VanderBot:

A spreadsheet-based system for creating and updating items in Wikidata

Steve Baskauf
steve.baskauf@vanderbilt.edu
Wikimedia username: Baskaufs

Program for Cooperative Cataloging
Wikidata Pilot monthly meeting
2021-02-18

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

VANDERBILT UNIVERSITY®

Parts of the presentation:

1. Background: motivation, definitions, issues
2. Loading data into Wikidata with VanderBot: theory, workflow, and generalization
3. Maintaining an "authoritative" dataset in Wikidata

This may be the basis for future blog posts

http://baskauf.blogspot.com/

- Initial posts based on rudimentary knowledge:
  - Python and SPARQL queries http://baskauf.blogspot.com/2019/05/getting-data-out-of-wikidata-using.html
  - Python and the Wikidata API http://baskauf.blogspot.com/2019/06/putting-data-into-wikidata-using.html
  - The "how can I use this?" question …
- Series documenting VanderBot, starting with:
  - http://baskauf.blogspot.com/2020/02/vanderbot-python-script-for-writing-to.html
  - and 3 subsequent posts
- Also see VanderBot repo landing page: http://vanderbi.lt/vanderbot

Background:
motivation, definitions, issues

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

VANDERBILT UNIVERSITY®

Why are people interested in Wikidata?

- Easy-to-use user interface for community contributions
- Strong community and high profile in knowledge graph community
- Interactivity with Wikidata and Wikimedia Commons
- "identifier central" (important for this group!)
- Linked Open Data (LOD) capabilities (e.g. SPARQL queries)

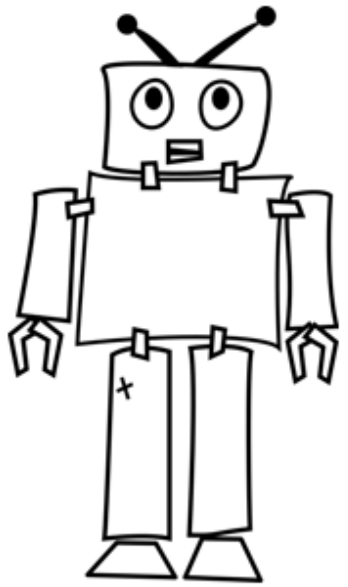
Grab-bag of issues related to the VanderBot approach

- Do your data belong in Wikidata?
  - notability
  - scalability (vs. Wikibase)
  - linkability (would somebody ever link something to this?)
- Scalability of CSV2RDF (limits maybe 1M triples, 10 000 row CSV)
- Establishing the limits of the graph model (to fit "flat" spreadsheet)
- Practical matters of building a schema for your data consistent with the approach
- Importance of references

"Authoritative" data into Wikidata

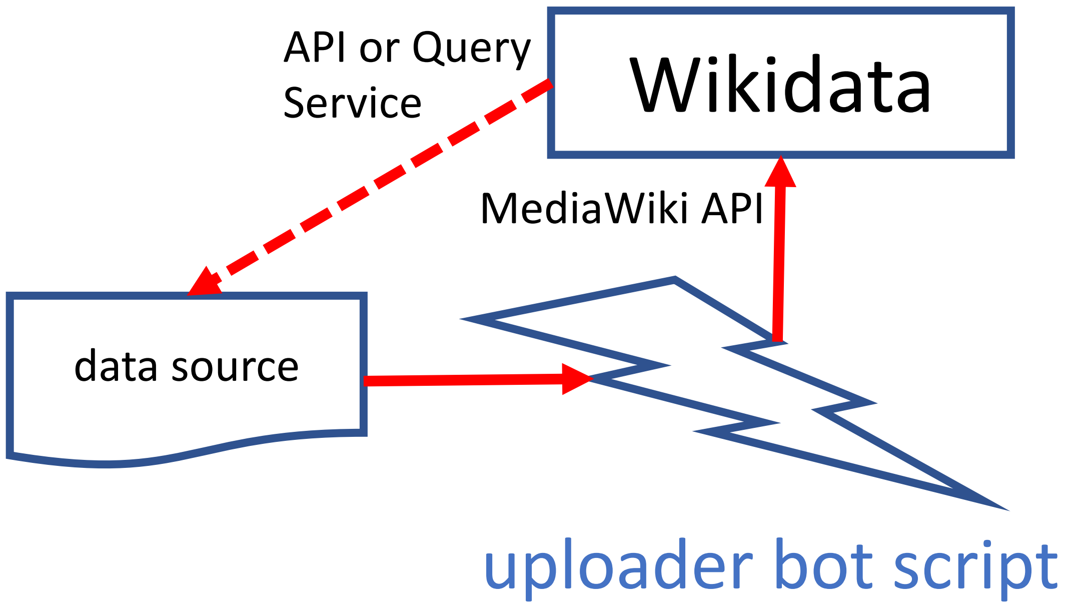
- Need to decide on data model/properties to be managed
  - Existing WikiProject with properties?
  - Query for properties of same type (e.g. what properties are used for coins?)
- Evaluate the current state of data in Wikidata
  - What fraction of possible items are already there?
  - Do all properties needed for the model exist?
  - Is there a value of a property (or property chain) that will link all items of interest?
- Evaluate "authoritative" data source for ease of linking
  - clean data
  - standardized strings
  - unique identifiers

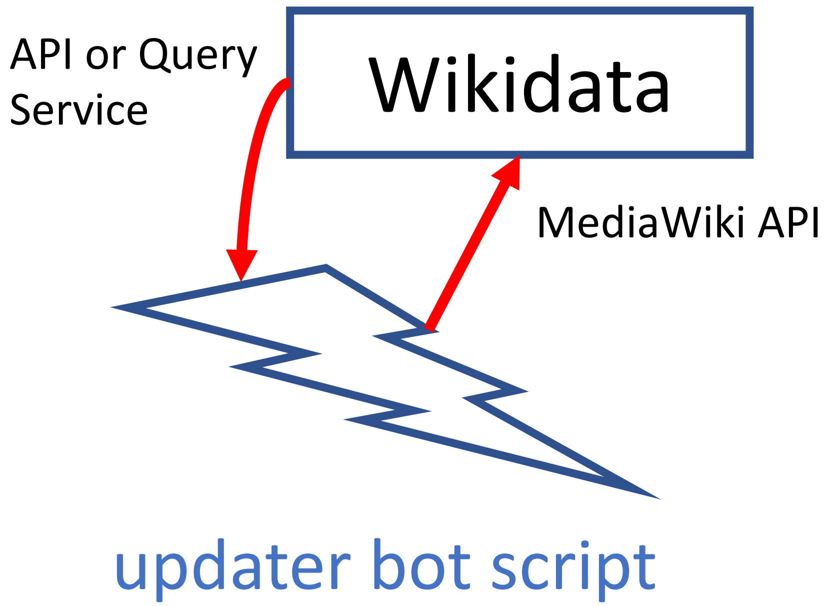
What's a bot?



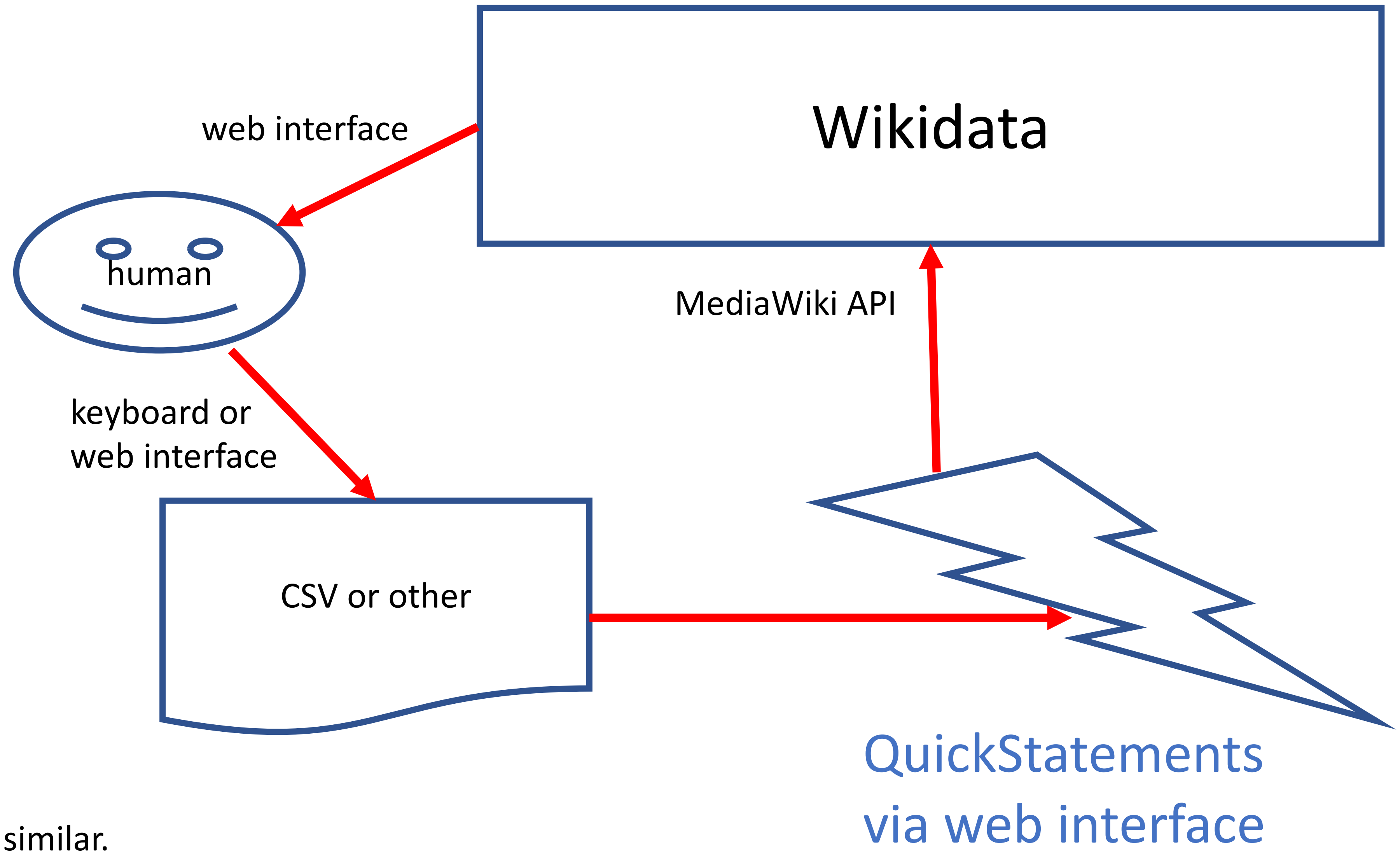
- Software that can read and write to Wikidata via an API (application programming interface)
  - autonomous (no human intervention)
  - non-autonomous (human monitors and may intervene)

"dumb" (autonomous) bot overview



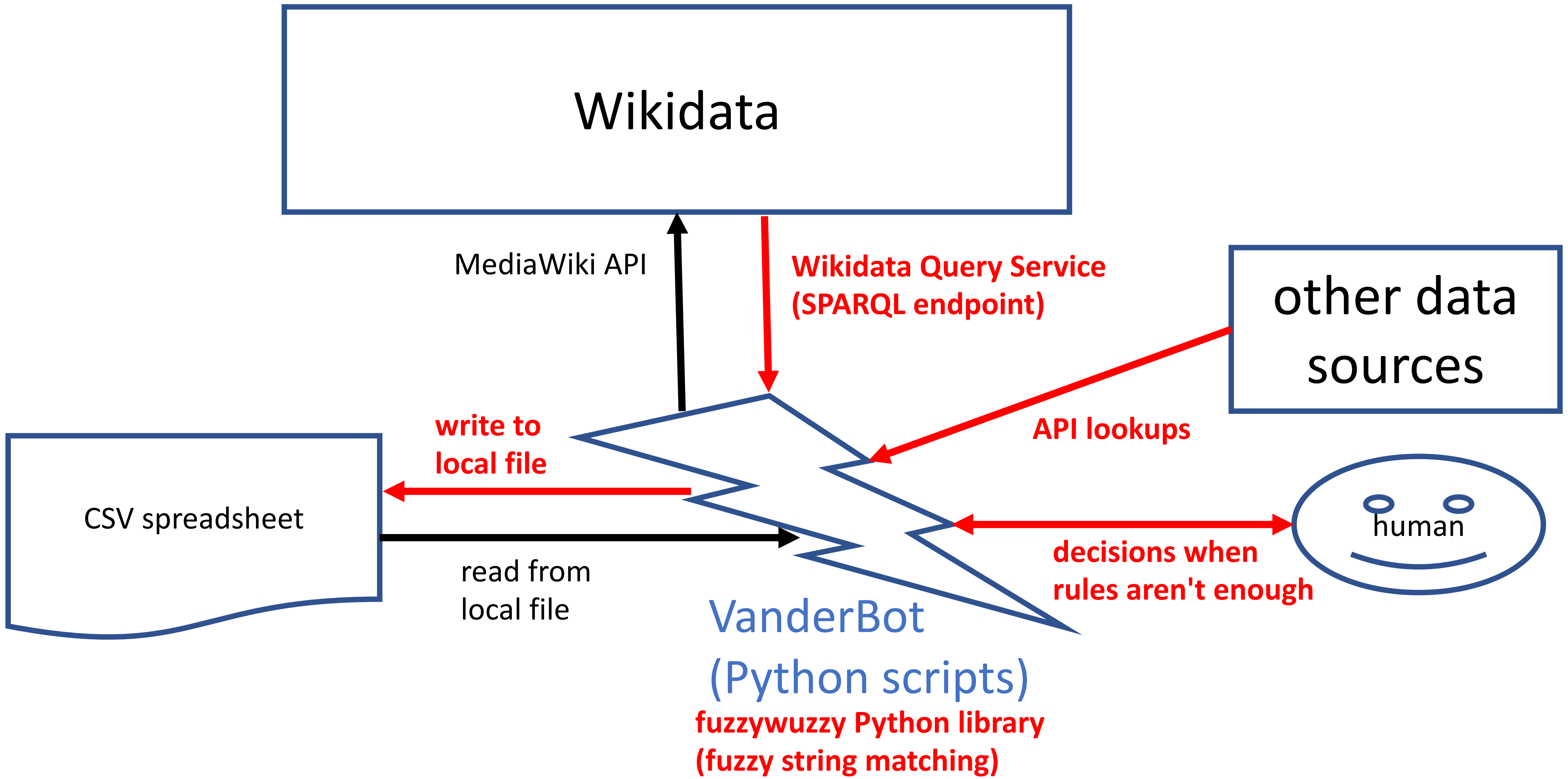


QuickStatements



OpenRefine is similar.

