P.A.P.E.R.

keeping track of what you read using Jupyter notebooks

We read and forget a lot

- Why read it if you are going to forget it?
- Keep **metadata** about what you read.
  - Breadcrumbs to find it again.
- **Known-item search** has become hideously difficult with Web search engines.
  - A wall of spammers hide your item!

Memory hooks

- Many times I don’t remember the title, nor the authors.

- But I remember:
  - Where I read it (physical place)
  - How I read it (physical device)
  - How I found it
  - Approximate date when I read it.

P.A.P.E.R.

- Since 2012, I have been developing a paper management solution cater to my needs.
  - Today I am making available open source.
- It is a set of python scripts that offer many functionalities.
- Can help you get started on your own scripts.

What about alternatives?

- If Zotero or others fulfill your needs, then don’t worry about this tool.
- This tool might make it more time consuming to onboard papers.
- I believe we spend a lot of time reading but too little cataloging what we read.

Basics

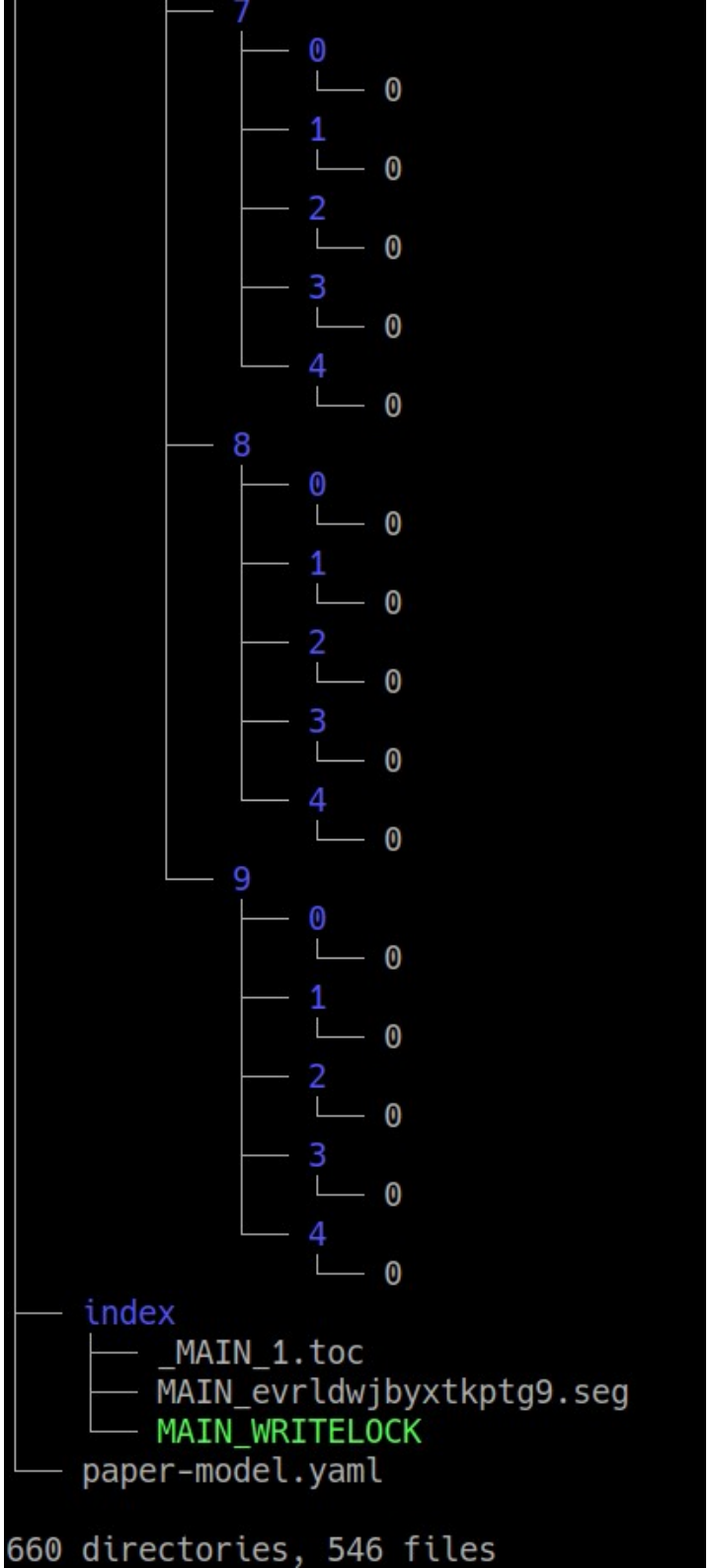
- Data model:
  - A hierarchical attribute-value pair DAG, completely contained in a YAML file.
    - BibTeX file + Mind Map
      - You can define your own relations.
  - Can be opened with any text editor, search inside of it, carry it in your phone, etc.

