terminal:

```
conda create -y --name group_mention_detection python=3.10 pip
conda activate group_mention_detection
pip install -r python_requirements.txt
```

The **R version** used in the analysis is 4.2.2. The **R environment** is managed by `renv`. The required R packages and versions are documented in the file `r_requirements.txt`: To replicate the python setup, run the following command in the terminal in R:

```
devtools::install_version("renv", version = "0.15.5")
library(renv)
init(bare = TRUE)
pkgs <- readLines("r_requirements.txt")
install(pkgs)
```

Hardware requirements

We ran all analysis requiring GPU computing (model fine-tuning and inference) on the ETH Zurich's HPC cluster EULER using SLURM and a single NVIDIA RTX 4090 with 24GB GPU memory.

All other code was run on a MacBook Pro 2021 with an Apple M1 Chip and 32 GB RAM.