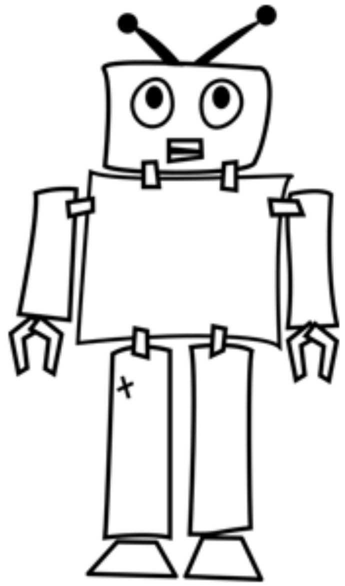
VanderBot-human team overview

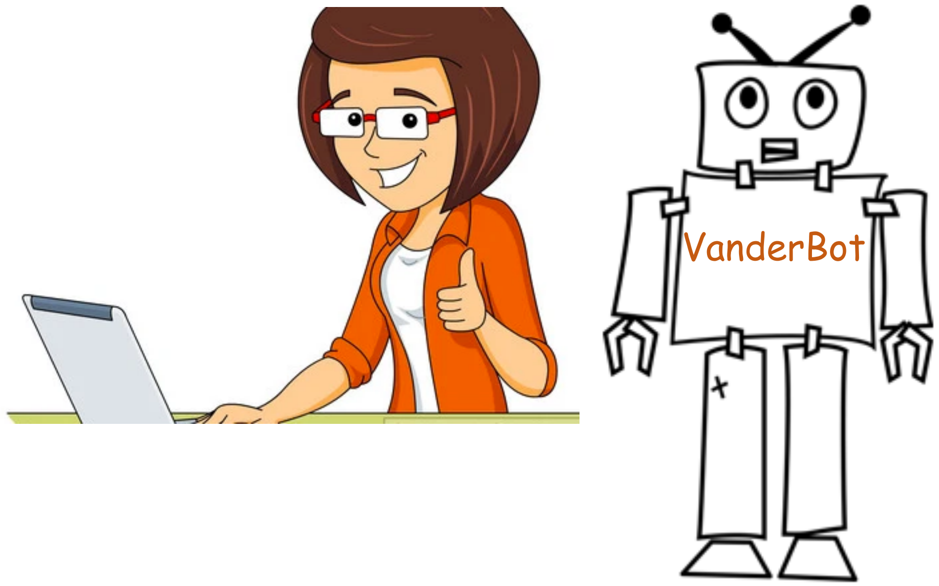


Human editors

"dumb" bot

bot-human team





Greg Weldy (our star editor) averages **160 edits per day**

Wikidata bot without a bot flag: max **3000 edits per hour**

Create/edit about **200 items per hour** **(vs. 40 edits)**

high quality,
low speed

low quality,
high speed

optimizes quality
and speed

Public domain? https://www.iconspng.com/image/96600/exhausted-computer-user

https://images.app.goo.gl/i86cVyfwUxMb3Wr7A

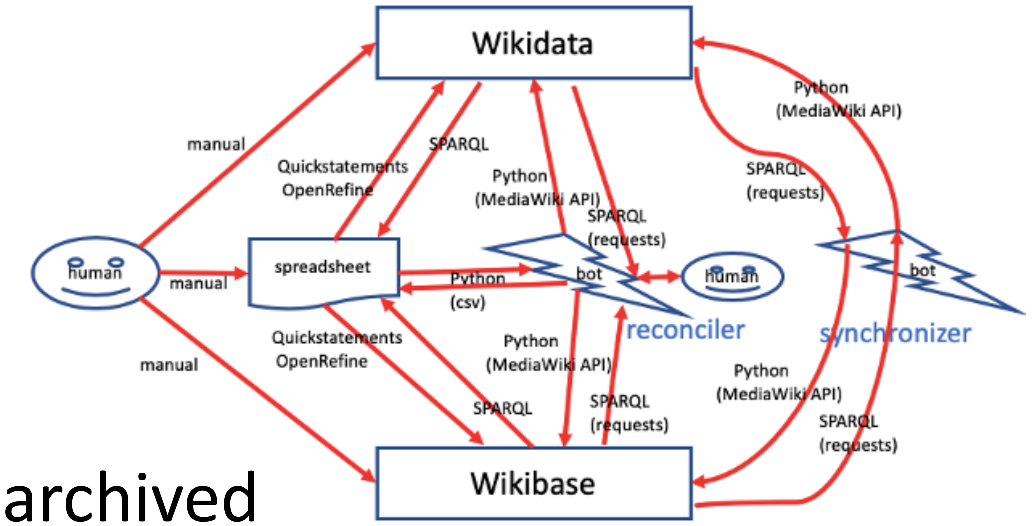
Loading data into Wikidata with VanderBot: theory, workflow, and generalization

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

VANDERBILT UNIVERSITY®

Origins of the project



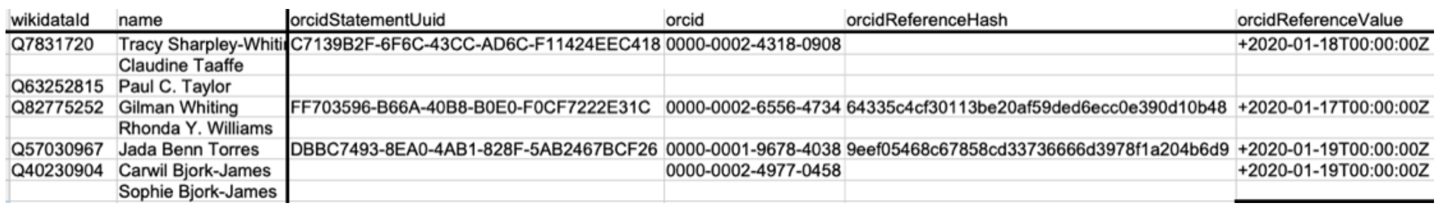
- Thinking about how Wikidata data could be archived locally (original focus on Wikibase)
- Previous (nonstandard) attempts to turn CSVs into RDF (Guid-O-Matic https://github.com/baskaufs/guid-o-matic)
- Inability to understand and use Pywikibot and Wikidataintegrator
- Frustration with labor-intensiveness and inability to script OpenRefine and QuickStatements

the Guid-O-Matic squid

W3C Recommendation: Generating RDF from Tabular Data on the Web

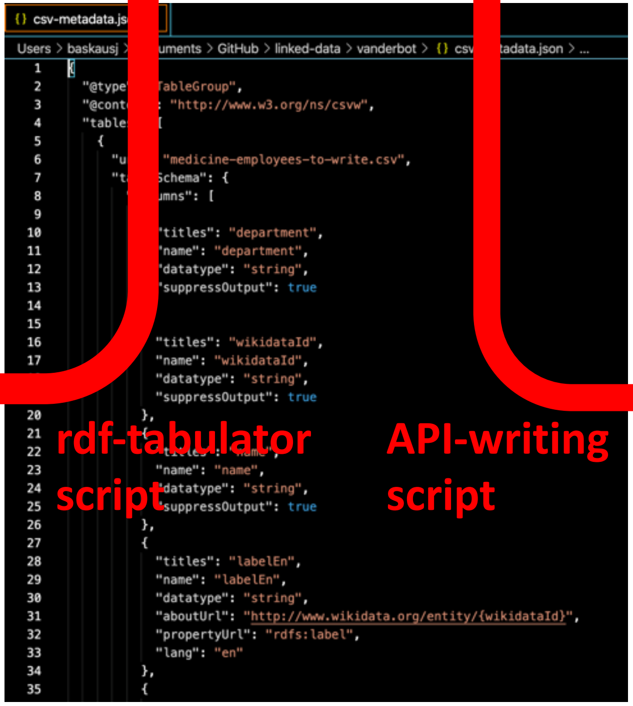
- International standard to relate CSV tables to RDF
- https://www.w3.org/TR/csv2rdf/
- Approach: use a JSON-LD schema to map the table columns to the Wikibase graph model
- Applications:
  - VanderBot API-writing script knows how to convert CSV data to Wikibase-based JSON required by Wikidata API
  - Applying the schema to the CSV data will emit exactly the same RDF as is queried by the Wikidata Query Service
- Advantage: stable, standard way to unambiguously archive data in Wikidata in an easy-to-read tabular form, implementation independent.

Generating a knowledge graph (RDF) from a CSV



wikidataId	name	orcidStatementUuid	orcid	orcidReferenceHash	orcidReferenceValue
Q7831720	Tracy Sharpley-Whiti	C7139B2F-6F6C-43CC-AD6C-F11424EEC418	0000-0002-4318-0908		+2020-01-18T00:00:00Z
	Claudine Taaffe				
Q63252815	Paul C. Taylor				
Q82775252	Gilman Whiting	FF703596-B66A-40B8-B0E0-F0CF7222E31C	0000-0002-6556-4734	64335c4cf30113be20af59ded6ecc0e390d10b48	+2020-01-17T00:00:00Z
	Rhonda Y. Williams				
Q57030967	Jada Benn Torres	DBBC7493-8EA0-4AB1-828F-5AB2467BCF26	0000-0001-9678-4038	9eef05468c67858cd33736666d3978f1a204b6d9	+2020-01-19T00:00:00Z
Q40230904	Carwil Bjork-James		0000-0002-4977-0458		+2020-01-19T00:00:00Z
	Sophie Bjork-James				

spreadsheet (actively maintained or archived)



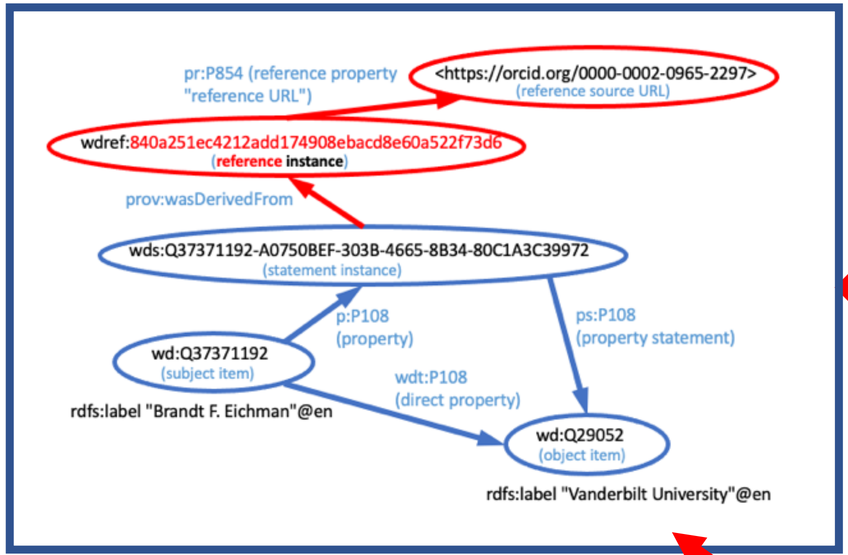
csv-metadata.js

Users > baskausj > uments > GitHub > linked-data > vanderbot > csv tadata.json > ...

```
{
  "@type  TableGroup",
  "@cont  : "http://www.w3.org/ns/csvw",
  "table    [
    {
      "u      "medicine-employees-to-write.csv",
      "t      Schema": {
              umns": [

              "titles": "department",
              "name": "department",
              "datatype": "string",
              "suppressOutput": true


             "titles": "wikidataId",
             "name": "wikidataId",
             "datatype": "string",
             "suppressOutput": true
           },
           {
             "titles": "name",
             "name": "name",
             "datatype": "string",
             "suppressOutput": true
           },
           {
             "titles": "labelEn",
             "name": "labelEn",
             "datatype": "string",
             "aboutUrl": "http://www.wikidata.org/entity/{wikidataId}",
             "propertyUrl": "rdfs:label",
             "lang": "en"
           },
           {
```

