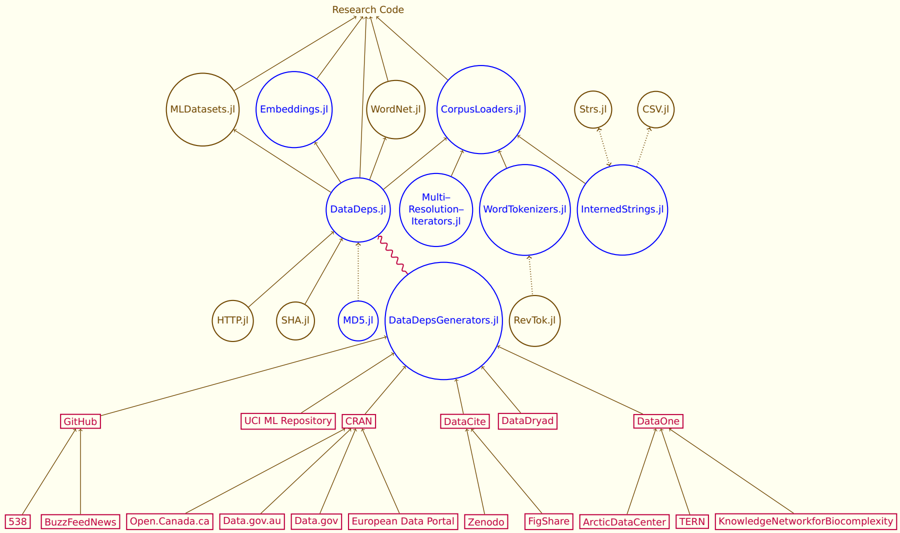
# DataDeps.jl

## and other foundational tools for data driven research (Especially NLP)

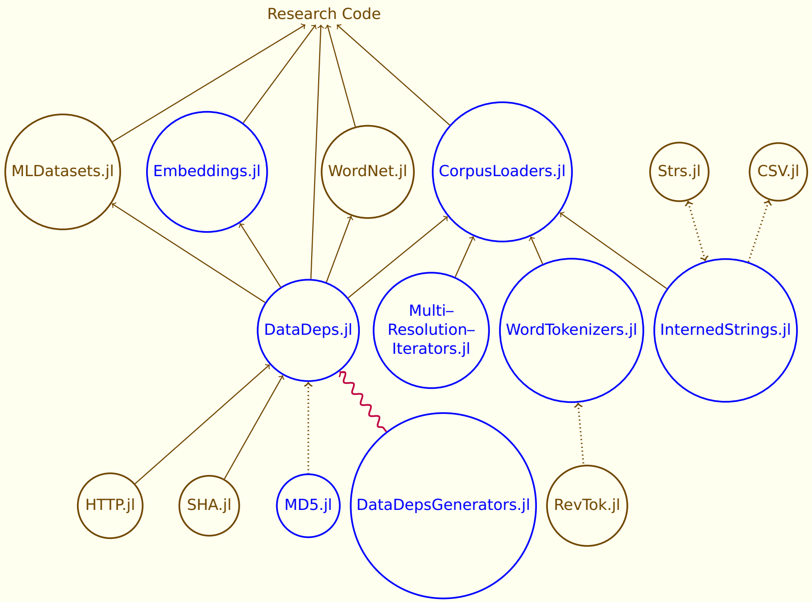
**Lyndon White**

School of Electical, Electronic and Computer Engineering
The University of Western Australia