Basics

- A data model for the 500 papers I have in the tool weights 800kb.
  - About 20% of what I read since 2012.
- If the tool breaks, you still have the data.
- No complex binary upgrade issues as using Dbs.
  - Can commit it to version control.

Basics

- Paper repository:
  - Papers (or other artifacts) can be “absorbed” into the tool to a folder-balanced hash-based file.



Basics

- Paper repository:
  - Downloaded same paper multiple times,
    - With different filenames,
    - Get a canonical name and place in your hard drive.
  - You can use the DAG without the repo.

Exports to a static website

P.A.P.E.R. - Main Index

- Artifact
- Bibtex
- External
- File
- Place
- Reading List
- Relation
- Scenario
- Search
- Topic

Per Type

Artifact

- aaai-2012 - Quantifying the Lexical Overlap Trap in Textual Entailment using a Sentence Compression Of Domain Adaptive Classifiers In Industrial Settings
- ijcai-adaptive-nlp - The Value Of Domain Adaptive Classifiers In Industrial Settings
- mallet - the MALLET software package
- mybook-feat-eng - The Art of Feature Enginering: Essentials for Machine Learning
- mycourse-eci2016-hybrid-ie - ECI Hybrid Information Extraction Systems - Buenos Aires UBA
- mypaper-Iberamia2016-ordering - Using Robustness to Learn to Order Semantic Properties in Referring Expression Generation Pablo Ariel Duboue and Martin Ariel Domínguez -- Iberamia 2016
- mypaper-MICAI2015-robustness - Evaluating Robustness of Referring Expression Generation Algorithms -- Pablo Ariel Duboue ; Martin Ariel Domínguez ; Paula Estrella -- Artificial Intelligence (MICAI), 2015
- mypaper-NAACLHLT2012-referring - On The Feasibility of Open Domain Referring Expression Generation Using Large Scale Folksonomies by Fabián Pacheco, Pablo Ariel Duboue, Martín Ariel Domínguez at NAACLHLT 2012
- mypaper-WebNLG2016-robustness - On the Robustness of Standalone Referring Expression Generation Algorithms Using RDF Data Pablo Ariel Duboue and Martin Ariel Domínguez and Paula Estrella -- WebNLG workshop at INLG 2016
- mypaper-lexical-choice - paper for the lafken series on lexical choice for MT (2016)
- paper-1 - Finding latent code errors via machine learning over program executions Y Brun, MD Ernst - Proceedings of the 26th International Conference on …, 2004 - dl.acm.org
- paper-10 - Agents that learn to explain themselves WL Johnson - Proceedings of the twelfth national conference on …, 1994 - aaai.org
- paper-100 - Private-iye: A framework for privacy preserving data integration - SS Bhowmick, L Gruenwald, M Iwaihara - Data Engineering 2006
- paper-101 - Privacy-preserving history mining for web browsers - M Jakobsson, A Juels, J Ratkiewicz - Proceedings of the Workshop 2008

