Replication for *Going against the Grain: Climate Change as a Wedge Issue for the Radical Right*

Instructions

This repository contains the replication files for the the paper "Going against the Grain: Climate Change as a Wedge Issue for the Radical Right" by Zachary P Dickson and Sara Hobolt The paper is forthcoming in *Comparative Political Studies*.

To replicate the analysis, users will need to have Python and Jupyter installed on their machine. Anaconda provides a convenient way to install both.

There are **Detailed Instructions** in the `main.ipynb` file that provide a step-by-step guide to replicating the analysis.

There are three folders in this repository: `data`, `code`, and `output`. The `data` folder contains the data files required for the analysis. The `code` folder contains the code files required to replicate the analysis. The `output` folder contains the output files generated by the analysis. Additionally, there is a `requirements.txt` file that lists the Python packages required to run the analysis.

Data

The `data` folder contains the following files:

1. `subsidise_renewables.csv` - A CSV file containing the EES data on support for subsidising renewables.
2. `fossil_fuel_taxes.csv` - A CSV file containing the EES data on support for fossil fuel taxes.
3. `topic_distribution.csv` - A CSV file containing the topic distribution data (Supplementary Material).
4. `fig5_df.pkl` - A pickle file containing the data for Figure 5

5. `fig7_df.pkl` - A pickle file containing the data for Figure 7
6. `gles_15.pkl` - A pickle file containing the GLES data (wave 15)
7. `press_releases.pkl` - A pickle file containing the press release data

Code

All code files are available in the `code` folder. The `code` folder contains the following files:

1. `main.ipynb`: Jupyter notebook with the analysis code - (**Note:** this is the main file to run to replicate the analysis.)
2. `functions.py`: A python file containing all functions necessary for the analysis.

Data Collection

Election Study Data: We use data from the European Election Study, British Election Study, and the German Longitudinal Election Study. The data are available for download from the respective websites. We also provide the code required to clean these data and create the figures in the article (see `main.ipynb`). We also provide the cleaned versions of the data in the `data` folder.

Press Releases: We included an abbreviated version of the Party Press Release dataset in the `data` folder. The included data is everything that is required to replicate the entire analysis, but does not include the additional countries from the original dataset from Erfort et al. or the text associated with the press releases. As the full dataset is not ours, we ask that you request access from the authors of the original dataset if you want the text associated with the press releases. With that said, the dataset we include has all the outcomes and covariates required to replicate the analysis in the paper and could be used in a similar manner to the full dataset.

Language Model

We use two language models in the analysis. The first is a classification model and is publicly available on the Huggingface model hub https://huggingface.co/z-dickson/CAP_multilingual. The second is a generative model and is publicly available on the Huggingface model hub https://huggingface.co/z-dickson/bart-large-cnn-climate-change-summarization. The models can be accessed using the following code:

Classification Model:

```
from transformers import AutoModelForSequenceClassification
from transformers import TextClassificationPipeline, AutoTokenizer
```

```
mp = 'z-dickson/CAP_multilingual'
model = AutoModelForSequenceClassification.from_pretrained(mp)
tokenizer =  AutoTokenizer.from_pretrained('bert-base-uncased')

classifier = TextClassificationPipeline(tokenizer=tokenizer, model=model, device=0)

classifier("""
To ask the Secretary of State for Energy and Climate \\
Change what estimate he has made of the proportion of carbon \\
dioxide emissions arising in the UK attributable to burning.
"""
)
```

Generative Model:

```
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer

mp = 'z-dickson/bart-large-cnn-climate-change-summarization'
model = AutoModelForSeq2SeqLM.from_pretrained(mp)
tokenizer = AutoTokenizer.from_pretrained(mp)

text = """
To ask the Secretary of State for Energy and Climate Change \\
what estimate he has made of the proportion of carbon dioxide emissions \\
arising in the UK attributable to burning.
"""
inputs = tokenizer(text,
                   return_tensors='pt',
                   max_length=1024,
                   truncation=True)
summary_ids = model.generate(
    inputs['input_ids'],
    num_beams=4,
    max_length=100,
    early_stopping=True)
summary = tokenizer.decode(
    summary_ids[0],
    skip_special_tokens=True)

print(summary)
```

Note

There are **Detailed Instructions** in the `main.ipynb` file that provide a step-by-step guide to replicating the analysis.

In the case that you're unable to find any of the required data, or have any issues with the replication, please don't hesitate to contact me at zachdickson94@gmail.com.