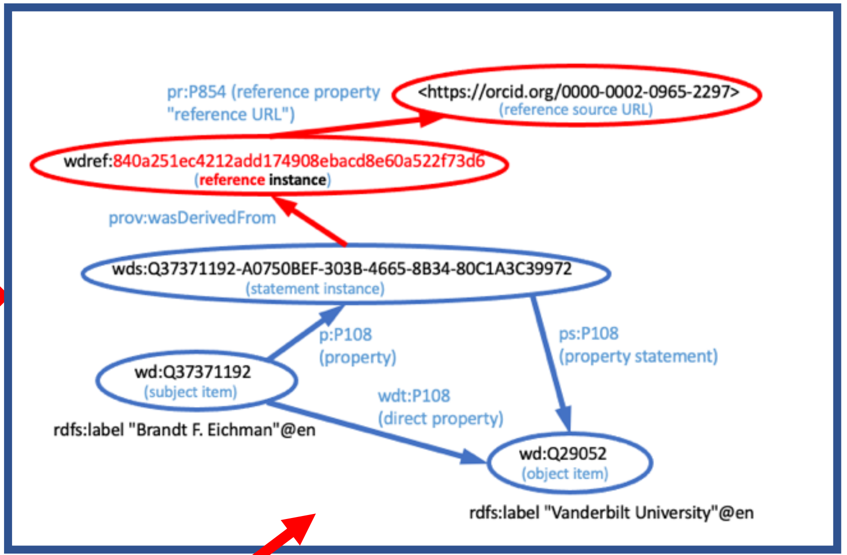
rdf-tabulator script

API-writing script



pr:P854 (reference property "reference URL")
<https://orcid.org/0000-0002-0965-2297>
(reference source URL)
wdref:840a251ec4212add174908ebacd8e60a522f73d6
(reference instance)
prov:wasDerivedFrom
wds:Q37371192-A0750BEF-303B-4665-8B34-80C1A3C39972
(statement instance)
p:P108
(property)
ps:P108
(property statement)
wd:Q37371192
(subject item)
wdt:P108
(direct property)
rdfs:label "Brandt F. Eichman"@en
wd:Q29052
(object item)
rdfs:label "Vanderbilt University"@en

local triplestore (e.g. Fuseki)



pr:P854 (reference property "reference URL")
<https://orcid.org/0000-0002-0965-2297>
(reference source URL)
wdref:840a251ec4212add174908ebacd8e60a522f73d6
(reference instance)
prov:wasDerivedFrom
wds:Q37371192-A0750BEF-303B-4665-8B34-80C1A3C39972
(statement instance)
p:P108
(property)
ps:P108
(property statement)
wd:Q37371192
(subject item)
wdt:P108
(direct property)
rdfs:label "Brandt F. Eichman"@en
wd:Q29052
(object item)
rdfs:label "Vanderbilt University"@en

Wikidata (now)

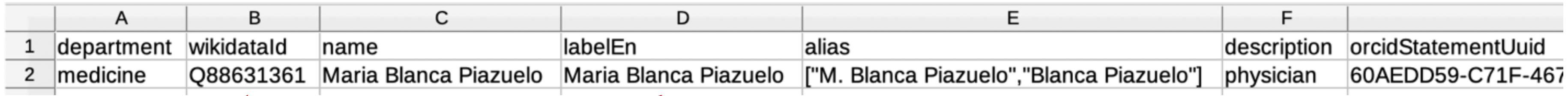
compare by federated SPARQL query

Advantages

- Humans can review the spreadsheet to get a quick overview of the state of things (absence of references, frequency and diversity of values, label and description quality)
- Humans can change the spreadsheet to make many changes with few button clicks (copy and paste the same reference into hundreds of cells, quick cleanup of names and descriptions)
- The spreadsheet (along with mapping schema) is a snapshot of part of the Wikidata graph at a moment in time; can be reconstituted as RDF in a triplestore.
- Using Python scripts (vs. OpenRefine, Quickstatements) to have the potential to automate steps more fully.

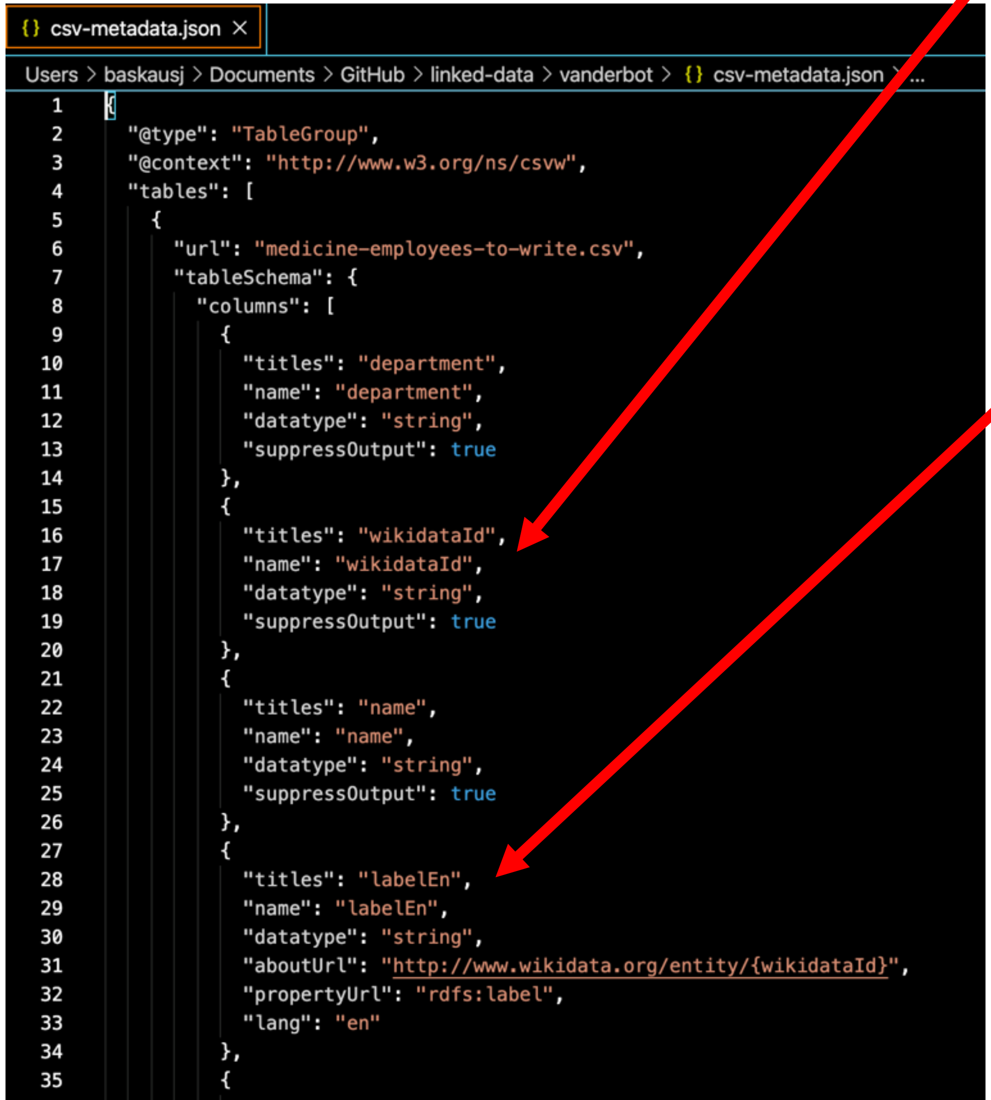
Example mapping

JSON mapping file



	A	B	C	D	E	F	
1	department	wikidataId	name	labelEn	alias	description	orcidStatementUuid
2	medicine	Q88631361	Maria Blanca Piazuelo	Maria Blanca Piazuelo	["M. Blanca Piazuelo","Blanca Piazuelo"]	physician	60AEDD59-C71F-467

CSC data file



{} csv-metadata.json ×
Users > baskausj > Documents > GitHub > linked-data > vanderbot > {} csv-metadata.json > ...

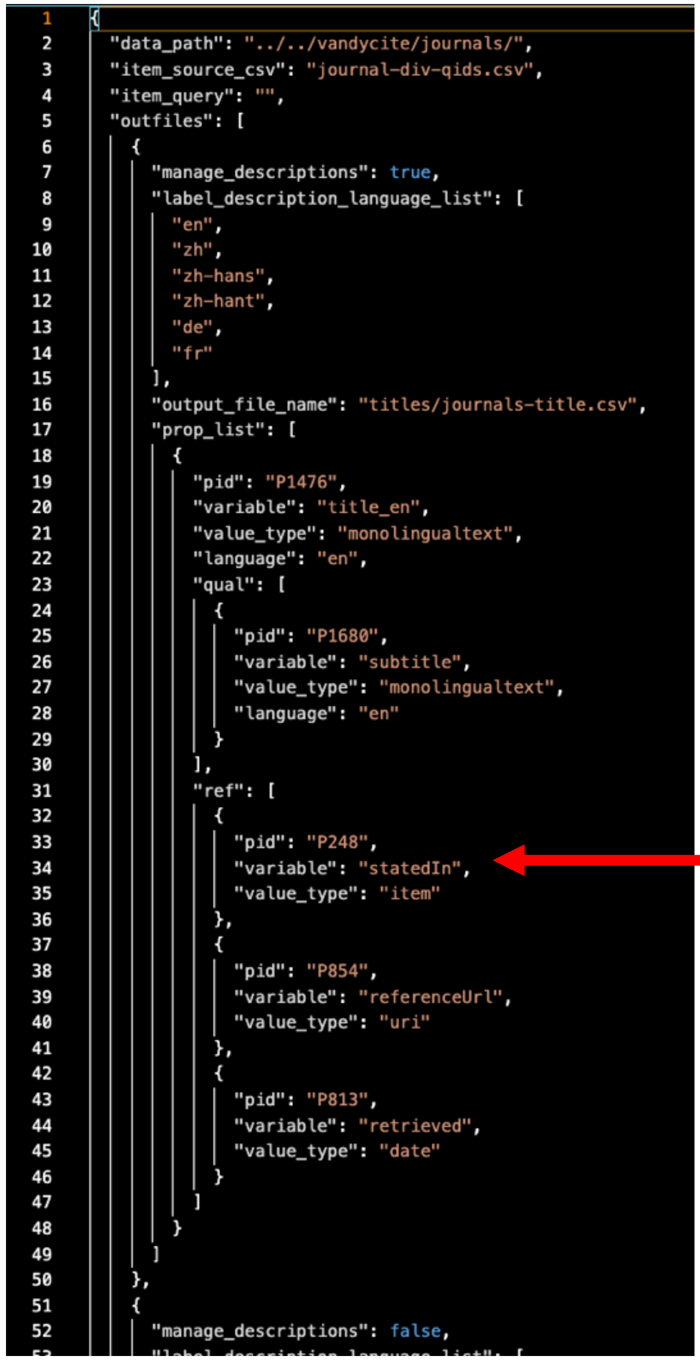
```
{
  "@type": "TableGroup",
  "@context": "http://www.w3.org/ns/csvw",
  "tables": [
    {
      "url": "medicine-employees-to-write.csv",
      "tableSchema": {
        "columns": [
          {
            "titles": "department",
            "name": "department",
            "datatype": "string",
            "suppressOutput": true
          },
          {
            "titles": "wikidataId",
            "name": "wikidataId",
            "datatype": "string",
            "suppressOutput": true
          },
          {
            "titles": "name",
            "name": "name",
            "datatype": "string",
            "suppressOutput": true
          },
          {
            "titles": "labelEn",
            "name": "labelEn",
            "datatype": "string",
            "aboutUrl": "http://www.wikidata.org/entity/{wikidataId}",
            "propertyUrl": "rdfs:label",
            "lang": "en"
          },
          {
```

The JSON mapping schema makes it possible to "understand" what the table means in terms of the Wikibase model (specifically Wikidata).

```
<http://www.wikidata.org/entity/Q88631361> rdfs:label "Maria Blanca Piazuelo"@en;
  schema:description "physician"@en;
  <http://www.wikidata.org/prop/P496> <http://www.wikidata.org/entity/statement/60AEDD59-C71F-467E-8A23-E6550C3070EC>;
  <http://www.wikidata.org/prop/direct/P496> "0000-0002-0000-1324" .

<http://www.wikidata.org/entity/statement/60AEDD59-C71F-467E-8A23-E6550C3070EC> prov:wasDerivedFrom
<http://www.wikidata.org/reference/7f81f24a4148728c2eb8e030140bb2808f105296> .
```

emitted RDF/Turtle
(same as from Wikidata Query Service)