Website

Artifact: Paper-101

Back to index

type

`artifact`

id

`paper-101`

status

`read-abstract`

text

`Privacy-preserving history mining for web browsers - M Jakobsson, A Juels, J Ratk`

note

`top relevant, web history mining`

related-to

topic-queryfiltering - Filtering personal queries

bibtex

Jakobsson08privacy-preservinghistory - inproceedings

external

external-7 -

found-in

search-11 - on paper citing Agrawal et al (2000), paper-98, search for "query filter"

on-disk

file-106 - pphm.pdf

Backlinks

target^{-1}

- relation-1 - citing

Features

- BibTeX import
- Reading Lists support
- Arbitrary relations between papers
  - Makes use of the “citing” relation
  - Generates a .bib file for you

Generated BibTeX file

```
%  paper-473
@phdthesis{heaton2017automated
  , school={Nova Southeastern University}
  , title={Automated Feature Engineering for Deep Neural Networks with
Genetic Programming}
  , author={Heaton, Jeff}
  , year={2017}
}

%  paper-474
@article{domingos2012few
  , publisher={ACM}
  , author={Domingos, Pedro}
  , journal={Communications of the ACM}
  , title={A few useful things to know about machine learning}
  , number={10}
  , pages={78--87}
  , volume={55}
  , year={2012}
}

%  paper-475
@misc{paper475url
  , title={{COS} 424: {I}nteracting with {D}ata. {P}rinceton CS class 18,
{F}eature {E}ngineering}
  , author={Leon Bottou}
```

