Replication data and code for "Measuring and understanding parties' anti-elite strategies"

Hauke Licht

2023-11-03

This file describes the replication data and code for the *Journal of Politics* paper "Measuring and understanding parties' anti-elite strategies" by Hauke Licht, Tarik Abou-Chadi, Pablo Barberá, and Whitney Hua.

Contact Hauke Licht (hauke.licht@wiso.uni-koeln.de) with questions about the code and data.

Folder structure and files

```
.
|_ 'renv.lock'       <== lock file to manage R dependencies
|_ 'replicate.sh'    <== bash script replicating all figures and tables
|_ code/
   |_ 01-preproc/    <== scripts for data pre-processing
   |_ 02-annotation/ <== scripts for data sampling and annotation aggregation
   |_ 03-classify/   <== scripts for final classifier training, evaluation, and application
   |_ 04-postproc/   <== scripts for creating party--time unit measurements and dataset
   |_ 05-validate/   <== scripts for conducting validation assessments
   |_ 06-report/     <== scripts for reporting results
|_ data/
   |_ exdata/        <== external data
   |_ fits/          <== fitted models and classifiers
   |_ input/         <== inputs to measurement process
   |_ intermediate/  <== intermediate data products and objects
      |_ embeddings/ <== embeddings of tweets' texts
      |_ samples/    <== tweets sample for 1st and 2nd round of annotation
      |_ codings/    <== codings from 1st and 2nd round of annotation
      |_ labelings/  <== model-induced labelings from from 1st- and 2nd-round codings
      |_ training/   <== training data and configurations
   |_ output/        <== outputs of measurement process
|_ documentation/    <== the supporting materials and data appendix
|_ results/
   |_ figures/       <== all figures
   |_ tables/        <== all tables
```

Replication instructions

You can reproduce the results reported in the paper, supporting materials, and data appendix by running (on MacOS/Linux): `bash ./replicate.sh > run.log`

This script first sets up the R environment and then runs the R scripts in folder 'code/06-report/' for generating all figures and tables in the paper, supporting materials, the data appendix:

- '01-compile_paper_objects.R': creates all R objects used to create figures and tables reported in the paper, supporting materials, and data appendix (regressions are run in section "8. DOWNSTREAM ANALYSES", lines 1049ff.)
- '02-report_results_in_main_paper.R': creates all figures and tables reported in the paper
- '03-report_results_in_supporting_materials.R': creates all figures and tables reported in the supporting materials
- '04-report_results_in_data_appendix.R': creates all figures and tables reported in the data appendix

Note that you need to your system needs to meet the R requirements outlined in section "Requirements" below.

Party–quarter level dataset The dataset with the party–quarter level anti-elite strategy estimates used in our regression analyses is 'data/output/party_antielite_strategies_quarterly_data.rds'. The file 'data/output/party_antielite_strategies_quarterly_data-codebook.tsv' is the corresponding codebook that describes what variables indicate.

Further details For any further details, please consult the Excel spreadsheet 'code/scripts_overview.xlsx'. It tabulates the R scripts respectively python Jupyter notebooks that implement our measurement process and analyses in chronological.

IMPORTANT: No tweet texts and metadata shared! Due to Twitter's (now X) terms of service, we are not allowed to share tweets' texts and metadata (e.g., number of quotes and likes, etc.).

Tweets' text and metadata has thus been removed from the following data files before sharing:

- 'data/input/all_party_tweets.rds'
- 'data/input/all_tweets_classified_political.rds'
- 'data/input/all_tweets_texts.csv'
- 'data/input/parl_party_tweets.rds'
- 'data/input/parl_party_tweets_translated_2020-04-14.rds'
- 'data/input/parl_party_tweets_features_2020-04-24.rds'
- 'data/fits/political_en_tweets_clustered.rds'
- 'data/output/parl_party_tweets_labeled.rds'

This implies that some of the scripts ingesting these data files will not run. For example, one can't LASER-embed tweets without their texts. Please contact hauke.licht@wiso.uni-koeln.de in case of questions.

Requirements

You require R ($\geq$4.1.0) to run the main analysis in our paper.

However, as indicated in file 'code/scripts_overview.xlsx' (column "language"), there are a two steps in our measurement procedure that require an installation of Python ($\geq$3.8) and conda ($\geq$23.5.0) and appropriate conda environments (see python subsecton below):

1. embedding tweets with the pre-trained LASER model
2. machine-learning classifier training on the final set of labeled tweets
3. transformer finetuning on the final set of labeled tweets

Moreover, if you want to reproduce the Transformer training and inference/predictions steps, you'll require access to a Nvidia GPU, have `cuda` (≥11.0) installed (e.g. through *Google Colab*), and you require `rust` to run the sentence-piece tokenizer.

All R code and all python code requiring *no* GPU has been run on a Apple MacBook M1 Pro with MacOS 13.5.2 "Ventura" (22G91). All python code requiring GPU processors has been run on Google Colab with a (Nvidia) Tesla T4 GPU.

R

R dependencies are managed with `renv`. R script 'code/setup_renv.R' takes care of the setup process.

Moreover, all required R packages are listed in file 'code/r_requirements.txt'. In this file, lines that start with '#' are comments. All other lines list one one package per line in the format `<PACKAGE NAME>@<PACKAGE VERSION>`. Packages with a slash ('/') in the package name are packages that need to be installed from GitHub.

Python

Python dependencies are managed with `conda`. In total, we use three separate conda environments with different dependencies and package versions:

1. 'laser_embedding': conda environment used for embedding tweets' text with a pre-trained LASER multilingual embedding model (required for stratified sampling and classifier training in step 1 of our two-step selective sampling procedure and because the `laserembeddings` package depends on old(er) versions of `torch` and `subword-nmt` compared to the package version used in the 'machine_learning' conda environment)
2. 'machine_learning': conda environment used for cross-validation fold sampling and training of bag-of-words and multilingual sentence embedding-based classifiers at the end of step 2 of our two-step measurement procedure
3. 'xlmt': conda environment used for finetuning and applying our (transformer-based) XLM-T classifier at the end of step 2 of our two-step measurement procedure

The bash/shell script 'code/setup_conda_envs.sh' takes care of creating the conda environments we use and installing the required packages.

The list below indicates which environments to use for which Python Jupyter notebooks:

- 'laser_embedding'
  - code/01-preproc/01-create_tweet_laser_embeddings.ipynb
- 'machine_learning'
  - code/03-classify/02-create_cv_folds.ipynb
  - code/03-classify/03a-train_bow_classifiers.ipynb
  - code/03-classify/03b-train_mse_classifiers.ipynb
- 'xlmt'
  - code/03-classify/03c-finetune_xlmt_classifier.ipynb
  - code/03-classify/05-label_tweets_using_xlmt_classifier.ipynb