On-demand Research Notebooks with all the Trimmings

The Australian Text Analytics Platform

Alex Ip and Steele Cooke, AARNet
Peter Sefton and Moises Sacal Bonequi, University of Queensland
River Tae Smith, Monash University

eResearch Australasia - October 17, 2023
In the spirit of reconciliation AARNet acknowledges the Traditional Custodians of country throughout Australia and their connections to land, sea and community.

We pay our respect to their Elders past and present and extend that respect to all Aboriginal and Torres Strait Islander peoples today.
AARNet and HASS Research

The Australian Text Analytics Platform (ATAP)

Describing Data and Tools for FAIRness and Reproducibility

Future Work, Impact & Opportunities

Questions + wrap up
AARNet and HASS Research

Not just a pretty network
AARNet and HASS Research
More than just networks - Advancing national research infrastructure by collaborating directly with researchers

Language Data Commons of Australia (LDaCA) and its Australian Text Analytics Platform (ATAP) project

HASS

Humanities and Social Sciences (HASS)

Funded by the Australian Research Data Commons (ARDC)

ARDC is a National Collaborative Research Infrastructure Strategy (NCRIS) facility

NCRIS

National Research Infrastructure for Australia

NCRIS is an initiative of the Department of Education

UQ is the lead organisation

© AARNet Pty Ltd
AARNet and HASS Research
The Language Data Commons of Australia (LDaCA)

What is a Data Commons?

- Technological platform for aggregating, and storing and managing shareable data sets
- Tools and utilities that make use of those data sets possible
- Principles and governance
The Australian Text Analytics Platform

Bringing Easier Computational Analysis to the HASSes
Australia is a massively multilingual society with the world’s oldest continuous cultures situated in one of the most linguistically diverse regions in the world. Large collections of language data have been amassed in Australia, but many remain under-utilised or at risk. Growing demand for text analytics capability and computational methods among HASS researchers. Working group since 2018 to develop a national strategy.
The Australian Text Analytics Platform (ATAP)

General objective: Make reproducible computational analysis easier for researchers
The Australian Text Analytics Platform (ATAP)

What are the major components of the ATAP ecosystem?

ATAP major components for discovery and reproducible code execution:

- **LDaCA Website** for finding data, code and infrastructure
- **Notebook Repositories** with Binder-compatible environment specifications
- **Archival Data Repositories** containing well-structured and well-described language data
- **BinderHub** for reproducible code execution

ATAP components for authentication and authorisation:

- **CILogon** for international institutional authentication
- **REMS** for fine-grained, user-managed access control for both data and compute resources
- **LDaCA API** to provide controlled access to data
The Australian Text Analytics Platform (ATAP)

Binder supports sophisticated Jupyter notebooks with extensive dependencies across all research domains.

Linguistics notebooks courtesy of Sydney Informatics Hub

Non-linguistics example: Geoscience Australia geophys-utils demo notebook
The Australian Text Analytics Platform (ATAP)

Binder can load data from the ATAP Portal through Jupyter notebooks

Loading Farms to Freeways from the API and re-create metadata file

The Language Data Component of Australia EDCat packages all their data collections in an API called "freeways". There is a metadata file called "re-create-structure", which comes with every data collection and is how you can get metadata on this collection of research objects.

The metadata file is in the proper format, and so we're learning how to read a json file in this notebook.

**Skills**
- json file format (see https://en.wikipedia.org/wiki/JSON)
- working with databases, via pandas
- discovering and exploring metadata
- extracting names, via Scrapy

**Skill level** intermediate

**Import libraries**

Python needs the libraries that will be used by the notebook to be specified before they are used. We do this with the reserved word `import`, as shown below.

```python
In [1]: import json  # json library to read json file formats
In [2]: import pandas  # working with databases, via pandas
In [3]: from collections import Counter  # discovering and exploring metadata
In [4]: from freeways_utils import json_list  # handy utility for converting to list
```
Describing Data and Tools for FAIRness and Reproducibility

Everybody hates metadata - unless somebody else looks after it
Describing Data and Tools for FAIRness and Reproducibility

Good metadata is key to integrating data, software and infrastructure

Data repositories well described using RO-crates and OCFL (Oxford Common File Layout).

Notebook repositories well described using RO-crates. Reproducible software & hardware execution environments specified using YAML files.

