Guide to the Replication of

Fabio Franchino, Marta Migliorati, Giovanni Pagano, Valerio Vignoli Franchino, Fabio. "Identifying Delegation and Constraints in Legislative Texts: A Computational Method Applied to the European Union". *European Union Politics*

Table of Contents

Datasets and Coding Manuals 2
Replication of the Results in the Article 2
Figure 1: Number of laws and legal sentences 2
Figure 2: Trends in authority delegation and constraint across executive actors 2
Replication of the Results in the Online Appendix 3
Table A2: Performance metrics: NER model for actors 3
Table A3: Performance metrics: NER model for verbs 3
Table A5: Performance metrics: rule-based application vs. authors classification 3
Table A6: Performance metrics: rule-based application vs. annotators classification 3
Table A7: Benchmarking: BERT model vs. authors classification 3
Table A8: Benchmarking: RoBERTa model vs. authors classification 3
Table A9: Benchmarking: DistilBERT model vs. authors classification 3
Table A10: Benchmarking: EurLexBERT model vs. authors classification 3
Comparison with Manually Annotated Datasets 4
Intercoder Reliability 4
Figure A1: Trends in delegation and constraint of national authorities in internal market policies 4
Figure A2: Mean differences in discretion ratio between directives and regulations 4
Classification Pipeline Replication Code Overview 5
Step 1 – Preprocessing 5
Step 2 – NER model training and evaluation 5
Step 3 – Full classification of legal sentences 5
Step 4 – Transformer-based classification 6
Environment 6
References 6

Datasets and Coding Manuals

The replication material includes the following Stata datasets.

File name	Description
annotations.dta	Classification of 3,000 sentences of a) the rule-based application, b) the authors, and c) the two annotators
dataset.dta	Number of provisions assigned to the thirteen classes in 9,319 secondary measures (directives and regulations), and several ratio measures
dataset_disaggregated.dta	Classification of the sentences of the 9,319 secondary measures
intercoder_reliability.dta	Annotations of the authors and the two coders
manual_coded_MF.dta	Ratio measures from the manually annotated datasets of Franchino (2007) and Migliorati (2021)

The replication material also includes the full coding manual of the extraction rules and a summary.

PLEASE NOTE:

The files replicating the results in the Article, Tables A5 and A6, the comparison with manually annotated datasets, the intercoder reliability test, and Figures A1 and A2 are in the supplemental material folder.

The files replicating the machine learning analyses reported in Tables A2, A3, and A7 to A10 are in the NLP-named supplemental material folder.

Replication of the Results in the Article

Run the do file to replicate the results reported in Figures 1 and 2. We recommend using Stata version 18 or above. Make sure to download all the replication files and save them in the *same* folder, from which you will be able to replicate all the results. If datasets are modified by the commands, we suggest not saving the changes before exiting.

Figure 1: Number of laws and legal sentences
Figure 2: Trends in authority delegation and constraint across executive actors
Commands to reproduce Figure 1 and Figure 2.
Dataset: dataset.dta
Do file: Figures_1_2_trends.do
Output: Figure_1.gph, Figure_1.png, trendMS.gph, trendCA.gph, trendCOM.gph, trendAGE.gph, Figure_2.gph, Figure_2.gph

Replication of the Results in the Online Appendix

Run the do files to replicate the results of Tables A5 and A6, the comparison with manually annotated datasets, the intercoder reliability test, and Figures A1 and A2. We recommend using Stata version 18 or above. Make sure to download all the replication files and save them in the *same* folder, from which you will be able to replicate all the results. If datasets are modified by the commands, we suggest not saving the changes before exiting.

Please read the next section to replicate Tables A2, A3, and A7 to A10.

Table A2: Performance metrics: NER model for actors
Folder: NLP-named supplemental material
Files needed: config files, training and validation sets, in the
models_files\NER_validation\NER_institutions folder
Py file: scripts/02_script_train_eval_ner_institutions.py
Output: Table_A2.csv

Table A3: Performance metrics: NER model for verbs
Folder: NLP-named supplemental material
Files needed: config files, training and validation sets, in the
models_files\NER_validation\NER_verbs folder
Py file: scripts/03_script_train_eval_ner_verbs.py
Output: Table_A3.csv

Table A5: Performance metrics: rule-based application vs. authors classification
Datasets: annotations.dta
Do file: Table_A5_manualAU_model_analysis.do
Output: TableA5manualAU_model.doc

Table A6: Performance metrics: rule-based application vs. annotators classification
Datasets: annotations.dta
Do file: Table_A6_manualANU_model_analysis.do
Output: TableA6manualAN_model.doc

