Campaign Communication and Legislative Leadership

Stefan Müller and Naofumi Fujimura

Political Science Research and Methods

This README file contains information on all R and Python scripts used to produce tables and plots reported in the main paper and Supporting Information.

The R package versions required to reproduce all results are listed at the start of each script. If the code does not run, package updates might have resulted in these errors. The authors ran all code scripts successfully on 15 February 2024. To reproduce the code, you can install the R version listed in the R script or use the *groundhog* package.[1] In order to run the R scripts, we recommend creating RProject[2] in the folder containing all data and scripts. Alternatively, you can use the *here*[3] package or set the working directory with the `setwd()` function. We strongly recommend downloading files in the original format from the Dataverse (select "Download" -> "Original File Format").

If you want to reproduce the entire analysis, just run the script **00_reproduce_analysis.R**. Below, we describe each file, the required input data, and outputs (temporary data, figures, tables).

00_reproduce_analysis.R

Running this script will execute all scripts listed below in the correct order. The script lists all packages (and CRAN versions) used in one or more R scripts in the folder. Please make sure to install these packages before rerunning our analysis. If you have the *rmarkdown* package installed, the script will also create html log files of all scripts. These html log files are included in the Dataverse folder when the entire code was last run successfully.

01_validate_statement_segmentation.R

Validation of the statement segmentation by comparing human codings of 60 manifestos to automated segmentation of the same set of manifestos.

Input:

- `data_corpus_japmanifestos.rds`: corpus containing all available candidate manifestos
- `data_manifestos_manual_segmentation.rds`: corpus with 60 manifestos that has been manually segmented into statements by human coders

Output:

- Figure A04: fig_a04.pdf

---

[1] Change `library(name_of_package)` to `groundhog::groundhog.library(name_of_package, date = "2024-01-31")` after installing groundhog: `install.packages("groundhog")`. More details are provided in the R scripts and at https://groundhogr.com/using/

[2] https://support.posit.co/hc/en-us/articles/200526207-Using-RStudio-Projects

[3] https://here.r-lib.org

02_prepare_bert_data.R

Segment manifesto corpus to the level of statements, remove empty statements, and store data as CSV file for statement-level predictions in Python

Input:

- `data_corpus_japmanifestos.rds`: corpus containing all available candidate manifestos

Output:

- `data_sentences_all.csv`: corpus of all valid statements to be used for BERT classification

03a_finetune_transformer.ipynb

Fine-tune a transformer model for classification of policy areas in Japanese candidate manifestos. Note: running this script is optional since it takes over 3.5 hours on a Mac Studio M2 Ultra (192 GB RAM) to fine-tune the model. It could take much longer on less powerful machines. You can proceed with 04_clean_transformers_output.R if you prefer not to run this Python script.

Input:

- `data_sentences_train.csv`: training set of 2,000 statements
- `data_sentences_eval.csv`: eval set of 500 statements
- `data_sentences_test.csv`: test set of 485 statements

Output:

- Fine-tuned BERT model (stored in `bert_manifestos` folder), consisting of three files: `config.json`; `model.safetensors`; `training_args.bin`
- `data_test_predicted_bert.csv`: spreadsheet containing the BERT predictions for the held-out test set

03b_predict_all_sentences.ipynb

Apply the fine-tuned transformer model to the entire corpus. Running this script could take several hours, depending on the computing power of the user's machine. You can proceed with 04_clean_transformers_output.R if you prefer not to run this Python script.

Input:

- `data_sentences_all.csv`: corpus of all valid statements
- Files in `bert_manifestos` folder: fine-tuned BERT model

Output:

- `data_predicted_full.csv`: CSV file with all variables from data_sentences_all.csv and BERT predictions

04_clean_transformers_output.R

Clean output of transformer predictions and analyse class distributions

Input:

- `data_predicted_full.csv`: CSV file with all variables from data_sentences_all.csv and BERT predictions
- `data_handcoded_sentences.csv`: set of all hand-coded statements

Output:

