Creating and Comparing Dictionary, Word Embedding, and Transformer-based Models to Measure Discrete Emotions in German Political Text

Tobias Widmann & Maximilian Wich

Political Analysis

Replication Material

This document describes the datasets, R scripts, and Python notebooks needed to reproduce all plots and tables included in the paper and in the Online Appendix. In the beginning of each script, the necessary packages, the versions they were used in, and the necessary data files are listed.

We recommend that you set the working directory (using *setwd()*) to the replication folder that contains the data at the beginning of each script. We also recommend to leave all the python files in the sub-folder called 'electra'.

Furthermore, in order for the files to run without errors, you must unzip all zipped files in the replication folder before starting the replication. This means that the files **boja_wiki.de.vec.zip** and **miko_cc.de.300.vec.zip** need to be unzipped first.

Computational Platform

MacBook Pro 2017, MacOS Big Sur 11.5.1, 2.3 GHz Dual-Core Intel Core i5, 16 GB RAM 2133 MHz LPDDR3.

Running time

The running time for all files together, using the environment mentioned above to run the code, is approximately 40 hours.

R Version

R version 4.0.5 (2021-03-31)

Necessary R packages

library("quanteda") Version 3.0.0

library("keras") Version 2.6.0

library("corpus") Version 0.10.1

library("ggplot2") Version 3.3.3

library("openxlsx") Version 4.2.3

library("stringr") Version 1.4.0

library("syuzhet") Version 1.0.6

library("irr") Version 0.84.1

library("stats") Version 4.1.1

library("reticulate") Version 1.21

library("randomForest") Version 4.6-14

library("glmnet") Version 4.1-1

library("e1071") Version 1.7-6

library("ROCit") Version 2.1.1

library("dplyr") Version 1.0.5

library("ggpubr") Version 0.4.0

library("Rmisc") Version 1.5

library("dotwhisker") Version 0.6.0

library("devtools") Version 2.4.0

library("reshape2") Version 1.4.4

library("readr") Version 1.4.0

library("rword2vec") Version 1.1

library("readr") Version 1.4.0

library("stargazer") Version 5.2.2

library("stringi") Version 1.5.3

library("tensorflow") Version 2.6.0

Necessary Python Packages

Python version 3.9.5

pandas Version 1.3.4

numpy Version 1.19.5

torch Version 1.10.0

tqdm Version 4.62.3

transformers Version 4.12.0

datasets Version 1.14.0

scikit-learn Version 1.0

openpyxl Version 3.0.9

seaborn Version 0.11.2

h5py Version 2.9.0 (The python h5py package is necessary to use the keras package in R)

LIWC Standalone Application

LIWC 2015 Version 1.6.0

```
- Internal German Dictionary 2015
```

Descriptions of Scripts

- **01_replication_main_analysis.R**: This R script prepares both datasets used in the analysis: the pre-sampled data set (data1) and the randomly sampled data set (data2). The script shows how we applied the different tools to the data, and creates all necessary variables needed for the analysis. It also recreates all tables and graphs displayed in the main text of the manuscript. Please note, however, that the LIWC dictionary cannot be included in this replication material due to copyright reasons. Furthermore, the ELECTRA model is applied by using the Python script "**05_apply_transformer_based_model.ipynb**".
- **02_replication_appendix.R**: This R script replicates all tables and graphs included in the Online Appendices. **Note:** For Appendix M, the ELECTRA model is applied by using the Python script "**05_apply_transformer_based_model.ipynb**". This means that human intervention is required to apply the ELECTRA model. Instructions on how to use the python script are provided in the R script.
- **03_training_models.R**: This R script provides code that we used to train the different machine learning models (excluding the ELECTRA model).
- **04_training_embeddings.R**: This R script shows how the locally trained word embeddings have been trained relying on large 'transformation data' ("**ger_corpus.Rdata**").
- **05_apply_transformer_based_model.ipynb**: This python script shows how the ELECTRA classifier can be applied to text data. In the python scripts, please set the working directory (using *%cd*) to the folder 'electra' within the replication folder. **Note: All necessary python files can be found in the electra ,sub-folder.**

- **05__train__transformer-based__classifier.ipynb**: This python script shows how the ELECTRA classifier has been trained using the training data. In the python scripts, please set the working directory (using *%cd*) to the folder 'electra' within the replication folder. **Note: All necessary python files can be found in the electra sub-folder.**