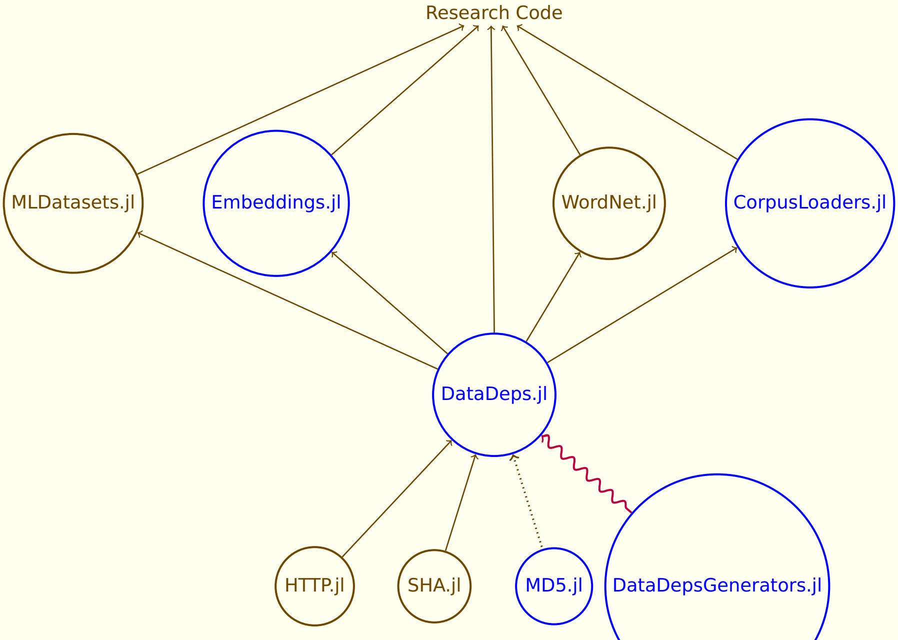
# Big Picture


Research Code
MLDatasets.jl
Embeddings.jl
WordNet.jl
CorpusLoaders.jl
Strs.jl
CSV.jl
DataDeps.jl
Multi–Resolution–Iterators.jl
WordTokenizers.jl
InternedStrings.jl
HTTP.jl
SHA.jl
MD5.jl
DataDepsGenerators.jl
RevTok.jl
GitHub
UCI ML Repository
CRAN
DataCite
DataDryad
DataOne
538
BuzzFeedNews
Open.Canada.ca
Data.gov.au
Data.gov
European Data Portal
Zenodo
FigShare
ArcticDataCenter
TERN
KnowledgeNetworkforBiocomplexity

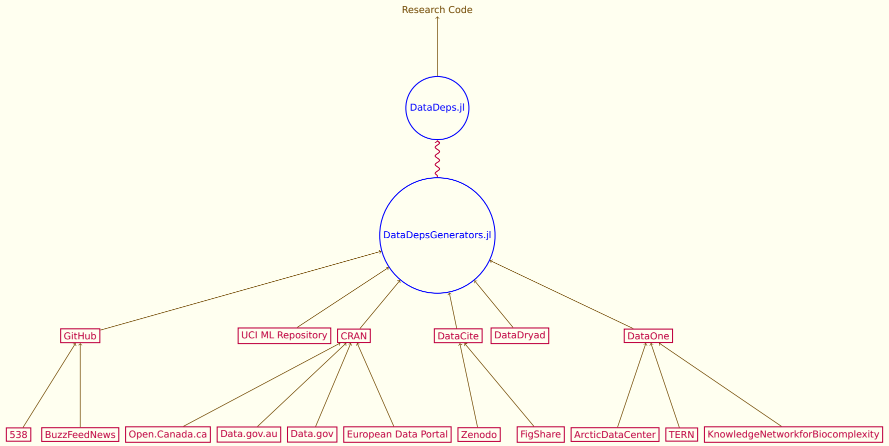
# Julia Packages



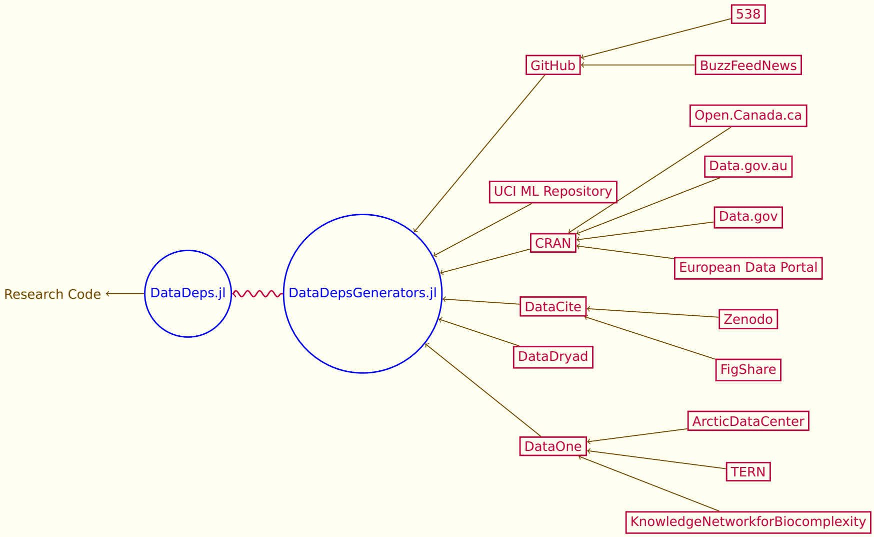
# DataDeps.jl


Research Code
MLDatasets.jl
Embeddings.jl
WordNet.jl
CorpusLoaders.jl
DataDeps.jl
HTTP.jl
SHA.jl
MD5.jl
DataDepsGenerators.jl