- `data_policy_areas_manifesto_bert.rds`: data frame measuring manifesto-level salience of policy areas
- Figure A06 (fig_a06.pdf)
- Table A05 (tab_a05.html)

05_performance_all_classifiers.R

Compare BERT predictions to human codings; compare fine-tuned BERT model with SVM and Naïve Bayes (both bag-of-words models)

Input:

- `data_test_predicted_bert.csv`: test set of 485 classified sentences
- `data_sentences_train.csv`: training set of 2,000 statements
- `data_sentences_eval.csv`: eval set of 500 statements
- `data_sentences_test.csv`: test set of 485 statements

Output:

- Figure 01 (fig_01.pdf)
- Table A03 (tab_a03.html)

06_merge_sources.R

Merge all sources: manifesto-level salience; legislative posts; Reed and Smith (2018) dataset containing detailed information on each candidate-election observation

Input:

- `data_positions_all.rds`: data containing legislative posts
- `Reed-Smith-JHRED-CANDIDATES.dta`: *The Reed-Smith Japanese House of Representatives Elections Dataset*. Version: 11 July 2018. URL: https://doi.org/10.7910/DVN/QFEPXD
- `data_policy_areas_manifesto_bert.rds`: data frame measuring manifesto-level salience of policy areas
- `data_importance_portfolios.csv`: CSV file with perceived importance of portfolios based on expert surveys

Output:

- `data_candidates_available.csv`: CSV file containing the percentage of available manifestos for each election (mentioned in paper)

- `data_manifestos_included.csv`: list of manifestos included in analysis
- `data_analysis_bert.rds`: merged dataset

07_keyness_and_wordfish.R

Keyness analysis to identify predictive terms of each policy area; and Wordfish scaling of all manifestos (based on the subsample of manifestos included in the analysis)

Input:

- `data_predicted_full.csv`: CSV file with statement-level BERT predictions
- `data_manifestos_included.csv`: list of manifestos included in analysis
- `data_keyness_jp_translated.csv`: manually translated keyness terms (first created in script, then translated, then loaded again)

Output:

- Figure A05 (fig_a05.pdf)
- Table A04 (tab_a04.html)
- `data_regression_bert_wf.rds`: dataset used for analysis

08_analysis.R

Conduct all remaining analyses reported in the paper and Supporting Information

Input:

- `data_regression_bert_wf.rds`: dataset used for analysis
- `Reed-Smith-JHRED-CANDIDATES.dta`: *The Reed-Smith Japanese House of Representatives Elections Dataset*. Version: 11 July 2018. URL: https://doi.org/10.7910/DVN/QFEPXD
- `data_candidates_available.csv`: CSV file containing the percentage of available manifestos for each election (mentioned in paper)
- `data_predicted_full.csv`: CSV file with all variables from data_sentences_all.csv and BERT predictions

Output:

- Figure 02 (fig_02.pdf)
- Figure 03 (fig_03.pdf)
- Figure 04 (fig_04.pdf)
- Figure 05 (fig_05.pdf)
- Figure 06 (fig_06.pdf)
- Table 01 (tab_01.html)
- Table 02 (tab_02.html)
- Figure A01 (fig_a01.pdf)
- Figure A02 (fig_a02.pdf)
- Figure A08 (fig_a08.pdf)
- Table A06 (tab_a06.html)
- Table A07 (tab_a07.docx)
- Table A08 (tab_a08.html)

- Figure A09 (fig_a09.pdf)
- Figure A10 (fig_a10.pdf)
- Figure A11 (fig_a11.pdf)
- Table A09 (tab_a09.html)
- Figure A12 (fig_a12.pdf)
- Table A10 (tab_a10.html)
- Figure A13 (fig_a13.pdf)
- Table A11 (tab_a11.docx)
- Figure A14 (fig_a14.pdf)
- Table A12 (tab_a12.html)
- Table A13 (tab_a13.html)
- Figure A15 (fig_a15.pdf)

The following figures and tables do not require any code:

- Table A01: aggregated statistics from election study reports
- Table A02: list of posts in each policy area
- Figure A03: flowchart
- Figure A07: three annotated manifestos