

Python Setup

Alexander P. Christensen and Hudson Golino

April 04, 2022

1. Getting Started with Python

If you have Python already installed on your computer, then skip to section 2 to check your Python installation.

If not, then the first step is to download Python. The recommended version is 3.9.7: <https://www.python.org/downloads/release/python-397/>. Below are the recommended settings when installing Python:

- Check the “Add Python VERSION.NUMBER to PATH”
- Then, select “Customize installation”
- Make sure all “Optional Features” are checked and click “Next”
- For “Advanced Options,” check “Precompile standard library” and click “Install”

Feel free to choose a custom install location. The `python_setup()` function will automatically detect where the path to your Python installation is. You can also enter this install location manually.

After Python is finished installing, you should check whether it’s been installed properly.

2. Check Python Installation

The following code will check if Python is installed and where Python is installed

```
print("Is Python installed?")
reticulate::py_available(TRUE)
```

```
[1] "Is Python installed?"
[1] TRUE
```

If `FALSE`, then try the following steps:

- Restart your computer and try the above code again
- If still `FALSE`, then go back to 1. and try re-installing Python on your computer following the instructions

If `TRUE`, then Python has successfully installed on your computer. To find out where, the following code can be used

```
print("Where is Python installed?")
reticulate::py_config()
```

```
[1] "Where is Python installed?"
python:      D:/Python/python.exe
libpython:   D:/Python/python39.dll
pythonhome:  D:/Python
version:     3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)]
Architecture: 64bit
numpy:       D:/Python/Lib/site-packages/numpy
numpy_version: 1.22.2
```

To make R aware of your Python installation, you can use the following code

```
python_setup()
[1] "D:/Python/python.exe"
```

3. Installing the transformers Module

With Python installed, the next step is to install the `transformers` module. To do so, you'll need to open a command line terminal on your computer. Once open, you can start by upgrading `pip`

```
python -m pip install --upgrade pip
```

Before installing the `transformers` module, a few other packages need to be installed: `PyTorch` and `Tensorflow`.

`PyTorch` can be installed by following the instructions on the website: <https://pytorch.org/>. Select the "Stable" build and your operating system. You will most likely use the "pip" package for install but "conda" is another common package if you're using Anaconda or miniconda (see <https://www.anaconda.com/>). Language should be "Python" and platform can be "CPU." "CUDA" is used for GPU processing but it is not necessary and requires additional steps for setting up CUDA.

From the command line, here are the installs for different operating systems

Windows

```
pip install torch torchvision torchaudio
```

Macs

```
pip install torch torchvision torchaudio
```

Linux

```
pip install torch==1.10.1+cpu torchvision==0.11.2+cpu torchaudio==0.10.1+cpu \newline -f
https://download.pytorch.org/whl/cpu/torch_stable.html
```

`TensorFlow` can be install using similar instructions (<https://www.tensorflow.org/install>) but it is much more straightforward. From the command line, you can use

```
pip install tensorflow
```

Finally, you can install the `transformers` module:

```
python -m pip install transformers
```

To check that the `transformers` module was properly installed, the following code can be run

```
print("'transformers' module installed?")
reticulate::py_module_available("transformers")
```

```
[1] "'transformers' module installed?"
[1] TRUE
```

At this point, the `transformers` module should be ready-to-go on your computer. If you're having trouble installing modules, then check out this page: <https://packaging.python.org/en/latest/tutorials/installing-packages/>.

4. Downloading Cross-Encoder's DistilRoBERTa

The final step is to download the [Cross-Encoder's DistilRoBERTa](#) transformer model. The simplest way is to run our `transformer_scores` function with an example

```
# Load data
data(neo_ipip_extraversion)

# Example text
text <- neo_ipip_extraversion$friendliness[1:5] # positively worded items only

# Run transformer function
transformer_scores(
  text = text,
  classes = c(
    "friendly", "gregarious", "assertive",
    "active", "excitement", "cheerful"
  )
)
```

```
$`make friends easily`
friendly gregarious assertive active excitement cheerful
0.579 0.075 0.070 0.071 0.050 0.155

$`warm up quickly to others`
friendly gregarious assertive active excitement cheerful
0.151 0.063 0.232 0.242 0.152 0.160

$`feel comfortable around people`
friendly gregarious assertive active excitement cheerful
0.726 0.044 0.053 0.042 0.020 0.115

$`act comfortably around people`
friendly gregarious assertive active excitement cheerful
0.524 0.062 0.109 0.183 0.019 0.103

$`cheer people up`
friendly gregarious assertive active excitement cheerful
0.071 0.131 0.156 0.190 0.362 0.089
```

The download will take some time: The model is 333MB. The model will download to your Python directory *not* R. The model only needs to be downloaded once and will be loaded each time the `transformer_scores` function is called in a new R session.

We can compare these results with a more basic natural language processing approach: continuous bag of words. This approach can be implemented using the `nlp_scores()` function

```
# Run NLP function
nlp_scores(
  text = text,
  classes = c(
    "friendly", "gregarious", "assertive",
    "active", "excitement", "cheerful"
  ),
  semantic_space = "cbow"
)
```

```
$`make friends easily`
friendly gregarious assertive active excitement cheerful
0.192 0.279 0.151 0.036 0.081 0.089

$`warm quickly others`
friendly gregarious assertive active excitement cheerful
0.126 0.276 0.072 0.150 0.141 0.176

$`feel comfortable around people`
friendly gregarious assertive active excitement cheerful
0.222 0.353 0.247 0.117 0.214 0.133

$`act comfortably around people`
friendly gregarious assertive active excitement cheerful
0.198 0.365 0.157 0.092 0.183 0.122

$`cheer people`
friendly gregarious assertive active excitement cheerful
0.225 0.229 0.168 0.034 0.204 0.279
```

Similarly, this function will take some time because the semantic space (249MB) needs to be downloaded. The semantic space will need to be downloaded during each R session.

The `nlp_scores()` function returns semantic similarity between the text and classes rather than probabilities (like `transformer_scores()`).

If you've made it this far, then you've successfully obtain sentiment analysis scores from [Cross-Encoder's DistilRoBERTa](#) transformer model. Go forth and quantify the qualitative!