Crate-O used to populate RO-Crate metadata for both data and notebook Repositories.
Describing Data and Tools for FAIRness and Reproducibility

RO-crate & Crate-O for managing metadata for both data and tools

RO-crate [https://www.researchobject.org/ro-crate/]
- A community effort to establish a lightweight approach to packaging research data (and now tools) with machine-readable metadata
- Based on schema.org annotations in JSON-LD
- Aims to make best-practice in formal metadata description accessible and practical for use in a wider variety of situations
- Already widely used in many research domains including HASS & Bioinformatics

Crate-O [https://github.com/Language-Research-Technology/crate-o]
- A browser-based editor for Research Object Crates (RO-Crates)
- No manual editing of JSON-LD required
- Describes files on a user’s computer and adds contextual information about those files
- Can import bulk metadata from an Excel spreadsheet
Describing Data and Tools for FAIRness and Reproducibility

Need to align with existing software source code metadata schemes wherever possible

**Codemeta**
(https://codemeta.github.io/)

- RO-crate schema has **100% semantic interoperability** with the Codemeta project, with its extensive library of tools
  - Both use terms mainly from schema.org, with no conflicting extension terms
  - Both expressed in JSON-LD
  - RO-crate applies different structural constraints on JSON-LD for predictability and ease of use
  - CodemetaPy can read RO-crate metadata directly with minimal modification (Pull request pending)
  - Codemeta does not explicitly specify hardware resource requirements – may need to harmonise

![How Standards Proliferate](https://xkcd.com/927/)

© AARNet Pty Ltd
Describing Data and Tools for FAIRness and Reproducibility

Software environments specified using standard YAML file: `environment.yml`

An `environment.yml` file in the repo can specify a Conda environment exactly for reproducibility

- Easily created using "conda env export" in the original Conda environment
- Used by BinderHub to re-create the Conda environment in a custom container
- File is currently referenced in the repo’s RO-crate in the schema.org `softwareRequirements` property

```
channels:
  - conda-forge
  - defaults

dependencies:
  - _libgcc_mutex=0.1=conda_forge
  - _openmp_mutex=4.5=3_kmp_llvm
  - affine=2.4.0=py37haf73eb6
  - alsa-lib=1.2.8=blackspear_0
  - aom=3.5.0=h27087fc_0
  - attr=2.5.1=h166bdaf_1
  - blosc=1.14.0=h06b7b65_0
  - Bolton=23.0.0=py310h06a4308_0
  - boost-cpp=1.78.0=h6582d0a_3
  - cftime=1.6.2
  - cftime-utils=0.1.0
  - netcdf4=1.6.4
  - pip=
    - cftime=1.6.2
    - cftime-utils=0.1.0
    - netcdf4=1.6.4
    - zipstream-new=1.1.8
```
Describing Data and Tools for FAIRness and Reproducibility

Hardware resource requirements specified using a new machine-readable YAML file: `resources.yml`

A machine-readable `resources.yml` file can specify the minimum hardware resource requirements of the notebook(s) in the repo

- Can include storage, memory, CPU count, CPU architecture and GPU requirements.
- Will be used by ATAP portal to filter known BinderHub installations by resource levels to ensure successful execution
- Labels and values are derived from JupyterHub configuration YAML to specify resources for Kubernetes pods
- File is currently referenced in the repo’s RO-crate in the schema.org `availableOnDevice` property

```yaml
resources:
  architecture:
    - x86_64
    - aarch64
  cpu: 2
  gpu:
    nvidia.com/gpu: 1
  memory: 2G
  storage: 2Gi
```
Future work, Impact, and Opportunities

ATAP functionality will be progressively migrated to operational LDaCA infrastructure

**Future work:**
- AARNet PaaS
- Support multiple, specialised BinderHubs besides ATAP, ARDC and MyBinder
- Using `resource.yml` to drive dynamic resource allocation for notebook containers in customised BinderHub deployment

**Impact:**
Language researchers will be more easily able to:
- find data and analysis tools
- launch notebooks against datasets quickly and easily
- cite specific data and tools for reproducibility

**Opportunities:**
- Interfacing with National Collections (e.g. Trove)
- Possible adaptation of work for other research domains
Thank You.
Questions?