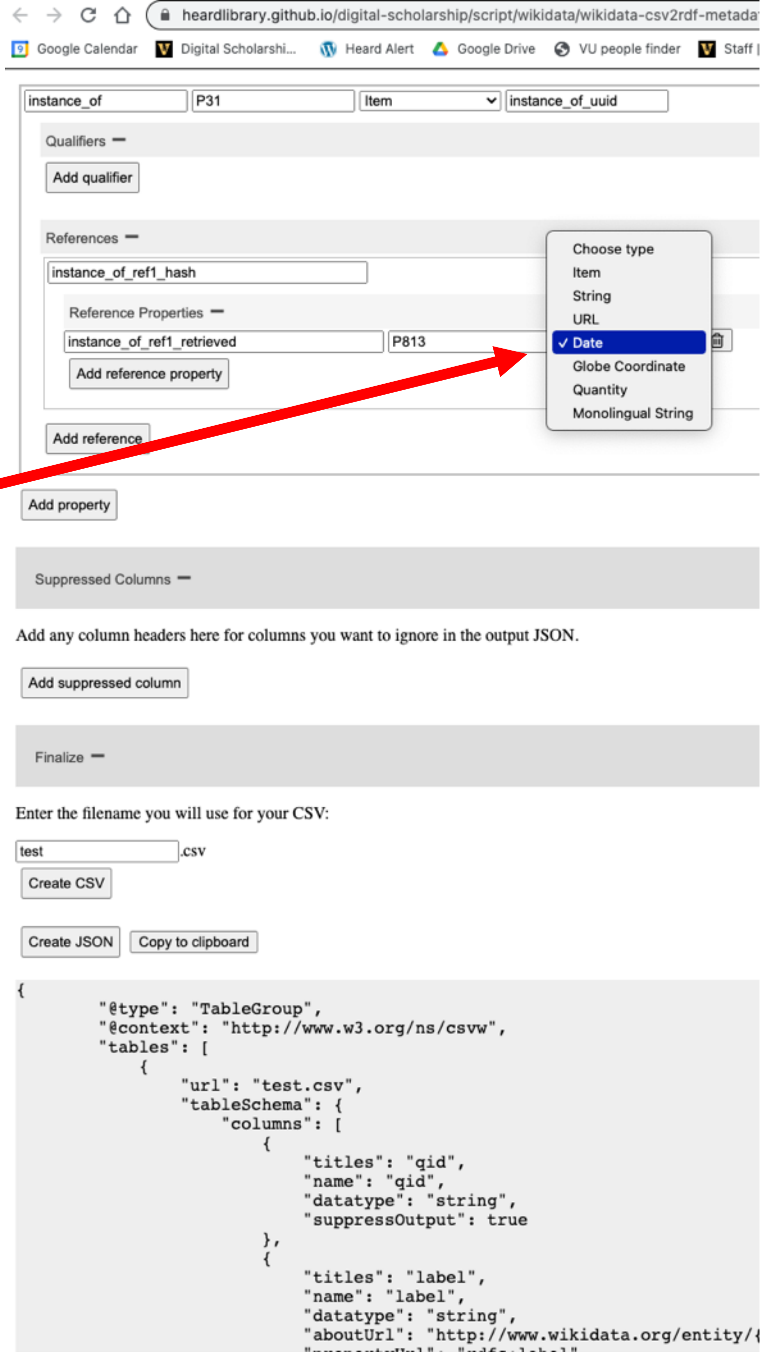
Generating the table schema



```
{
  "data_path": "../../vandycite/journals/",
  "item_source_csv": "journal-div-qids.csv",
  "item_query": "",
  "outfiles": [
    {
      "manage_descriptions": true,
      "label_description_language_list": [
        "en",
        "zh",
        "zh-hans",
        "zh-hant",
        "de",
        "fr"
      ],
      "output_file_name": "titles/journals-title.csv",
      "prop_list": [
        {
          "pid": "P1476",
          "variable": "title_en",
          "value_type": "monolingualtext",
          "language": "en",
          "qual": [
            {
              "pid": "P1680",
              "variable": "subtitle",
              "value_type": "monolingualtext",
              "language": "en"
            }
          ],
          "ref": [
            {
              "pid": "P248",
              "variable": "statedIn",
              "value_type": "item"
            },
            {
              "pid": "P854",
              "variable": "referenceUrl",
              "value_type": "uri"
            },
            {
              "pid": "P813",
              "variable": "retrieved",
              "value_type": "date"
            }
          ]
        }
      ]
    },
    {
      "manage_descriptions": false,
```

From simplified bespoke JSON (more restricted)

Using a web GUI (mostly unrestricted)



heardlibrary.github.io/digital-scholarship/script/wikidata/wikidata-csv2rdf-metada

instance_of	P31	Item	instance_of_uuid

Qualifiers —

References —

instance_of_ref1_hash

Reference Properties —

instance_of_ref1_retrieved	P813

Suppressed Columns —

Add any column headers here for columns you want to ignore in the output JSON.

Finalize —

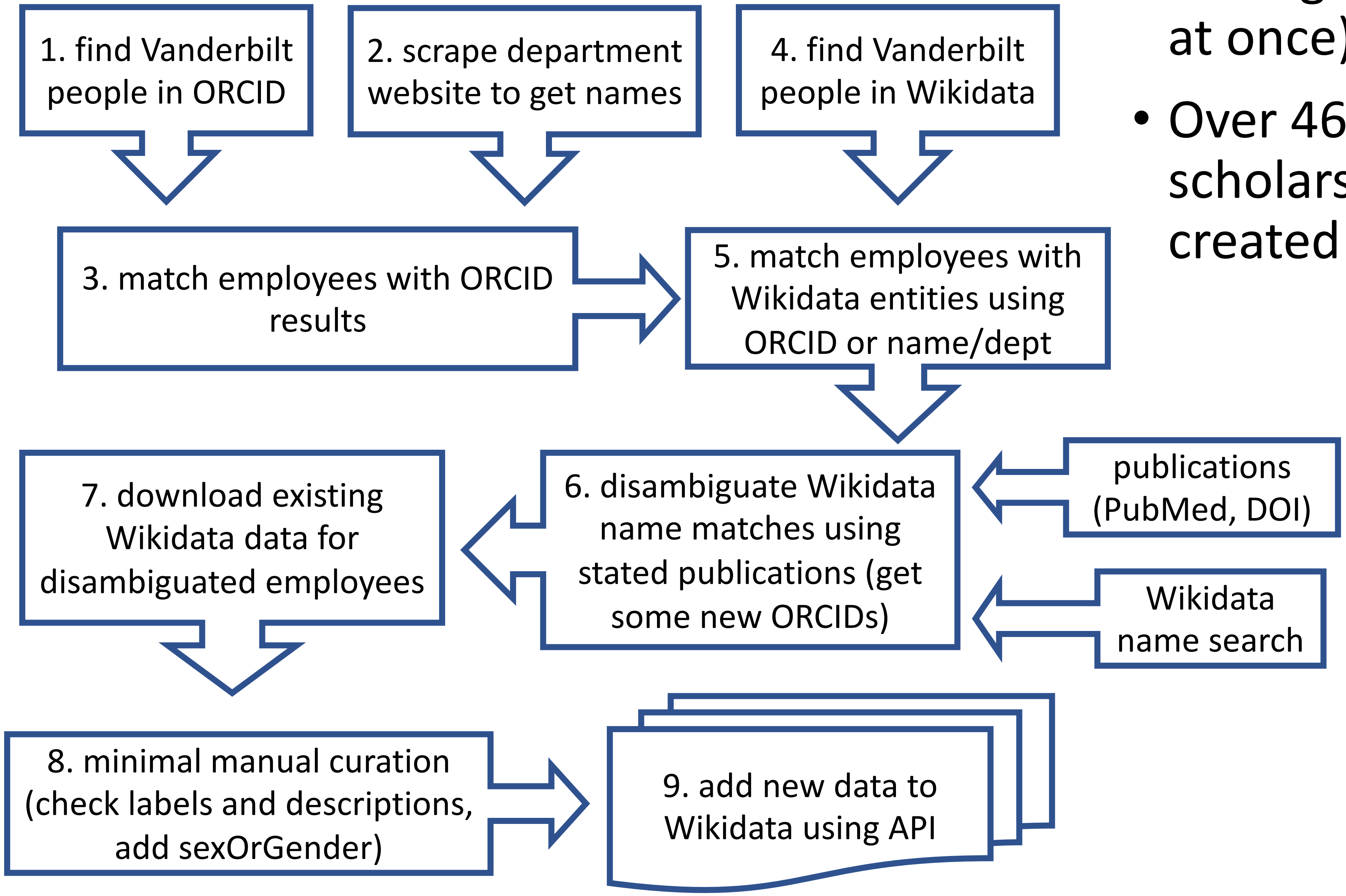
Enter the filename you will use for your CSV:

test.csv

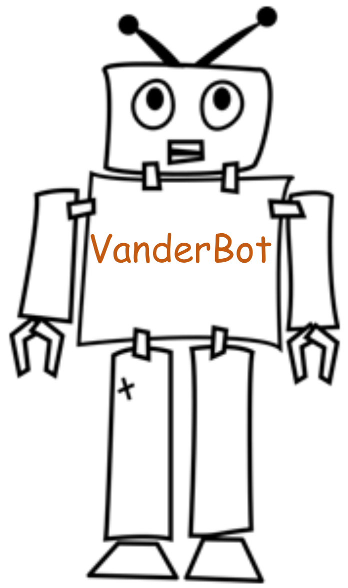
```
{
        "@type": "TableGroup",
        "@context": "http://www.w3.org/ns/csvw",
        "tables": [
            {
                "url": "test.csv",
                "tableSchema": {
                    "columns": [
                        {
                            "titles": "qid",
                            "name": "qid",
                            "datatype": "string",
                            "suppressOutput": true
                        },
                        {
                            "titles": "label",
                            "name": "label",
                            "datatype": "string",
                            "aboutUrl": "http://www.wikidata.org/entity/{
```

https://heardlibrary.github.io/digital-scholarship/script/wikidata/wikidata-csv2rdf-metadata.html

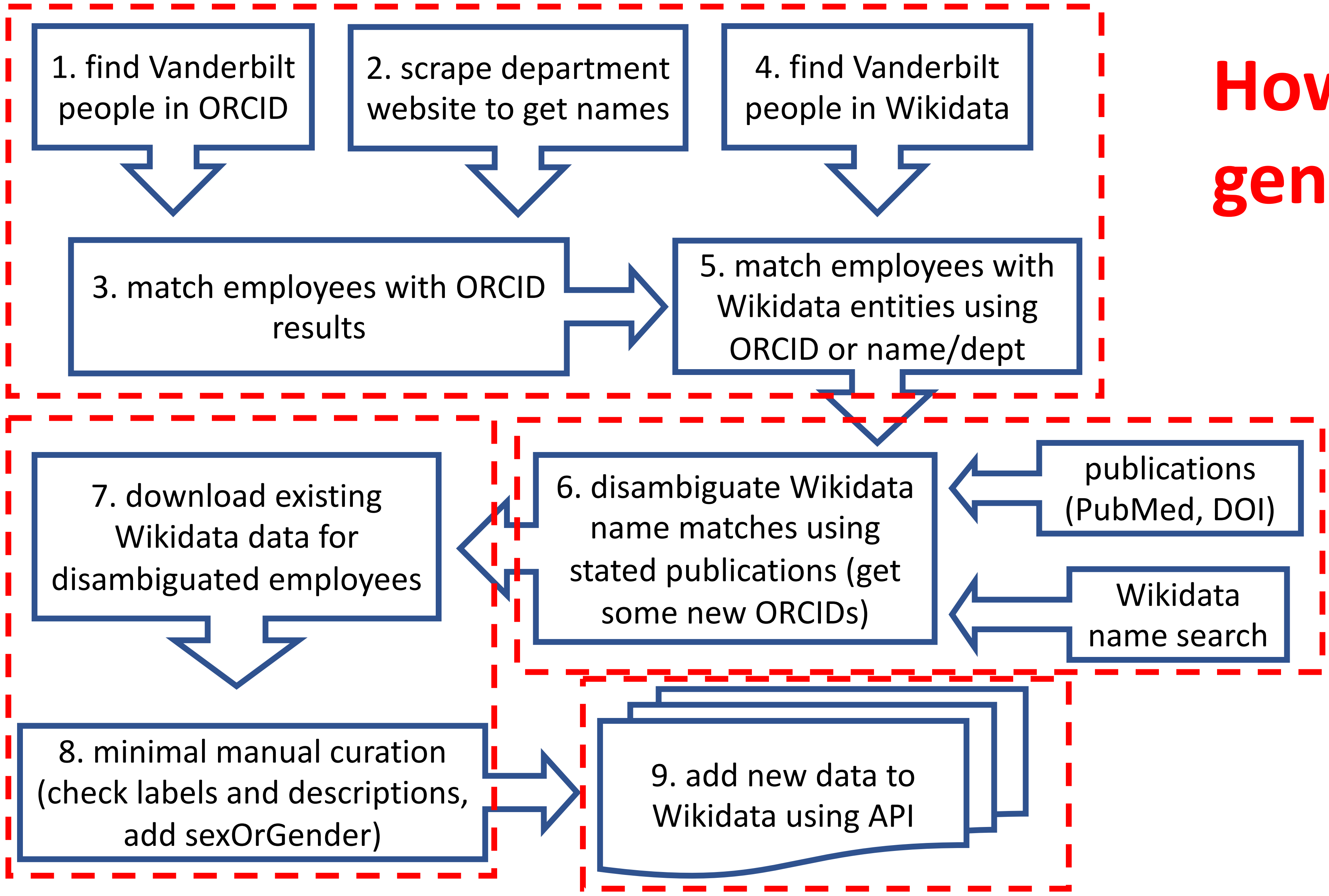
VanderBot workflow (researchers)


1. find Vanderbilt people in ORCID
2. scrape department website to get names
4. find Vanderbilt people in Wikidata
3. match employees with ORCID results
5. match employees with Wikidata entities using ORCID or name/dept
publications (PubMed, DOI)
Wikidata name search
6. disambiguate Wikidata name matches using stated publications (get some new ORCIDs)
7. download existing Wikidata data for disambiguated employees
8. minimal manual curation (check labels and descriptions, add sexOrGender)
9. add new data to Wikidata using API

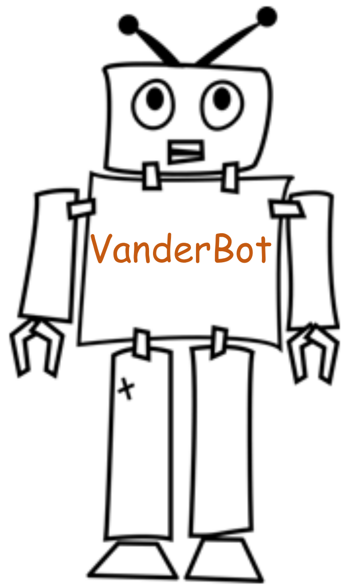
- Over 8000 item edits (most editing multiple statements at once)
- Over 4600 Vanderbilt scholars/researcher items created or linked



VanderBot workflow


1. find Vanderbilt people in ORCID
2. scrape department website to get names
4. find Vanderbilt people in Wikidata
3. match employees with ORCID results
5. match employees with Wikidata entities using ORCID or name/dept
7. download existing Wikidata data for disambiguated employees
6. disambiguate Wikidata name matches using stated publications (get some new ORCIDs)
publications (PubMed, DOI)
Wikidata name search
8. minimal manual curation (check labels and descriptions, add sexOrGender)
9. add new data to Wikidata using API

How can this be generalized?



