Readme

"Word Embeddings for the Analysis of Ideological Placement in Parliamentary Corpora" Political Analysis Dataverse

Ludovic Rheault[†] and Christopher Cochrane[‡]

Description

The files contained in this repository can be used to reproduce the results reported in the main text and the supplementary materials (online appendix) of the paper "Word Embeddings for the Analysis of Ideological Placement in Parliamentary Corpora" (*Political Analysis*).

A Python module is also made available in a public GitHub repository to explore the data (https://github.com/lrheault/partyembed). The Python notebook examples.ipynb available in that repository allows to reproduce the key results using an interactive computing environment.

Minimum Requirements

The scripts contained in this repository can be run on most modern computers. We recommend using a 64-bit operating system, a 2 core computer with a minimum of 2 GB of RAM. The uncompressed files and library installation procedure outlined below require 1.3 GB of disk space.

Package Requirements

The results were computed using Python 3.5. We recommend using a version of Python 3.5 or above.

The R script requires installing the package *ggplot2*. The requirements for the main Python scripts are detailed in the file requirements.txt (see procedure below).

[†]Assistant Professor, Department of Political Science and Munk School of Global Affairs and Public Policy, University of Toronto.

[‡]Associate Professor, Department of Political Science, University of Toronto.

Running Time

The following benchmarks were clocked on a personal computer with four Intel Core i7-4790 CPUs (3.60GHz), 16Gb of RAM and a Debian GNU/Linux 9.8 operating system:

File	Wall Clock Running Time (in Seconds)
main_results.py	8.778s
appendix_results.py	11.667s
polarization.R	0.208s

Replicating Figures and Tables

We recommend using a Python virtual environment to avoid compatibility issues with the libraries needed, which are listed in the requirements.txt file. Below is a summary of the key steps to set up a virtual environment and run the main scripts. These commands are designed to be run from the operating system's command line interface.

Note: On some operating systems, the *python3* and *pip3* commands may be needed below if Python 3 is not the default language associated with the *python* command, as is the case in the default settings on several Linux distributions.

1. Download and extract the Dataverse repository files, then enter the directory:

```
unzip dataverse_files.zip
cd dataverse_files
```

2a. Install the Python virtualenv package (if needed):

```
pip install virtualenv
```

2b. Create and activate a virtual environment:

```
virtualenv partyembed
source partyembed/bin/activate

# On Windows:
partyembed\Scripts\activate
```

3. Install the requirements:

```
pip install -r requirements.txt
```

4. Run the main results script (note, figures and tables will be overwritten in the figures/ and tables/ directory):

```
python main_results.py
```

5. To reproduce the results from the online appendix:

```
python appendix_results.py
```

6. Deactivate the virtual environment:

```
deactivate
```

7. Polarization figures (must be run after appendix_results.py):

```
Rscript polarization.R
```

To fully remove the virtual environment files, simply delete the directory partyembed/ that has been created during the procedure.

Notes on Virtual Environments

Using a Python virtual environment allows to install specific library requirements to reproduce results while avoiding conflicts with other versions of these libraries that may already be installed on a given machine. This step is not entirely necessary, but is a good practice to preserve stability when working on multiple projects on the same machine. Experienced users will be able to replicate the results without this, by updating the required packages if necessary.

License

These materials are released under a Creative Commons Zero ("CC0") license. Third-party software libraries are subjected to their respective licenses.

Fitting New Models with the Raw Corpora

The src/ directory contains example files to reformat and process the raw US parliamentary corpus, then fit a full model using the methodology described in the paper. Detailed information about the three corpora can be found in the appendix to the paper. Consult the associated GitHub repository for additional information and ongoing developments.