Descriptions of Data Files In Replication Folder

- **boja__keras__anger**: Neural network models trained on Bojanowski et al. (2017) word embeddings.
- **boja__keras__fear**: Neural network models trained on Bojanowski et al. (2017) word embeddings.
- **boja__keras__disgust**: Neural network models trained on Bojanowski et al. (2017) word embeddings.
- **boja__keras__sadness**: Neural network models trained on Bojanowski et al. (2017) word embeddings.
- **boja__keras__joy**: Neural network models trained on Bojanowski et al. (2017) word embeddings.
- **boja__keras__enthusiasm**: Neural network models trained on Bojanowski et al. (2017) word embeddings.
- **boja__keras__pride**: Neural network models trained on Bojanowski et al. (2017) word embeddings.
- **boja__keras__hope**: Neural network models trained on Bojanowski et al. (2017) word embeddings.
- **boja__wiki.de.vec.zip**: Word vectors from Bojanowski et al. (2017)[1]. Zip file needs to be unpacked first.
- **case__sample.Rdata**: Data set used in Appendix M. The data file includes a total of 168,349 sentences that come from press releases of German parties, from the period of January 2016 to October 15, 2018.
- **coder1.Rdata**: Classification of words into emotional categories, made by Coder 1.
- **coder2.Rdata**: Classification of words into emotional categories, made by Coder 2.
- **data1__prepared.Rdata**: The prepared data set 1, with all tools applied.
- **data1__raw.Rdata**: The raw data set 1, as received from the crowd-coding company.
- **data2__prepared.Rdata**: The prepared data set 2, with all tools applied.
- **data2__raw.Rdata**: The raw data set 2, as received from the crowd-coding company.
- **ed8.RData**: The ed8 dictionary.
- **ed8.yml**: The ed8 dictionary in yml format.

[1] https://arxiv.org/pdf/1607.04606.pdf

- **electra.zip**: Zip file containing all python code and data. Please leave these files within the 'electra' folder.
- **embed_boja.Rdata**: The embeddings for the data1 file produced by Bojanowski et al. (2017) word embeddings
- **embed_boja2.Rdata**: The embeddings for the data2 file produced by Bojanowski et al. (2017) word embeddings
- **embed_miko.Rdata**: the embeddings for the data1 file produced by Mikolov et al. (2017) word embeddings
- **embed_miko2.Rdata**: the embeddings for the data2 file produced by Mikolov et al. (2017) word embeddings
- **embed_poly.Rdata**: the embeddings for the data1 file produced by Polyglot word embeddings
- **embed.Rdata**: the embeddings for the data1 file produced by locally trained word embeddings
- **embed2.Rdata**: the embeddings for the data2 file produced by locally trained word embeddings
- **ger_corpus.Rdata**: the 'transformation data' file, consisting of ~ 2 million text documents (including ParlSpeech V2 speeches from Rauh & Schwalbach 2020[2])
- **keras_anger90**: Neural network models based on locally trained word embeddings.
- **keras_fear90**: Neural network models based on locally trained word embeddings.
- **keras_disgust90**: Neural network models based on locally trained word embeddings.
- **keras_sadness90**: Neural network models based on locally trained word embeddings.
- **keras_joy90**: Neural network models based on locally trained word embeddings.
- **keras_enthusiasm90**: Neural network models based on locally trained word embeddings.
- **keras_pride90**: Neural network models based on locally trained word embeddings.
- **keras_hope90**: Neural network models based on locally trained word embeddings.
- **lasso_anger**: Lasso models based on locally trained word embeddings.
- **lasso_fear**: Lasso models based on locally trained word embeddings.
- **lasso_disgust**: Lasso models based on locally trained word embeddings.

[2]Rauh, C., & Schwalbach, J. (2020). The ParlSpeech V2 data set: Full-text corpora of 6.3 million parliamentary speeches in the key legislative chambers of nine representative democracies [Data set]. Harvard Dataverse.

- **lasso_sadness**: Lasso models based on locally trained word embeddings.
- **lasso_joy**: Lasso models based on locally trained word embeddings.
- **lasso_enthusiasm**: Lasso models based on locally trained word embeddings.
- **lasso_pride**: Lasso models based on locally trained word embeddings.
- **lasso_hope**: Lasso models based on locally trained word embeddings.
- **liwc_results.csv**: Results returned from the LIWC stand-alone application (data1).
- **liwc_results2.csv**: Results returned from the LIWC stand-alone application (data2).
- **liwc.csv**: temporary data file (data1) that is exported from R to be used in the LIWC stand-alone application.
- **liwc2.csv**: temporary data file (data2) that is exported from R to be used in the LIWC stand-alone application.
- **miko_cc.de.300.vec.zip**: Word vectors from Mikolov et al. (2017)[3]. Zip file needs to be unpacked first.
- **miko_keras_anger**: Neural network models based on Mikolov et al.

(2017) word embeddings.

- **miko_keras_fear**: Neural network models based on Mikolov et al.

(2017) word embeddings.

- **miko_keras_disgust**: Neural network models based on Mikolov et al. (2017) word embeddings.
- **miko_keras_sadness**: Neural network models based on Mikolov et al. (2017) word embeddings.
- **miko_keras_joy**: Neural network models based on Mikolov et al.

(2017) word embeddings.

- **miko_keras_enthusiasm**: Neural network models based on Mikolov et al. (2017) word embeddings.
- **miko_keras_pride**: Neural network models based on Mikolov et al.

(2017) word embeddings.

- **miko_keras_hope**: Neural network models based on Mikolov et al.

(2017) word embeddings.