Generalized workflow

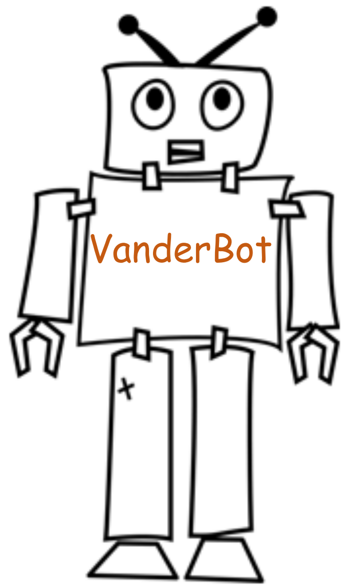
Assemble authoritative source data (idiosyncratic)

Disambiguate against existing items

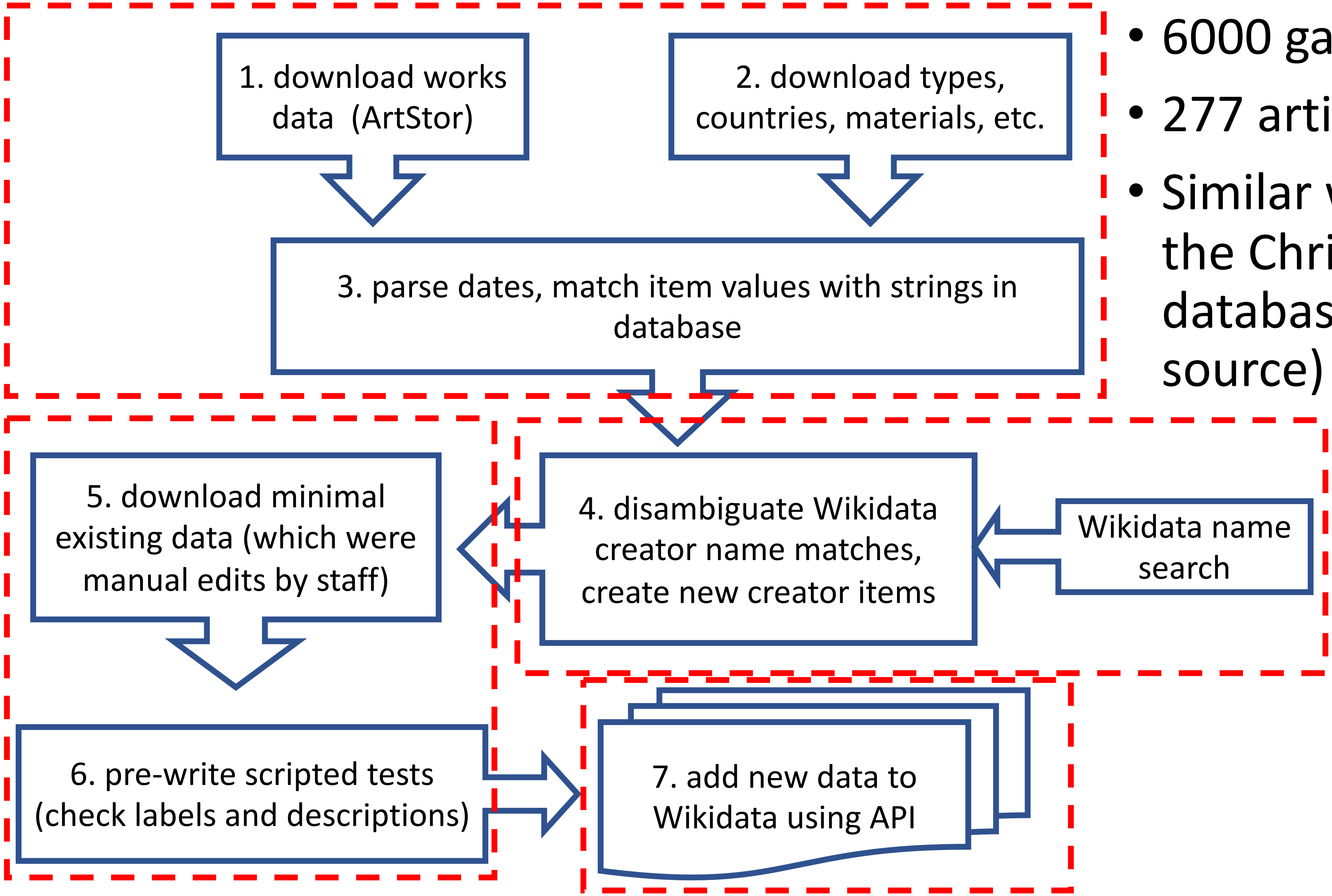
Merge existing item data (SPARQL) with source data

Add new data via API

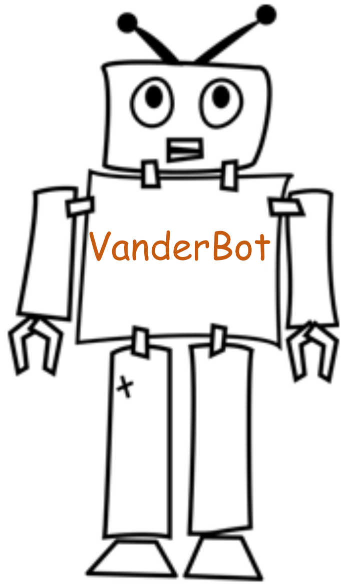
How can this be generalized?



VanderBot workflow (Fine Arts Gallery)


1. download works data (ArtStor)
2. download types, countries, materials, etc.
3. parse dates, match item values with strings in database
4. disambiguate Wikidata creator name matches, create new creator items
Wikidata name search
5. download minimal existing data (which were manual edits by staff)
6. pre-write scripted tests (check labels and descriptions)
7. add new data to Wikidata using API

- 6000 gallery items created
- 277 artist items created
- Similar workflow for Art in the Christian Tradition (ACT) database (different data source)



How can we keep track of existing items (of our "authoritative" interest)?

1. **Keep a list of Q IDs**
2. Use identifying property (or path) and value

Example:

```
SELECT DISTINCT ?qid ?label
WHERE {
VALUES ?qid
    {
    wd:Q102305506
    wd:Q102315563
    wd:Q102315787
    wd:Q102949359
    }
?qid rdfs:label ?label.
}
```

How can we keep track of existing items?

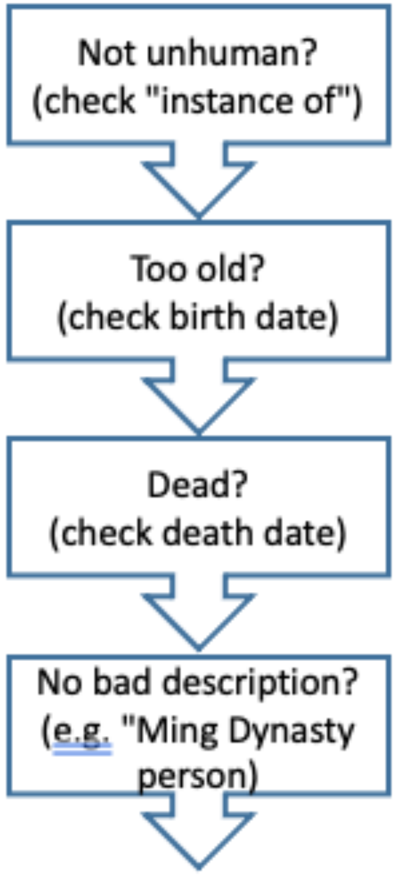
1. Keep a list of Q IDs
2. **Use identifying property (or path) and value in triple pattern**

Examples:

- `?work wdt:P195 wd:Q18563658.` (collection Vanderbilt U. Fine Arts Gallery)
- `?researcher wdt:P1416/wdt:P749 wd:Q29052.` (affiliated with dept., parent organization Vanderbilt)
- `?image wdt:P9092 ?id.` (Art in the Christian Tradition identifier)

How to disambiguate strings against existing items?

- Generate a list of variant string forms (e.g. initials in names)
- Screen Wikidata SPARQL hits against criteria (human, not dead, born after…)
- Check hits against linked sources (PubMed, CrossRef, ULAN)
- Compare hits against existing items using fuzzy string matching.
  - Robust against small differences (periods, capitalization, diacritics)
  - Must test to determine best matching algorithm and cutoff score.
  - Not perfect (Bob vs. Robert, language variants)
  - Must accept the fact that mistakes will be made, but minimize.
- Present multiple possible matches for human decision.
- **Only partly generalizable, must experiment for each use case.**



Software development

Assemble authoritative source data (idiosyncratic)

Download existing data

`acquire_wikidata_metadata.py`

Build csv-metadata schema

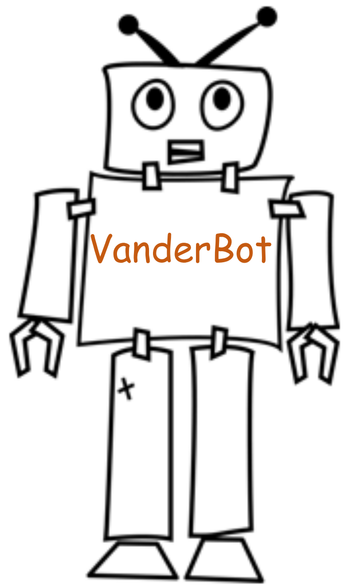
`convert_json_to_metadata_sch`
`ema.py`

Disambiguate against existing items

`vb3_match_wikidata.py`
`process_gallery.ipynb`

Add new data via API

`vb6_upload_wikidata.py`



Maintaining an "authoritative" dataset in Wikidata

DiSC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

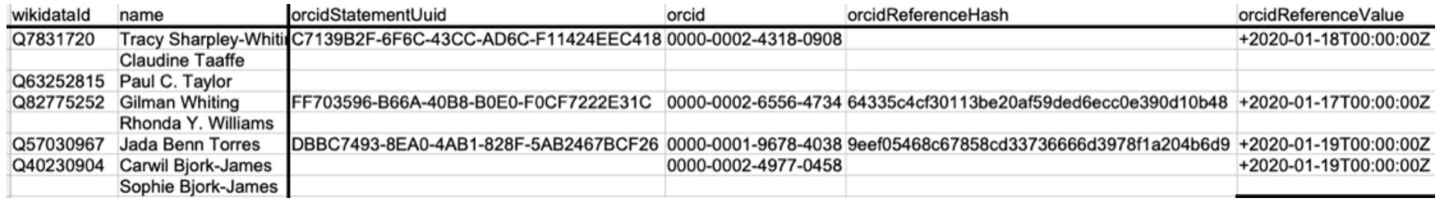
Jean & Alexander Heard LIBRARIES

VANDERBILT UNIVERSITY®

Maintaining existing authoritative data

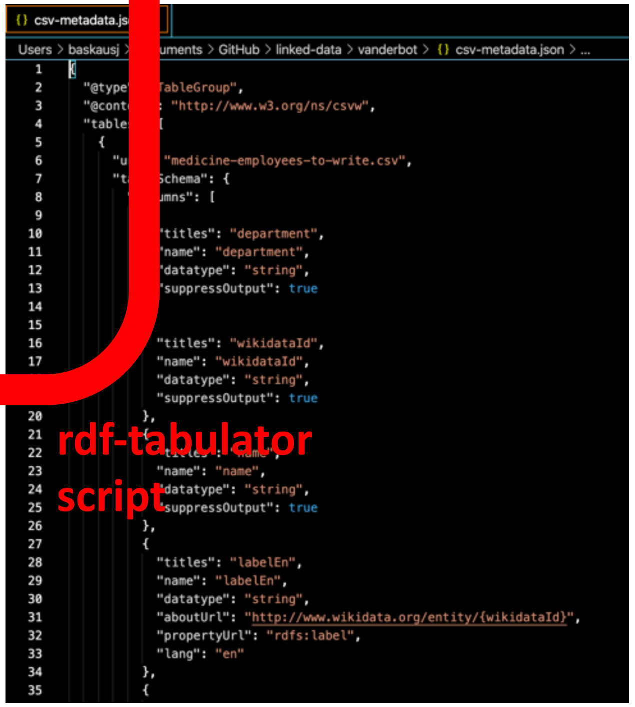
- "Authoritative data" is really an offense to the spirit of Wikidata, but…
- How do we detect changes from our authoritative data?
- How do we decide whether the changes are:
  - "good" (community contributions) that should be pulled into our local dataset?
  - "bad" (vandalism) that should be deleted/changed?
  - information outside the scope of our interest that should be ignored?
- How do we implement the transfer of new data?
  - Should a human intervene in every change?
  - What degree of automation is "safe"?
  - Should human approve changes in advance or review changes after the fact?

