

# Hybrid linked data approaches in traditional discovery environments using Share-VDE linked data

Jim Hahn, University of Pennsylvania Libraries

[jimhahn@upenn.edu](mailto:jimhahn@upenn.edu)



# Hybrid Linked Data

<https://guides.library.upenn.edu/linked-data>

# Linked data and more familiar formats

It is often possible to include linked data in more traditional representations, or to make connections between linked data and more familiar formats.

We expect these mixed-format or “hybrid” linked data environments to be the most common way in which linked data is used in production in the next few years

# Traditional Discovery

The screenshot displays the Blacklight search interface. At the top, the 'blacklight' logo is on the left, and 'Bookmarks 0', 'History', and 'Login' links are on the right. Below the header is a search bar with a dropdown menu set to 'All Fields' and a search button. Under the search bar, there are buttons for 'Start Over', 'Format > Book' (with a close icon), and 'Pivot Field > Book' (with a close icon). To the left of the results is a sidebar titled 'Limit your search' with filters for 'Format' (set to 'Book' with 30 results), 'Publication Year', 'Topic', and 'Language'. The main results area shows '1. "Strong Medicine speaks"' with a bookmark icon. Below the title, the following details are listed: Title: "Strong Medicine speaks", Author: Hearth, Amy Hill, 1958-, Format: Book, Language: English, Published: New York, and Call number: E99.D2 H437 2008. Navigation links include « Previous | 1 - 10 of 30 | Next » and buttons for 'Sort by relevance' and '10 per page'.

blacklight Bookmarks 0 History Login

All Fields Search...

Start Over Format > Book × Pivot Field > Book ×

Limit your search

Format

Book × 30

Publication Year >

Topic >

Language >

« Previous | 1 - 10 of 30 | Next »

Sort by relevance ▾ 10 per page ▾

1. ["Strong Medicine speaks"](#) ☐ Bookmark

Title: "Strong Medicine speaks"

Author: Hearth, Amy Hill, 1958-

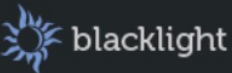
Format: Book

Language: English

Published: New York

Call number: E99.D2 H437 2008

# Hybrid Linked Data Discovery

blacklight

Bookmarks 0 History Login

All Fields Search...

Start Over Back to Search

« Previous | 1 of 30 | [Next](#) »

## "Strong Medicine speaks"

Title: "Strong Medicine speaks"

Subtitle: a Native American elder has her say : an oral history /

Author: Hearth, Amy Hill, 1958-

Format: Book

More Information: <http://www.loc.gov/catdir/toc/ecip0719/2007020969.html>, <http://www.loc.gov/catdir/enhancements/fy0808/2007020969-d.html>, and <http://www.loc.gov/catdir/enhancements/fy0808/2007020969-s.html>

Language: English

### Knowledge Panel

Original works by Publications by

Filter original works... Creator Genre

12 results

Sort by (A - Z)

- Having our say (Biography)**

Creators:

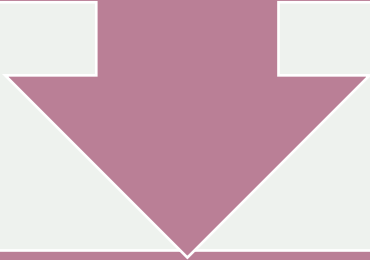
  - Sarah Louise Delany (author)
  - Annie Elizabeth Delany (other)
  - Amy Hill Hearth (other)
  - Joanna Banks Collection of African American Books (University of Pennsylvania) (other)
  - Gotham Book Mart Collection (University of Pennsylvania)



# SVDE Linked Data



The Share-VDE project (<https://svde.org>) is a collaborative discovery environment based on linked data. Explored in this talk are several lesser known and non-intuitive uses of Share-VDE linked data.



### Deliverables from Share-VDE

Enriched  
MARC

RDF  
triples

[SVDE.org](https://svde.org)