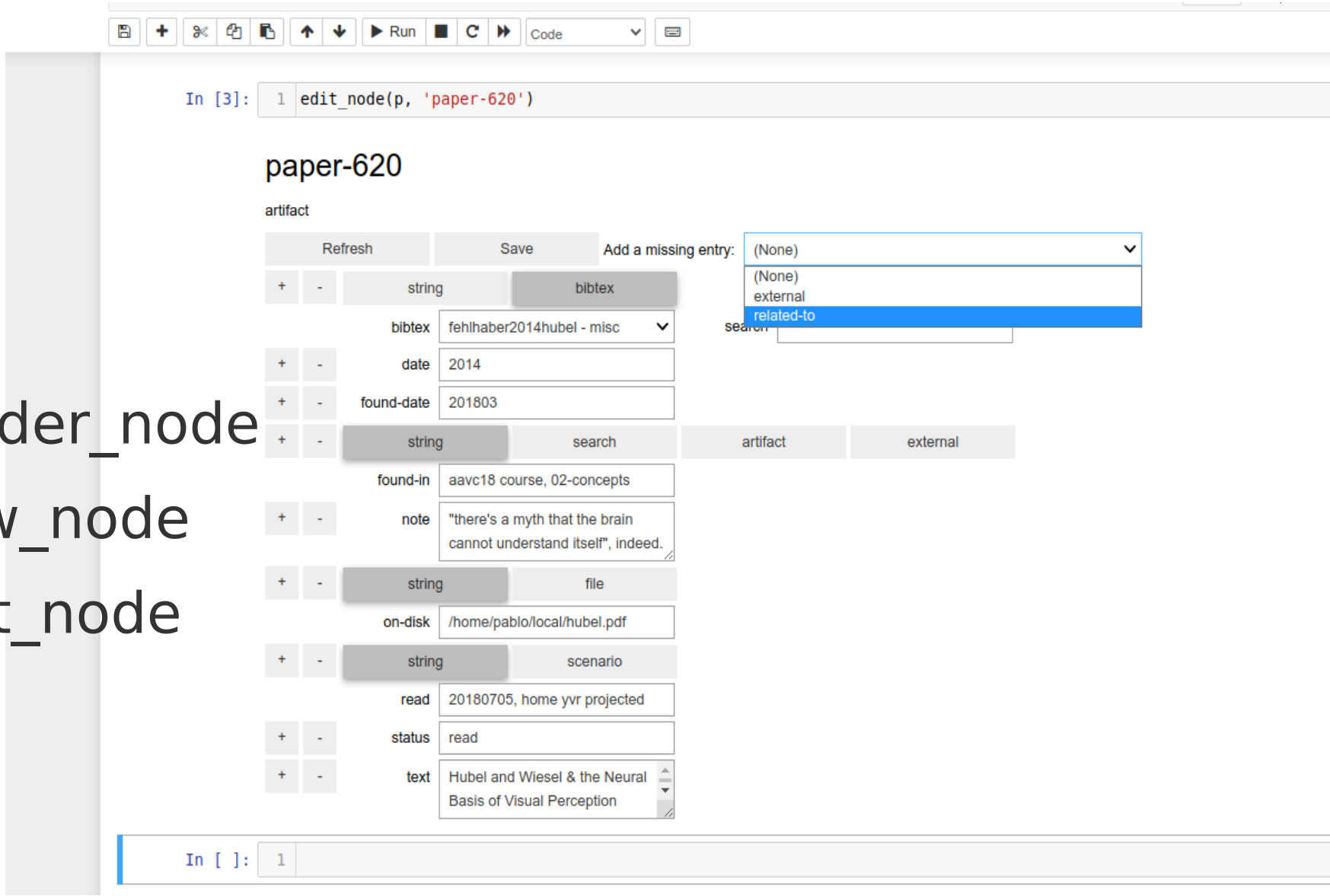
Features

- Command-line UI
  - Import notes with extra fields
    - create entries from plain text

- Custom Jupyter Notebook Widgets

Custom Widgets

- render_node
- new_node
- edit_node



Features

- Full text search engine.
- Query-by-example
  - Given the text of a paper, find matching papers.
- We used it for Ying and Duboue (2019).

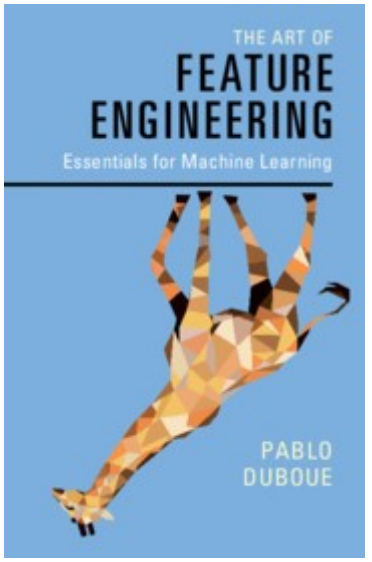
Hackable

- It is a data model and a paper repository.
  - 82 paperapp/code2py.py
  - 165 paperapp/file_repo.py
  - 3 paperapp/__init__.py
  - 255 paperapp/materializer.py
  - 126 paperapp/paper_bibtex.py
  - 31 paperapp/papercli.py
  - 449 paperapp/paper_ipython.py
  - 841 paperapp/paper_repo.py
  - 46 paperapp/paper_yaml.py
  - 140 paperapp/search_index.py
  - 2,138 lines, 10 python files
- You can hook it to external repositories, etc.

Proof is in the pudding

- I use this tool to keep track of the material for the book I just published.
  - The Art of Feature Engineering
    - Cambridge University Press
    - ISBN 978-1108709385

      `http://artoffeatureengineering.com/`



- 103 citations in-tool.
  - 200+ out of tool (oh, well).

What is coming

- Folder with PDFs vs. .bib file sync tool
- Any exciting features you would like to contribute, fork it on GitHub!

  `https://github.com/DrDub/PAPER`
- `Help test the installer:`
  - python3 -m pip install --index-url https://test.pypi.org/simple/ --no-deps paperapp_DrDub
  - pip install paperapp_DrDub[widgets]
  - pip install paperapp_DrDub[fulltext]

Thanks

- This tool started at Les Laboratoires Foulab
  - Montreal’s hackerspace
- Presented Jupyter for quick prototyping at Montréal Python #56.
- The Learn Data Science community for their help with my Feature Engineering book.

`https://github.com/DrDub/PAPER`