Checking for changes over time



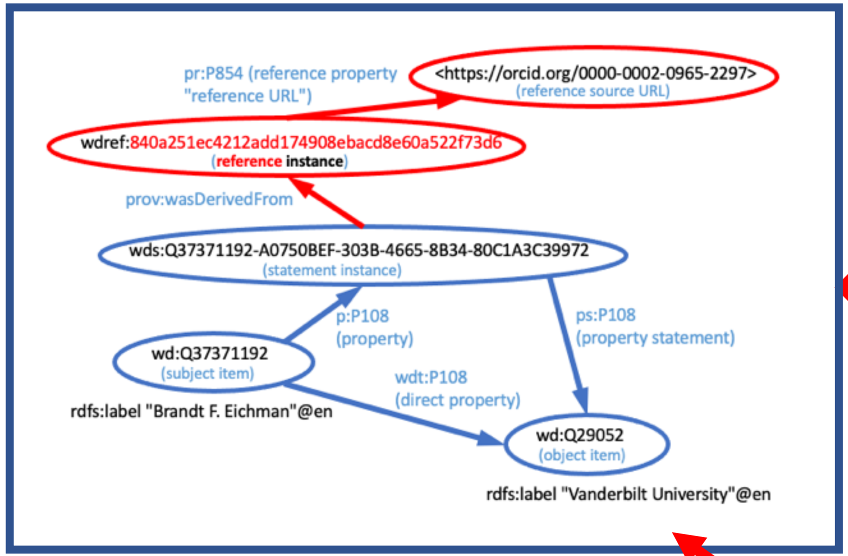
wikidataId	name	orcidStatementUuid	orcid	orcidReferenceHash	orcidReferenceValue
Q7831720	Tracy Sharpley-Whiti	C7139B2F-6F6C-43CC-AD6C-F11424EEC418	0000-0002-4318-0908		+2020-01-18T00:00:00Z
	Claudine Taaffe				
Q63252815	Paul C. Taylor				
Q82775252	Gilman Whiting	FF703596-B66A-40B8-B0E0-F0CF7222E31C	0000-0002-6556-4734	64335c4cf30113be20af59ded6ecc0e390d10b48	+2020-01-17T00:00:00Z
	Rhonda Y. Williams				
Q57030967	Jada Benn Torres	DBBC7493-8EA0-4AB1-828F-5AB2467BCF26	0000-0001-9678-4038	9eef05468c67858cd33736666d3978f1a204b6d9	+2020-01-19T00:00:00Z
Q40230904	Carwil Bjork-James		0000-0002-4977-0458		+2020-01-19T00:00:00Z
	Sophie Bjork-James				

spreadsheet (archived)

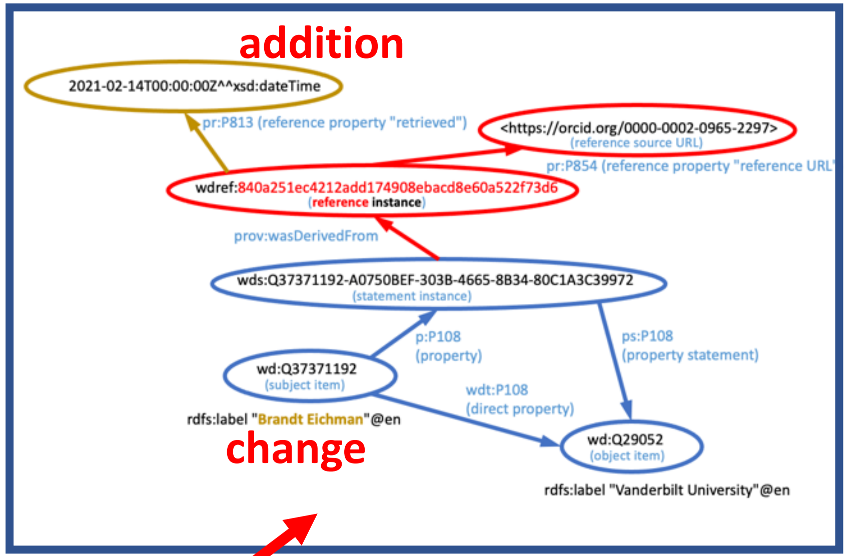

{} csv-metadata.js
Users > baskausj > uments > GitHub > linked-data > vanderbot > {} csv-metadata.json > ...

```
{
  "@type' TableGroup",
  "@cont  : "http://www.w3.org/ns/csvw",
  "table   [
    {
      "u     "medicine-employees-to-write.csv",
      "t    Schema": {
         umns": [

           'titles": "department",
           'name": "department",
           'datatype": "string",
           'suppressOutput": true


           "titles": "wikidataId",
           "name": "wikidataId",
           "datatype": "string",
           "suppressOutput": true
         },
         {
           "titles": "name",
           "name": "name",
           'datatype": "string",
           suppressOutput": true
         },
         {
           "titles": "labelEn",
           "name": "labelEn",
           "datatype": "string",
           "aboutUrl": "http://www.wikidata.org/entity/{wikidataId}",
           "propertyUrl": "rdfs:label",
           "lang": "en"
         },
         {
```

rdf-tabulator script


pr:P854 (reference property "reference URL")
<https://orcid.org/0000-0002-0965-2297>
(reference source URL)
wdref:840a251ec4212add174908ebacd8e60a522f73d6
(reference instance)
prov:wasDerivedFrom
wds:Q37371192-A0750BEF-303B-4665-8B34-80C1A3C39972
(statement instance)
p:P108
(property)
ps:P108
(property statement)
wd:Q37371192
(subject item)
wdt:P108
(direct property)
rdfs:label "Brandt F. Eichman"@en
wd:Q29052
(object item)
rdfs:label "Vanderbilt University"@en

local triplestore (e.g. Fuseki)


addition
2021-02-14T00:00:00Z^^xsd:dateTime
pr:P813 (reference property "retrieved")
<https://orcid.org/0000-0002-0965-2297>
(reference source URL)
pr:P854 (reference property "reference URL"
wdref:840a251ec4212add174908ebacd8e60a522f73d6
(reference instance)
prov:wasDerivedFrom
wds:Q37371192-A0750BEF-303B-4665-8B34-80C1A3C39972
(statement instance)
p:P108
(property)
ps:P108
(property statement)
wd:Q37371192
(subject item)
wdt:P108
(direct property)
rdfs:label "Brandt Eichman"@en
change
wd:Q29052
(object item)
rdfs:label "Vanderbilt University"@en

Wikidata some time later

compare by federated SPARQL query

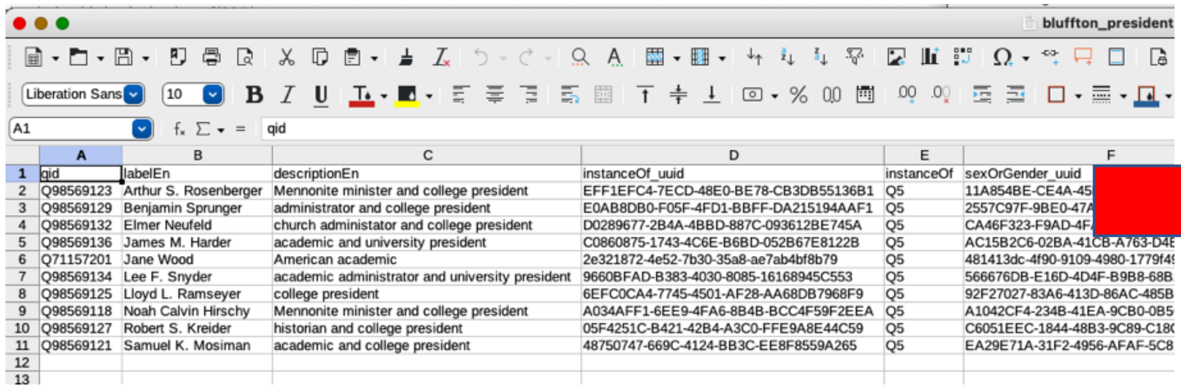
Simple example (manual RDF handling)

1. Generate RDF from CSV and schema using **`rdf-tabulator`** Ruby script (written by Greg Kellogg) and save as an RDF/Turtle file.
2. Load the file into a Fuseki triplestore.
3. Generate the entailed "shortcut path" triples using SPARQL **`INSERT`** (loaded directly into triplestore by script).
4. Compare the local graph in Fuseki with the graph in Wikidata using a federated query to the Wikidata Query Service.

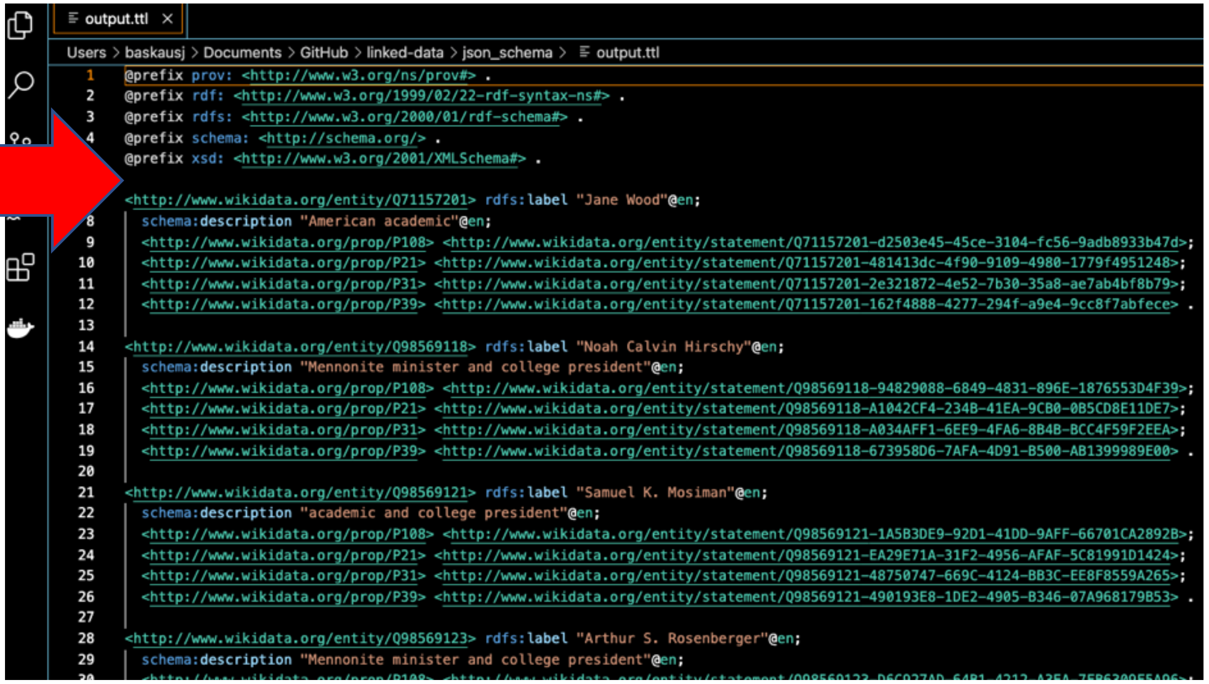
1. Generate RDF from CSV and schema

Command line to run Ruby script:

```
rdf serialize --input-format tabular --output-format ttl --metadata csv-metadata.json --minimal > output.ttl
```