[3]https://arxiv.org/abs/1607.04606

- **nb_anger.RData**: Naive Bayes classifiers based on locally trained word embeddings.
- **nb_fear.RData**: Naive Bayes classifiers based on locally trained word embeddings.
- **nb_disgust.RData**: Naive Bayes classifiers based on locally trained word embeddings.
- **nb_sadness.RData**: Naive Bayes classifiers based on locally trained word embeddings.
- **nb_joy.RData**: Naive Bayes classifiers based on locally trained word embeddings.
- **nb_enthusiasm.RData**: Naive Bayes classifiers based on locally trained word embeddings.
- **nb_pride.RData**: Naive Bayes classifiers based on locally trained word embeddings.
- **nb_hope.RData**: Naive Bayes classifiers based on locally trained word embeddings.
- **negative_ed8.RData**: ed8 dictionary with negation pattern.
- **poly_keras_anger**: Neural network models based on Polyglot word embeddings.
- **poly_keras_fear**: Neural network models based on Polyglot word embeddings.
- **poly_keras_disgust**: Neural network models based on Polyglot word embeddings.
- **poly_keras_sadness**: Neural network models based on Polyglot word embeddings.
- **poly_keras_joy**: Neural network models based on Polyglot word embeddings.
- **poly_keras_enthusiasm**: Neural network models based on Polyglot word embeddings.
- **poly_keras_pride**: Neural network models based on Polyglot word embeddings.
- **poly_keras_hope**: Neural network models based on Polyglot word embeddings.
- **polyglot-de.pkl**: Polyglot word vectors[4].
- **prob_elek.Rdata**: ELECTRA model classification probabilities for data1.
- **prob_elek2.Rdata**: ELECTRA model classification probabilities for data2.
- **read_pickle.py**: Helper file to read pickle format.
- **results2.Rdata**: The total set of 75,000 sentences from data2.
- **rFanger.RData**: Random Forrest classifiers based on locally trained word embeddings.
- **rFfear.RData**: Random Forrest classifiers based on locally trained word embeddings.
- **rFdisgust.RData**: Random Forrest classifiers based on locally trained word embeddings.

[4]https://sites.google.com/site/rmyeid/projects/polyglot

- **rFsadness.RData**: Random Forrest classifiers based on locally trained word embeddings.
- **rFjoy.RData**: Random Forrest classifiers based on locally trained word embeddings.
- **rFenthusiasm.RData**: Random Forrest classifiers based on locally trained word embeddings.
- **rFpride.RData**: Random Forrest classifiers based on locally trained word embeddings.
- **rFhope.RData**: Random Forrest classifiers based on locally trained word embeddings.
- **SnowballStopwordsGerman.txt**: Snowball Stopword list for German.
- **source_info.Rdata**: Data file including information about text type for each sentence of data1.
- **source_info2.Rdata**: Data file including information about text type for each sentence of data2.
- **test_data.Rdata**: Data file including test data from data 1.
- **text_ed_preprocessed**: locally trained word embeddings.
- **text_ed_preprocessed.txt**: locally trained word embeddings in txt-format.

Descriptions of Data Files In Electra Sub-Folder

- **data/:** the folder contains the data that is required for training, testing, and inferring
  - **labeled_training_data_df.pkl**: the Pickle file contains the training data used by 05_train_transformer-based_classifier.ipynb
  - **labeled_test_data_df.pkl**: the Pickle file contains the test data used by 05_train_transformer-based_classifier.ipynb
- **helper/:** the folder contains Python code that is used for training and inferring.
  - **data_preparation.py**: the file contains Python functions to load and prepare the data for training and inferring
  - **inferencing.py**: the file contains Python functions to predict the emotions for unlabeled data (sentecnes)
  - **training.py**: the file contains Python functions that are used during the training of the classification model (ELECTRA)
- **models/:** the folder contains the trained model
  - **config.json**: the config file for the trained ELECTRA classification model

  - **pytorch_model.bin**: the actual classification model

- **results/**: the folder contains temporary data that is created during training (checkpoints)
- **electra_appendixm.csv**: the CVS file contains the raw data from Appendix M; the file is used by the notebook 05_apply_transformer_based_model.ipynb
- **electra_results_appendixm.csv**: the CSV file contains the predictions generated by the ELECTRA model for data from Appendix M
- **electra_results.csv**: the CSV file contains the predictions generated by the ELECTRA model for the test data
- **electra_results2.csv**: the CSV file contains the predictions generated by the ELECTRA model for data2
- **electra.csv**: the CVS file contains the test data; the file is used by the notebook 05_apply_transformer_based_model.ipynb
- **electra2.csv**: the CSV file contains the data used in the robustness test; the file is used by the notebook 05_apply_transformer_based_model.ipynb
- **requirements.txt**: the file contains a list of necessary python packages. Note: To install all required python packages listed in requirements.txt you can simply use the following pip-command:

  pip install -r requirements.txt

  (see https://pip.pypa.io/en/stable/user_guide/)