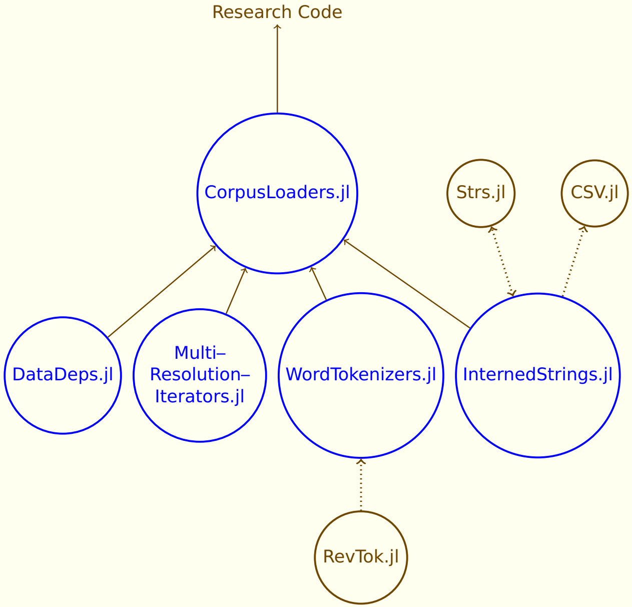
# DataDepsGenerators.jl



# DataDepsGenerators.jl



# CorpusLoaders.jl



# Vabdewakke’s 6 Degree’s of Replicability

1. The results cannot seem to be reproduced.
2. The results could be reproduced by, requiring extreme effort.
3. The results can be reproduced, requiring considerable effort.
4. The results can be easily reproduced with at most **15 minutes** of user effort, requiring some proprietary source packages (MATLAB, etc.).
5. The results can be easily reproduced with at most **15 min** of user effort, requiring only standard, freely available tools (C compiler, etc.).

Vandewalle, Kovacevic, and Vetterli
(2009),“Reproducible research in signal processing”

# What happens when I try and reproduce someone’s research code?

|  |  |
|---|---|
| 1min | Find the website from the paper, and download the code |
| 2min | Read enough of the README to get rough bearings |
| **1min** | Find out where to get the data from and download the data |
| **2min** | Try and remember how to use `tar -xzfvalphabetsoup` etc. |
| **2min** | Workout how to tell script where data is |
| 2min | Setup any software dependencies etc. |
| 3min | Run the code and make sure it isn’t crashing etc. |
| 2min | Interpret the output |

## You can’t trust hardcoded paths; but they are nice to work with.

- Ideally we’ld just use hard-coded, absolute paths
- Absolute paths work with all applications
- Hard-coding the paths in code means less typing
- But they break if anything is moved.
- Making the path be passed in as an argument to the script solves this
  - but now user has to be typing it in to run it.
  - So harder to use.
  - You could include a bash-script that invokes it with the path, but now you’re just hard coding it somewhere else

## You could making the path be passed in as an argument. But...

- Now user has to be typing it in to run it.
- So it is harder to use.
- You could include a bash-script that invokes it with the path, but now you’re just hard-coding it somewhere else

## `datadep"Census 2018/populations.csv"` A path you can trust

- Always resolves to an absolute path to that file
- Even if that means it has to download it first
- But before resorting to downloading checks a large number of places
  - `<PKG>/deps/data`,
  - `~/.julia/datadeps`,
  - `/usr/share/datadeps`, etc.
- You know that if you use a datadep path it will resolve to a file that exists.

# Current Usages of DataDeps.jl

MLDatasets.jl

- Provides easy access to a bunch of ML datasets
- `xs, ys = MNIST.traindata()`
- Gives you regular julia arrays

CorpusLoaders.jl

- Provides easy access to linguistic corpora
- `corpus_gen = load(WikiCorpus())`
- gives you a multi-resolution iterator

# Current Usages of DataDeps.jl

Embeddings.jl

- Provides access to hundreds of pretrained word embedding models.
- `load_embeddings(FastText_Text{:fr})`
- gives you a table of French word embeddings.