	A	B	C	D	E	F
1	qid	labelEn	descriptionEn	instanceOf_uuid	instanceOf	sexOrGender_uuid
2	Q98569123	Arthur S. Rosenberger	Mennonite minister and college president	EFF1EFC4-7ECD-48E0-BE78-CB3DB55136B1	Q5	11A854BE-CE4A-45
3	Q98569129	Benjamin Sprunger	administrator and college president	E0AB8DB0-F05F-4FD1-BBFF-DA215194AAF1	Q5	2557C97F-9BE0-47A
4	Q98569132	Elmer Neufeld	church administator and college president	D0289677-2B4A-4BBD-887C-093612BE745A	Q5	CA46F323-F9AD-4F
5	Q98569136	James M. Harder	academic and university president	C0860875-1743-4C6E-B6BD-052B67E8122B	Q5	AC15B2C6-02BA-41CB-A763-D4E
6	Q71157201	Jane Wood	American academic	2e321872-4e52-7b30-35a8-ae7ab4bf8b79	Q5	481413dc-4f90-9109-4980-1779f4
7	Q98569134	Lee F. Snyder	academic administrator and university president	9660BFAD-B383-4030-8085-16168945C553	Q5	566676DB-E16D-4D4F-B9B8-68B
8	Q98569125	Lloyd L. Ramseyer	college president	6EFC0CA4-7745-4501-AF28-AA68DB7968F9	Q5	92F27027-83A6-413D-86AC-485B
9	Q98569118	Noah Calvin Hirschy	Mennonite minister and college president	A034AFF1-6EE9-4FA6-8B4B-BCC4F59F2EEA	Q5	A1042CF4-234B-41EA-9CB0-0B5
10	Q98569127	Robert S. Kreider	historian and college president	05F4251C-B421-42B4-A3C0-FFE9A8E44C59	Q5	C6051EEC-1844-48B3-9C89-C18
11	Q98569121	Samuel K. Mosiman	academic and college president	48750747-669C-4124-BB3C-EE8F8559A265	Q5	EA29E71A-31F2-4956-AFAF-5C8
12						
13						


output.ttl
Users > baskausj > Documents > GitHub > linked-data > json_schema > output.ttl

```
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.wikidata.org/entity/Q71157201> rdfs:label "Jane Wood"@en;
  schema:description "American academic"@en;
  <http://www.wikidata.org/prop/P108> <http://www.wikidata.org/entity/statement/Q71157201-d2503e45-45ce-3104-fc56-9adb8933b47d>;
  <http://www.wikidata.org/prop/P21> <http://www.wikidata.org/entity/statement/Q71157201-481413dc-4f90-9109-4980-1779f4951248>;
  <http://www.wikidata.org/prop/P31> <http://www.wikidata.org/entity/statement/Q71157201-2e321872-4e52-7b30-35a8-ae7ab4bf8b79>;
  <http://www.wikidata.org/prop/P39> <http://www.wikidata.org/entity/statement/Q71157201-162f4888-4277-294f-a9e4-9cc8f7abfece> .

<http://www.wikidata.org/entity/Q98569118> rdfs:label "Noah Calvin Hirschy"@en;
  schema:description "Mennonite minister and college president"@en;
  <http://www.wikidata.org/prop/P108> <http://www.wikidata.org/entity/statement/Q98569118-94829088-6849-4831-896E-1876553D4F39>;
  <http://www.wikidata.org/prop/P21> <http://www.wikidata.org/entity/statement/Q98569118-A1042CF4-234B-41EA-9CB0-0B5CD8E11DE7>;
  <http://www.wikidata.org/prop/P31> <http://www.wikidata.org/entity/statement/Q98569118-A034AFF1-6EE9-4FA6-8B4B-BCC4F59F2EEA>;
  <http://www.wikidata.org/prop/P39> <http://www.wikidata.org/entity/statement/Q98569118-673958D6-7AFA-4D91-B500-AB1399989E00> .

<http://www.wikidata.org/entity/Q98569121> rdfs:label "Samuel K. Mosiman"@en;
  schema:description "academic and college president"@en;
  <http://www.wikidata.org/prop/P108> <http://www.wikidata.org/entity/statement/Q98569121-1A5B3DE9-92D1-41DD-9AFF-66701CA2892B>;
  <http://www.wikidata.org/prop/P21> <http://www.wikidata.org/entity/statement/Q98569121-EA29E71A-31F2-4956-AFAF-5C81991D1424>;
  <http://www.wikidata.org/prop/P31> <http://www.wikidata.org/entity/statement/Q98569121-48750747-669C-4124-BB3C-EE8F8559A265>;
  <http://www.wikidata.org/prop/P39> <http://www.wikidata.org/entity/statement/Q98569121-490193E8-1DE2-4905-B346-07A968179B53> .

<http://www.wikidata.org/entity/Q98569123> rdfs:label "Arthur S. Rosenberger"@en;
  schema:description "Mennonite minister and college president"@en;
```

Turtle RDF serialization using Wikibase model

1. Generate RDF from CSV and schema

Conversion with **`rdf-tabulator`** is the rate-limiting step in process

dataset	items	columns	CSV file size in kB (uncompressed/zip compressed)	conversion time (s)	triples	Turtle file size in kB (uncompressed/zip compressed)
Bluffton presidents	10	32	9/3	1	193	24/4
academic journals	431	71	160/58	27	13 106	1500/320
Vanderbilt researchers	5247	30	3900/960	70 (1 min)	62 634	7900/1300
gallery objects	4085	158	9300/1700	377 (6 min)	209 597	28000/4400

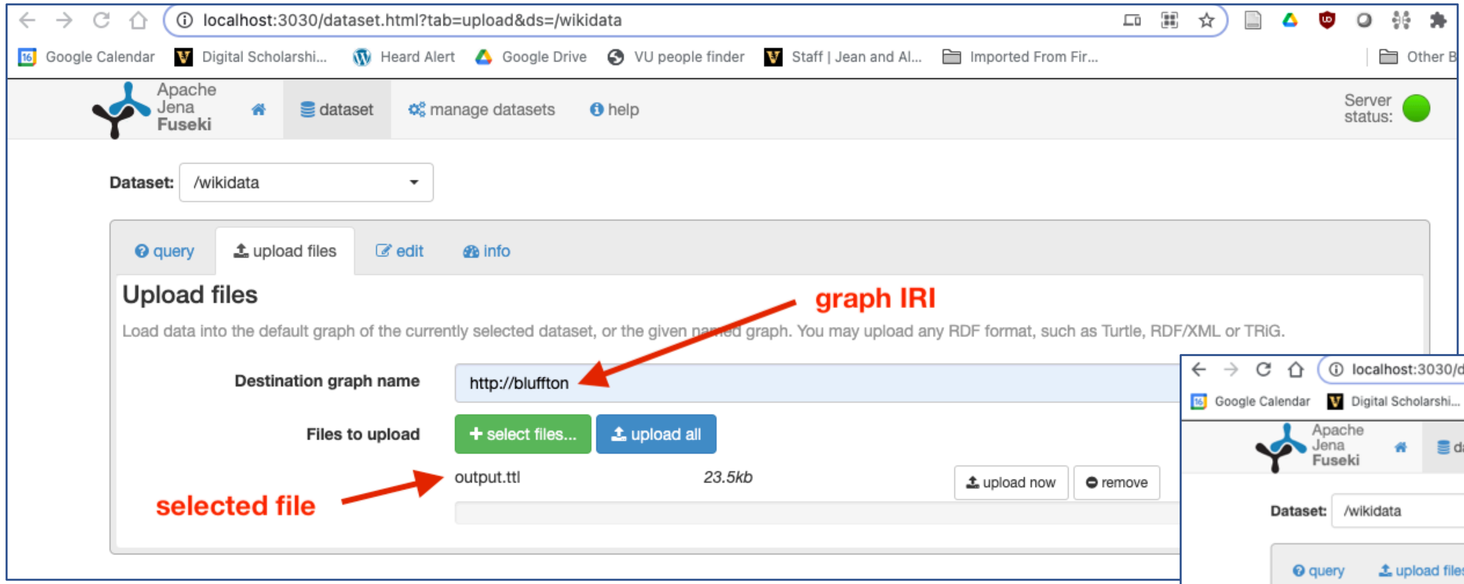
Baskauf and Baskauf (in review) Table 2

Uninvestigated alternatives:

https://github.com/AtomGraph/CSV2RDF (151 348 939 triples in under 27 minutes)

https://github.com/Swirrl/csv2rdf (have not checked speed)

2. Load the file into a Fuseki triplestore



localhost:3030/dataset.html?tab=upload&ds=/wikidata

Dataset: /wikidata

Upload files

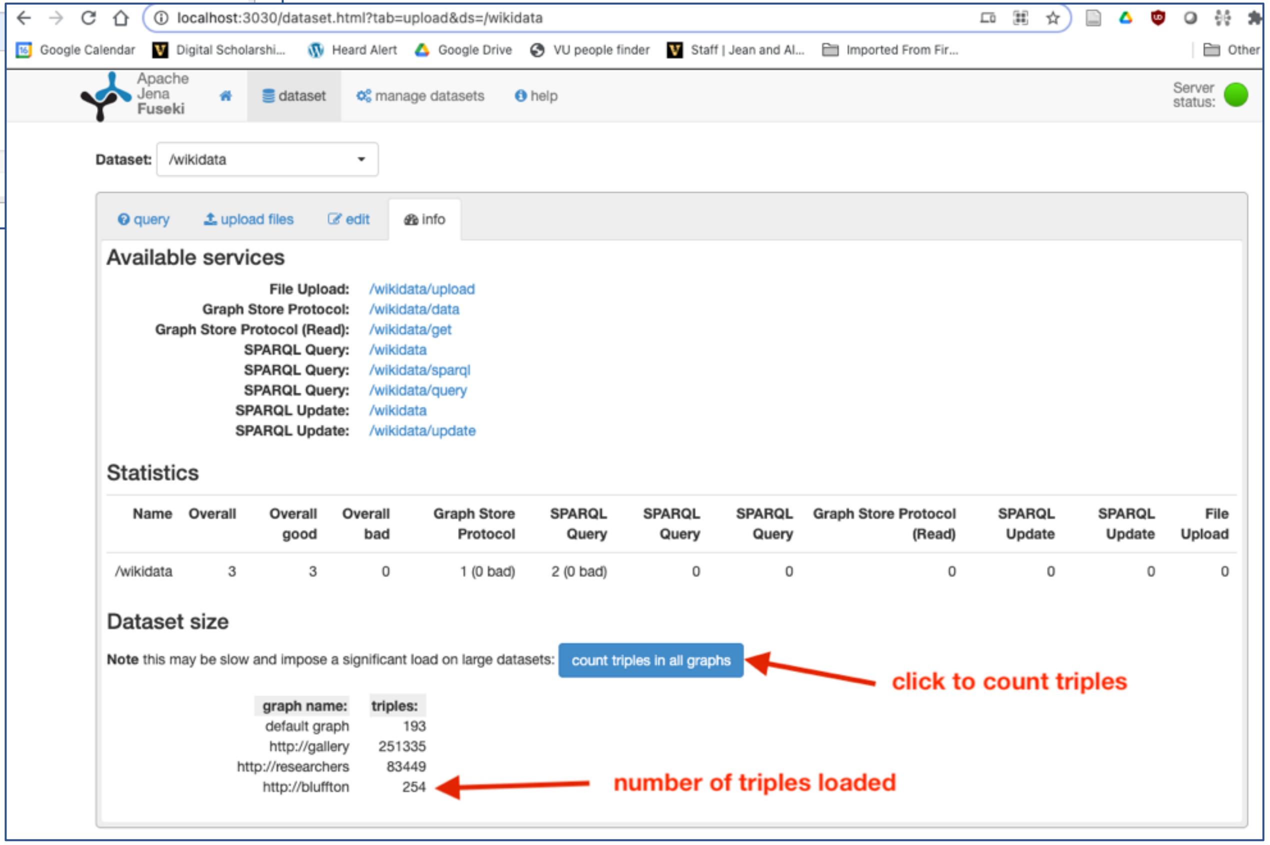
Load data into the default graph of the currently selected dataset, or the given named graph. You may upload any RDF format, such as Turtle, RDF/XML or TRiG.

graph IRI

Destination graph name	http://bluffton
Files to upload	+ select files... / upload all

selected file

output.ttl 23.5kb upload now remove



localhost:3030/dataset.html?tab=upload&ds=/wikidata

Dataset: /wikidata

Available services

File Upload:	/wikidata/upload
Graph Store Protocol:	/wikidata/data
Graph Store Protocol (Read):	/wikidata/get
SPARQL Query:	/wikidata
SPARQL Query:	/wikidata/sparql
SPARQL Query:	/wikidata/query
SPARQL Update:	/wikidata
SPARQL Update:	/wikidata/update

Statistics

Name	Overall	Overall good	Overall bad	Graph Store Protocol	SPARQL Query	SPARQL Query	SPARQL Query	Graph Store Protocol (Read)	SPARQL Update	SPARQL Update	File Upload
/wikidata	3	3	0	1 (0 bad)	2 (0 bad)	0	0	0	0	0	0

Dataset size

Note this may be slow and impose a significant load on large datasets: count triples in all graphs