## Share-VDE Data

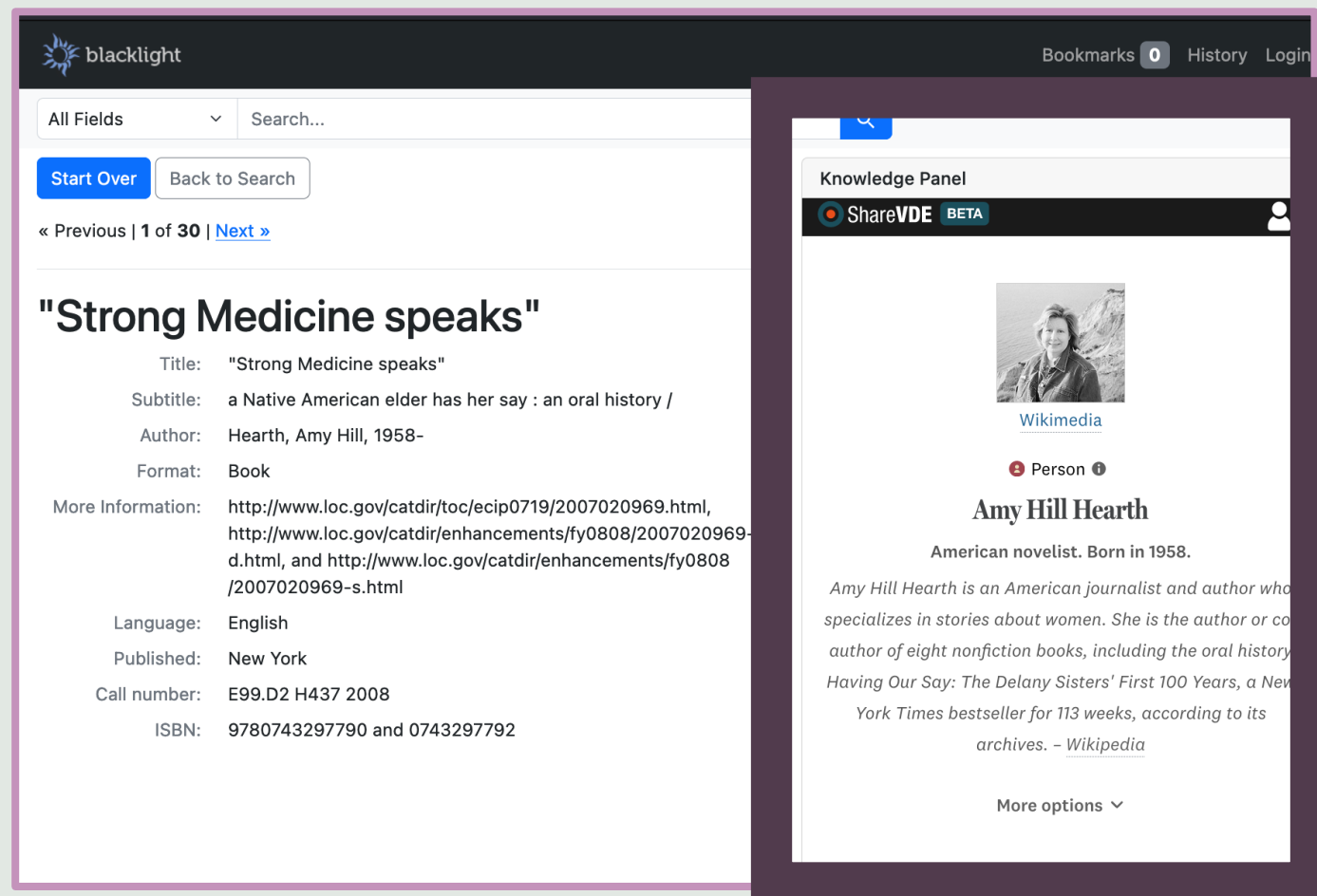


# Blacklight 8

Knowledge Panel



# SVDE Person page as Knowledge panel



The image shows a Blacklight search results page. The main entry is for the book "Strong Medicine speaks" by Amy Hill Hearth. To the right, a Knowledge Panel is displayed, featuring a photo of the author, a link to her Wikipedia page, and a brief biography.

**blacklight** Bookmarks 0 History Login

All Fields Search...

Start Over Back to Search

« Previous | 1 of 30 | Next »

**"Strong Medicine speaks"**

Title: "Strong Medicine speaks"

Subtitle: a Native American elder has her say : an oral history /

Author: Hearth, Amy Hill, 1958-

Format: Book

More Information: <http://www.loc.gov/catdir/toc/ecip0719/2007020969.html>,  
<http://www.loc.gov/catdir/enhancements/fy0808/2007020969-d.html>, and <http://www.loc.gov/catdir/enhancements/fy0808/2007020969-s.html>

Language: English


Published: New York

Call number: E99.D2 H437 2008

ISBN: 9780743297790 and 0743297792

**Knowledge Panel**

ShareVDE BETA

  
[Wikimedia](#)

Person ⓘ

**Amy Hill Hearth**

American novelist. Born in 1958.

*Amy Hill Hearth is an American journalist and author who specializes in stories about women. She is the author or co-author of eight nonfiction books, including the oral history *Having Our Say: The Delany Sisters' First 100 Years*, a New York Times bestseller for 113 weeks, according to its archives. – Wikipedia*

More options ▾

# SVDE Linked Data for Alma Automated Subject Assignment



# Alma Sandbox Experiment

## Alma/Annif Nightshift

jimhahn@upenn.edu

The project was inspired by the BookOps Nightshift project (<https://github.com/BookOps-CAT/NightShift>), a copy cataloging bot that used OCLC numbers to match brief records.

This variation may be useful to enhance brief records when OCLC matching isn't possible.

The data flow begins from Alma Brief record reports. The titles (and/or authors) are sent to a pre-packaged machine learning service for FAST subject assignment using the Annif codebase (<https://github.com/NatLibFi/Annif>).

