Natural Language Processing with R and Python

Steven Nydick Ben Wiseman Tristan Wisner

April 29, 2022

Setup

1. Requires
   - https://www.r-project.org/
2. Suggests
   - https://www.rstudio.com/products/rstudio/download/
3. Materials
   - Git: https://github.com/swnydick/siop-2022-nlp-r-python
   - Folder: https://github.com/swnydick/siop-2022-nlp-r-python/archive/refs/heads/master.zip

Package Installation

This demonstration requires the following R packages:

1. R/Python Section
   - reticulate
   - rstudioapi
2. Neural Nets Section
   - All of the above . . . plus
   - tensorflow
   - keras
   - ggplot2
   - caret
   - mlbench
3. Natural Language Processing Section
   - All of the above . . . plus
   - roperators
   - sentiment.ai
   - sentimentr
   - SnowballC
   - tfhub
   - tm

Package Installation

This folder uses the `renv` package. To set things up, simply connect to the internet, open the project in RStudio, and then run the following line of code:

```
renv::restore(prompt = FALSE)
```

If you do not want to use RStudio, you need to make sure you are in the correct directory and then run the `renv::restore(prompt = FALSE)` line of code.

If you do not want to use `renv`, you will need to install everything manually (for example):

```
install.packages("reticulate")
```

And then load them with the following line of code (for example):

```
library(reticulate)
```

And then there's keras/tensorflow ...

Keras/Tensorflow Package Installation

Keras/Tensorflow needs to link to python to work correctly, through the reticulate package in R. The easiest way of doing this is via miniconda.

1. Install the required packages

```
# installing keras should install reticulate if it's not a
install.packages("keras")
```

2. Install miniconda and create a new environment (called "r-reticulate" for ease, but you can call it anything you want). You can also do this via the command line if it's easier and then link it using Global Options in RStudio.

```
reticulate::install_miniconda()
reticulate::conda_create("r-siop-nlp")
```

3. Install keras into the conda environment

```
keras::install_keras(method  = "conda",
                     envname = "r-siop-nlp")
```

What we want from you:

1. This session assumes familiarity with `R`. Please keep questions relevant to the topics. We have a later section covering more basic `R` programming.
2. Try to run all of the code in `RStudio`. The setup of the demonstrations naturally works in `RStudio`. If you do not have `RStudio`, you can certainly run all of the code in `R` or a different IDE.
3. If you have trouble with setting up tensorflow/keras, we can walk around and try to debug! That said, hopefully the first section will prevent most issues.
4. Have fun!

Neural Networks

The goal of this presentation is to give you the tools to run fully-fledged neural net sentiment models in `R`.

The most common neural net interface is `keras` via `tensorflow`, which is a Google-developed software library commonly implemented via Python.

How will we do this? We will . . .

1. . . . describe how to link Python libraries to `R` using the `reticulate` package. There are several pitfalls to keep in mind that makes it somewhat tricky to set things up correctly!
2. . . . explain how `keras` works and use the `keras` package in `R` to illustrate basic tensor concepts.
3. . . . illustrate simple neural net models using tensors and show that the most basic neural nets are just a very complicated version of regression models.
4. . . . demonstrate how to turn text data into numbers and then use those numbers to predict sentiment scores using neural nets.

Neural Net Translations

When You See...	Think...
Activation	Link Function
Binary Cross-Entropy Loss	Log-Likelihood (Negated)
Epochs	Iterations
Dense Layer	Regression Model
Weights	Slopes
Bias	Intercepts

Exercises