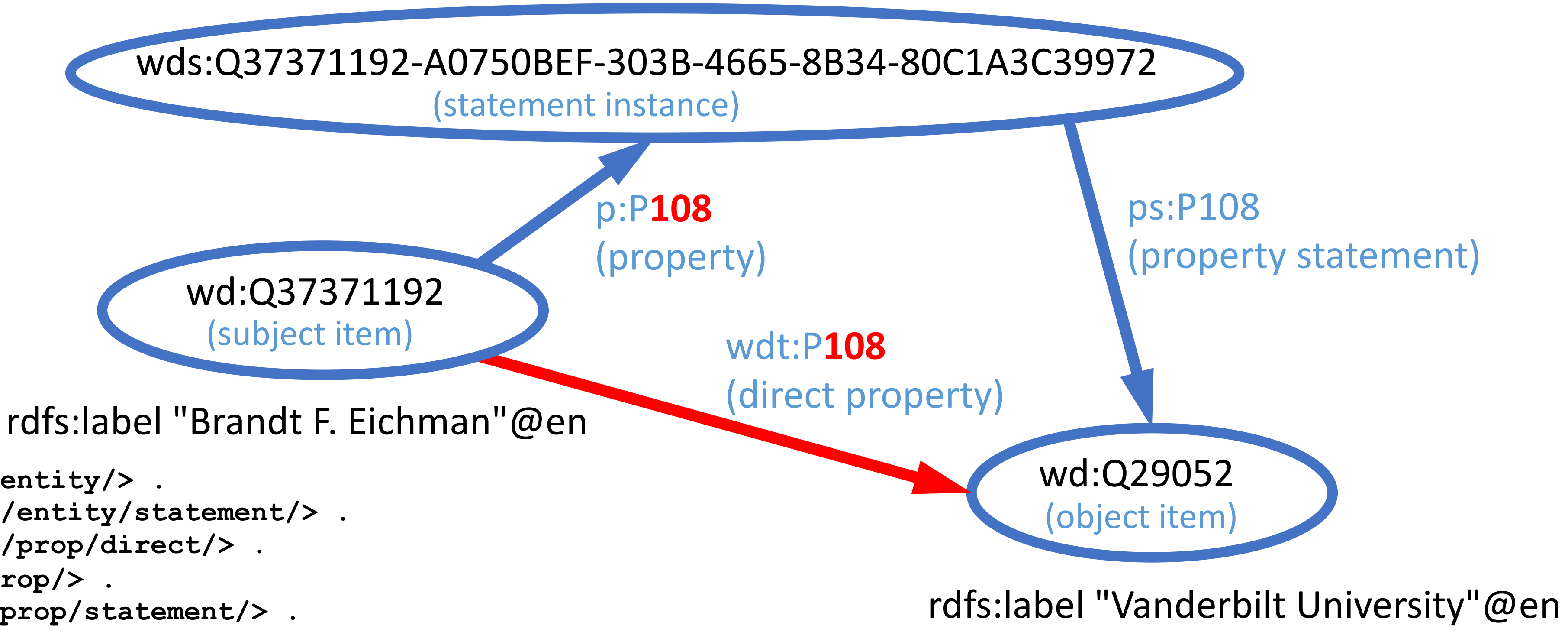
click to count triples

graph name:	triples:
default graph	193
http://gallery	251335
http://researchers	83449
http://bluffton	254

number of triples loaded

3. Generate the entailed "shortcut path" triples

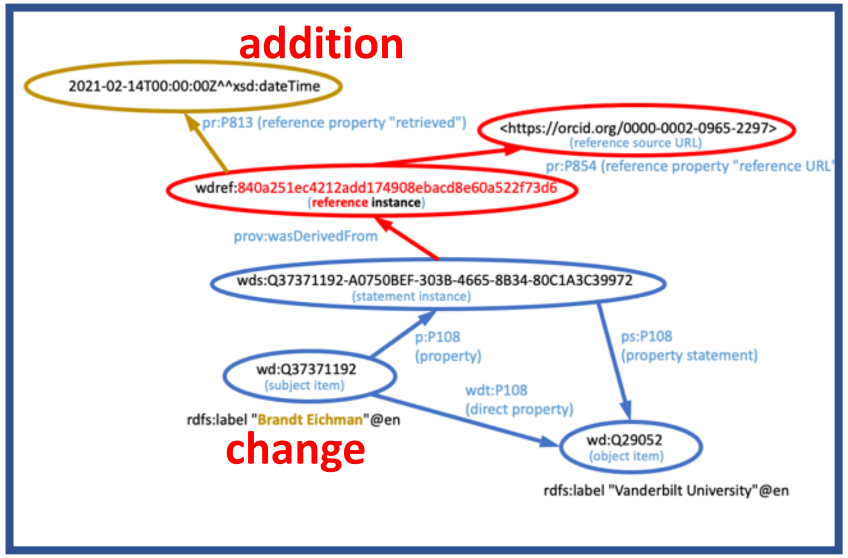
```
with <http://researchers>
insert {?item ?truthyProp ?value.}
where {
  ?item ?p ?statement.
  ?statement ?ps ?value.
  filter(substr(str(?ps),1,40)="http://www.wikidata.org/prop/statement/P")
  bind(substr(str(?ps),40) as ?id)
  bind(iri(concat("http://www.wikidata.org/prop/direct/", ?id)) as ?truthyProp)
  }
```



```
@prefix wd: <http://www.wikidata.org/entity/> .
@prefix wds: <http://www.wikidata.org/entity/statement/> .
@prefix wdt: <http://www.wikidata.org/prop/direct/> .
@prefix p: <http://www.wikidata.org/prop/> .
@prefix ps: <http://www.wikidata.org/prop/statement/> .
```

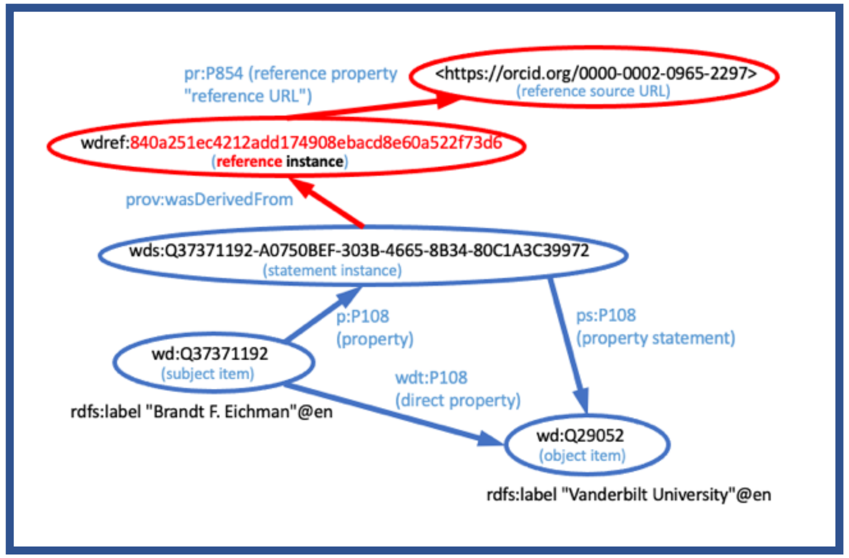
4. Compare local graph with Wikidata graph

```
SELECT DISTINCT *
   WHERE {
   SERVICE <https://query.wikidata.org/sparql> {?subject ?predicate ?object.}
     MINUS
     {
     GRAPH <http://researchers> {?subject ?predicate ?object.}
     }
   }
```


addition
2021-02-14T00:00:00Z^^xsd:dateTime
pr:P813 (reference property "retrieved")
<https://orcid.org/0000-0002-0965-2297>
(reference source URL)
pr:P854 (reference property "reference URL")
wdref:840a251ec4212add174908ebacd8e60a522f73d6
(reference instance)
prov:wasDerivedFrom
wds:Q37371192-A0750BEF-303B-4665-8B34-80C1A3C39972
(statement instance)
p:P108
(property)
ps:P108
(property statement)
wd:Q37371192
(subject item)
wdt:P108
(direct property)
rdfs:label "Brandt Eichman"@en
change
wd:Q29052
(object item)
rdfs:label "Vanderbilt University"@en

Wikidata

minus


pr:P854 (reference property "reference URL")
<https://orcid.org/0000-0002-0965-2297>
(reference source URL)
wdref:840a251ec4212add174908ebacd8e60a522f73d6
(reference instance)
prov:wasDerivedFrom
wds:Q37371192-A0750BEF-303B-4665-8B34-80C1A3C39972
(statement instance)
p:P108
(property)
ps:P108
(property statement)
wd:Q37371192
(subject item)
wdt:P108
(direct property)
rdfs:label "Brandt F. Eichman"@en
wd:Q29052
(object item)
rdfs:label "Vanderbilt University"@en

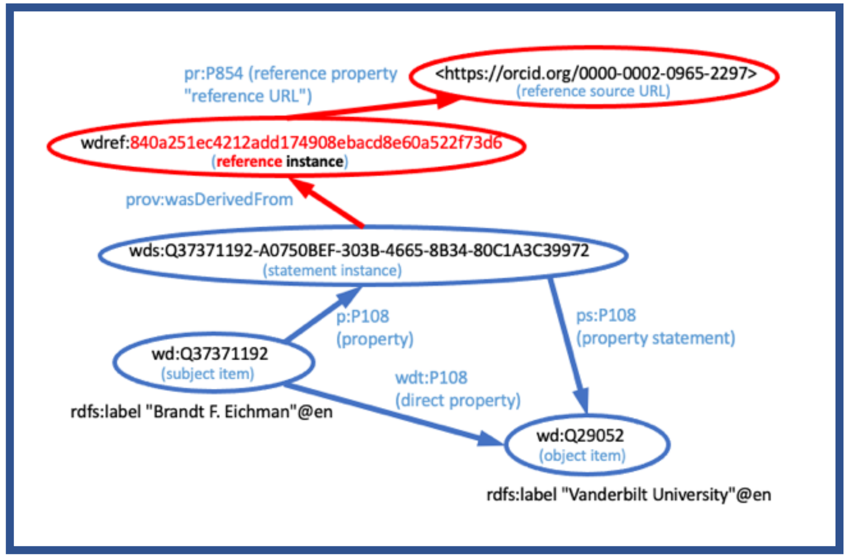
graph generated from CSV

Triples in Wikidata not in local copy (additions or changes)

wdref:840a251ec4212add174908ebacd8e60a522f73d6 pr:P813 2021-02-14T00:00:00Z^^xsd:dateTime.
wd:Q37371192 rdfs:label "Brandt Eichman"@en.

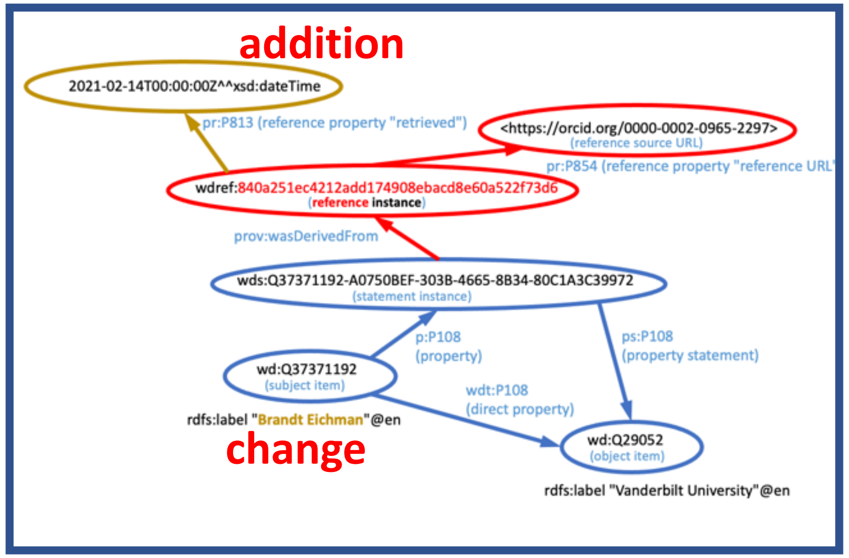
4. Compare local graph with Wikidata graph

```
SELECT DISTINCT *
   WHERE {
   GRAPH <http://researchers> {?subject ?predicate ?object.}
     MINUS
     {
     SERVICE <https://query.wikidata.org/sparql> {?subject ?predicate ?object.}
     }
   }
```


pr:P854 (reference property "reference URL")
<https://orcid.org/0000-0002-0965-2297>
(reference source URL)
wdref:840a251ec4212add174908ebacd8e60a522f73d6
(reference instance)
prov:wasDerivedFrom
wds:Q37371192-A0750BEF-303B-4665-8B34-80C1A3C39972
(statement instance)
p:P108
(property)
ps:P108
(property statement)
wd:Q37371192
(subject item)
wdt:P108
(direct property)
rdfs:label "Brandt F. Eichman"@en
wd:Q29052
(object item)
rdfs:label "Vanderbilt University"@en

graph generated from CSV

minus


addition
2021-02-14T00:00:00Z^^xsd:dateTime
pr:P813 (reference property "retrieved")
<https://orcid.org/0000-0002-0965-2297>
(reference source URL)
pr:P854 (reference property "reference URL"
wdref:840a251ec4212add174908ebacd8e60a522f73d6
(reference instance)
prov:wasDerivedFrom
wds:Q37371192-A0750BEF-303B-4665-8B34-80C1A3C39972
(statement instance)
p:P108
(property)
ps:P108
(property statement)
wd:Q37371192
(subject item)
wdt:P108
(direct property)
rdfs:label "Brandt Eichman"@en
change
wd:Q29052
(object item)
rdfs:label "Vanderbilt University"@en

Wikidata

Triples in local copy not in Wikidata (deletions or changes)

wd:Q37371192 rdfs:label "Brandt F. Eichman"@en.