WordNet.jl

- Look up lexical relations and definitions.
- `lemma = db['a', "glad"]`
- `antonyms(db, synsets(db, lemma)[1])`

# DataDep Registration Block

```
register(DataDep("DataDepName",
"""
Free Text Field Displayed to user before download.
Use to give credit, and tell people about licensing.
Or other messages.
""",
"Download URL",
"file hash (will be printed if not provided)";
post_fetch_method = function to run on downloaded files
))
```

# Registration Block Example

```
register(DataDep("WordNet 3.0",
"""
Dataset:  WordNet 3.0
Website:  https://wordnet.princeton.edu/wordnet

George A. Miller (1995).
WordNet:  A Lexical Database for English.
Communications of the ACM Vol.  38, No.  11:  39-41.

License:
This software and database is being provided to you,
the LICENSEE, by Princeton University under
the following license...
""",
"http://wordnetcode.princeton.edu/3.0/WNdb-3.0.tar.gz",
"658b1ba191f5f98c2e9bae3e25...";
post_fetch_method = unpack
))
```

## Registration Block: Recursive Example

```
using MD5

register(DataDep("DataDepNameRec",
"""
Warning these files are all together 39.8GB
""",
["http://example.com/readme.txt",
    ["http://example.com/data1.zip",
     "http://example.com/data2.tar.gz",
    ]
],
(md5, "d41d8cd98f00b204e9800998ecf8427e")
post_fetch_method = [identity, unpack]
))
```

## DataDepGenerators.jl

## Developers still have to write registration blocks

- DataDeps.jl shifts the work from manually to automatic
- But defining the work still has to be done.
- Writing a registration block normally means copy and pasting from a website.
    - Download URL
    - Author Name
    - Publication Date
    - License

For published data this information is all available from some API

## GitHub

- $85 \times 10^6$ repositories
  - Some of them are data
  - Not a great place for data, but commonly used
- BuzzFeedNews:
  - 1 Repo per dataset
- 538:
  - 1 shared Repo with a folder for each dataset
- We generate URLs pointing at `https://cdn.rawgit.com/`
  - This is backed by StackPath CDN
  - It is generated to point at the latest commit at generation time

## DataOne

- Data Observation Network for Earth
  - $\sim$40 Earth and Environment Science repositories
  - $1.2 \times 10^6$ Data Files
  - Seems to have iffy metadata, varying between nodes.

## DataDryad

- DataDryad
  - $7.1 \times 10^5$ Data Files
  - $1.15 \times 10^6$ Data Packages
  - Seems to have iffy metadata, varying between nodes.

## DataCite

- $11 \times 10^6$ DOIs issued
  - Mostly to datasets, though they count also figures as data
  - If you have a DOI and it points at data, it is probably from DataCite
- Problem is all their metadata is missing download URLs
  - So user has to add them manually after
- This is effectively the final fallback for anything with a DOI.

## WordTokenizers.jl

Configurable `tokenizer` and `sentence_segmenter`.
Abuses `eval` and `#265` so that you can change the tokenizer being used globally.
Also compatible with externally defined tokenizers like RevTok.jl.

Nabbed the original Penn Tokenizer sed-script.
Wrote some code that converts basic sed language into julia AST.
Ported some of NLTK's tokenizers into sed.

Rule-based sentence splitter based on Sampo Pyysalo & Yoshimasa Tsuruoka's perl script.

Regex is just really good at working with English.

## InternedStrings.jl: All these duplicate strings are using all my memory

- Strings are immutable, so we only need one copy of each
- We can maintain a pool of `WeakRefs` to each string allocated.
- `str = intern(str)`
  - Add the `str` to the pool if not already
  - replace `str` with an element of the pool
- Because the pool only has `WeakRefs` strings can still be garbage collected.

# MultiResolutionIterators.jl: The structure of a Corpus

- Corpus
- made up of: Documents
- made up of: Paragraphs
- made up of: Sentences
- made up of: Words or Tokens
- made up of: Letters or Characters

## MultiResolutionIterators.jl: Not everyone wants every level of structure

Full corpus structure is
Documents ►Paragraphs ►Sentences ►Words ►Letters

A Corpus Linguist may want
a stream of: Sentences ►Words

An Information Retrieval researcher may want
a stream of: Documents ►Words

A char-RNN language modeller might just want
a stream of Letters