Table A7: Benchmarking: BERT model vs. authors classification
Table A8: Benchmarking: RoBERTa model vs. authors classification
Table A9: Benchmarking: DistilBERT model vs. authors classification
Table A10: Benchmarking: EurLexBERT model vs. authors classification
Folder: NLP-named supplemental material
Datasets: source_files/authors_annotation.csv
Py File: scripts/06_script_train_eval_transformers.py
Output: Table_A7.csv, Table_A8.csv, Table_A9.csv, Table A_10.csv

Comparison with Manually Annotated Datasets
Datasets: dataset.dta, manual_coded_MF.dta
Do file: Comparison_manually_annotated_datasets.do
Output: Correlation coefficient between hand-coded (HC) and computational national (member states and national competent authorithies) discretion ratios.
Correlation coefficient between hand-coded (HC) and computational discretion ratios of the European Commission.

Intercoder Reliability
Datasets: intercoder_reliability.dta
Do file: Intercoder_reliability_test.do
Output: Cohen's kappa coefficient

Figure A1: Trends in delegation and constraint of national authorities in internal market policies
Commands to reproduce Figure A1.
Dataset: dataset.dta
Do file: Figure_A1_trends_int_mkt.do
Output: Figure_A1.gph, Figure_A1.png

Figure A2: Mean differences in discretion ratio between directives and regulations
Commands to reproduce Figure A2.
Dataset: dataset.dta
Do file: Figure_A2_regs_dirs.do
Output: Figure_A2.gph, Figure_A2.png

Classification Pipeline Replication Code Overview

The Python component of the replication package reproduces the machine learning analyses reported in Tables A2, A3, and A7 to A10. It complements the Stata replication files. It includes all scripts required to preprocess legal texts, train Named Entity Recognition (NER) models, classify sentences according to the Extraction Rules, and evaluate transformer-based classifiers. All scripts should be executed from the project root after activating the *spacy36* conda environment specified in *environment.yml*.

Step 1 – Preprocessing
The script *scripts/01_script_preprocess_eurlex.py* loads all EurLex source files from the *source_files* directory, merges them, and applies text normalization and sentence segmentation functions defined in *replication_src/text_utils.py*. These functions remove preambles (e.g., "Whereas…"), trim start and end formulas, convert uppercase text to proper case, and divide sentences into smaller analytical chunks. The output is a JSONL file of processed sentences (*corpus_files/EurLex_sentences.jsonl*).

Step 2 – NER model training and evaluation
The script *scripts/04_script_train_ner_models.py* trains two spaCy models for Named Entity Recognition: one for institutional actors and one for verbs. Once trained, the models are stored in the *models_files/NER_institutions* and the *models_files/NER_verbs* folders, respectively. Their evaluation is handled by two scripts:

- *scripts/02_script_train_eval_ner_institutions.py*, which performs five-fold training and validation for institutional actors and produces the results in Table A2;
- *scripts/03_script_train_eval_ner_verbs.py*, which trains and evaluates the verbs model and produces Table A3.

Each script outputs the corresponding performance tables (precision, recall, F1) as CSV files in the *output_tables* directory.

Step 3 – Full classification of legal sentences
The main classification pipeline is implemented in *scripts/05_script_pipeline_main.py*. It loads the trained NER models and applies syntactic and rule-based extraction functions from *replication_src/eurlex_functions.py*. These functions identify the grammatical and semantic structure of each sentence to classify provisions as *delegating*, *soft obligation*, *constraining* [..] for each institutional actor (Member States, National Competent Authorities, Commission, Agencies).

Step 4 – Transformer-based classification
The script *scripts/06_script_train_eval_transformers.py* fine-tunes four transformer architectures (BERT, RoBERTa, DistilBERT, and EurLexBERT) on the annotated dataset of 3,000 sentences. Each model is trained using five-fold cross-validation, and the average metrics across folds (precision, recall, F1, and MCC) are saved as CSV outputs in *output_tables*. These correspond to Tables A7–A10.

Environment
All Python scripts were executed under a controlled conda environment to ensure reproducibility. The environment specification file *environment.yml* includes the following key dependencies: Python 3.11, spaCy 3.6.1, Pandas 2.1.4, NumPy 1.26.4, and Scikit-learn 1.7.2. All paths and folder structures are automatically created by the configuration file *replication_src/config.py*, allowing full end-to-end replication of the preprocessing, model training, and evaluation steps.

References

Franchino, Fabio. 2007. *The Powers of the Union: Delegation in the EU*. Cambridge University Press.

Migliorati, Marta. 2021. "Where Does Implementation Lie? Assessing the Determinants of Delegation and Discretion in Post-Maastricht European Union." *Journal of Public Policy* 41 (3): 489–514. https://doi.org/10.1017/S0143814X20000100.