The machine learning model used for this project is a NN-ensemble ([https://github.com/NatLibFi/Annif/wiki/Backend%3A-nn\\_ensemble](https://github.com/NatLibFi/Annif/wiki/Backend%3A-nn_ensemble)) of Omikuji (<https://github.com/NatLibFi/Annif/wiki/Backend%3A-Omikuji>) and TF-IDF (<https://github.com/NatLibFi/Annif/wiki/Backend%3A-TF-IDF>) models.

The data in the customized Fast Annif API were collected from Penn Libraries and IvyPlus POD data (<https://pod.stanford.edu/>).

## Get a row of the Alma report of titles to be processed

```
In [...]: #import the data as string data types
nosubjectsdf = pd.read_csv('/Users/jimhahn/Documents/GitHub/alma-nightshift/emptyk')
```

```
In [...]: #what does the data look like?
nosubjectsdf.head()
```

```
Out [...]:
```

	Type / Creator / Imprint	Title	Barcode	Inventory Number	Receiving Number	Library	Library Unit	Temporary Library	Creation Date
0	Book By Cartarescu, Mircea (2022)	Premio FIL de Literatura en Lenguas Romances /	NaN	NaN	NaN	LIBRA	NaN	NaN	03/30/2023 16:24:59

1 rows x 44 columns

Annif-client

Public

Watch 5

Fork 2

Star 2

master

3 branches

2 tags

Go to file

Add file

<> Code

juhoinkinen

Bump version: 0.3.0 → 0.3.1

16fd749 on Mar 11, 2021

42 commits

.github/workflows	Create python-package.yml	2 years ago
tests	Add minimal test suite (only tests projects property)	4 years ago
.gitignore	Initial commit	5 years ago
LICENSE.txt	switch license to Apache 2.0 (same as used by Annif)	5 years ago
README.md	update README to mention PyPI install	5 years ago
annif_client.py	Merge branch 'master' into issue1-support-for-learn...	4 years ago
setup.cfg	Bump version: 0.3.0 → 0.3.1	2 years ago
setup.py	Bump version: 0.3.0 → 0.3.1	2 years ago

README.md

# Annif-client

About

Python client library for accessing Annif REST API

annif

Readme

View license

Activity

2 stars

5 watching

2 forks

Report repository

Releases

2 tags

Contributors 2

<https://github.com/NatLibFi/Annif-client>



```
In [... if __name__ == '__main__':
    annif = AnnifClient()

    #select the title from the nosubjectsdf,
    title = nosubjectsdf['Title'].iloc[0]

    # send it to Annif API
    url = 'http://jimhahn-dev.library.upenn.int:5000/v1/projects/nn-ensemble-Fast,
    payload = {'text': title}
    req = requests.post(url, data=payload)
    req.raise_for_status()
    print(req.json()['results'])
    #print(req.json()['results'][0]['uri'])
    #save the response as a key value pair in pandas dataframe called subjectsdf
    subjectsdf = pd.DataFrame(req.json()['results'])
    print(subjectsdf)
```

```
{'label': 'Literary prizes', 'notation': None, 'score': 0.553928792476654, 'uri': 'http://id.worldcat.org/fast/999945'}, {'label': 'Premio FIL de Literatura en
Lenguas Romances', 'notation': None, 'score': 0.2416485697031021, 'uri': 'http://
id.worldcat.org/fast/1895046'}, {'label': 'Premio Nacional de Literatura', 'notat
ion': None, 'score': 0.17720665037631989, 'uri': 'http://id.worldcat.org/fast/107
5380'}, {'label': 'Contests in literature', 'notation': None, 'score': 0.17324309
051036835, 'uri': 'http://id.worldcat.org/fast/876660'}, {'label': 'Spanish liter
ature--Study and teaching (Secondary)', 'notation': None, 'score': 0.172929123044
01398, 'uri': 'http://id.worldcat.org/fast/1128613'}, {'label': 'Premio Cremona',
'notation': None, 'score': 0.17263424396514893, 'uri': 'http://id.worldcat.org/fa
st/1075351'}, {'label': 'Building, Brick--Awards', 'notation': None, 'score': 0.1
6265563666820526, 'uri': 'http://id.worldcat.org/fast/840900'}, {'label': 'Drawin
g--Competitions', 'notation': None, 'score': 0.160080701127472, 'uri': 'http://i
d.worldcat.org/fast/897729'}, {'label': 'Revolutionary literature, Spanish Americ
an', 'notation': None, 'score': 0.15990696847438812, 'uri': 'http://id.worldcat.o
rg/fast/1096640'}, {'label': 'Press releases', 'notation': None, 'score': 0.15116
143226623535, 'uri': 'http://id.worldcat.org/fast/1075892'}]
```

	label	notation	score \
0	Literary prizes	None	0.553929
1	Premio FIL de Literatura en Lenguas Romances	None	0.241649
2	Premio Nacional de Literatura	None	0.177207
3	Contests in literature	None	0.173243
4	Spanish literature--Study and teaching (Second...	None	0.172929
5	Premio Cremona	None	0.172634
6	Building, Brick--Awards	None	0.162656
7	Drawing--Competitions	None	0.160081
8	Revolutionary literature, Spanish American	None	0.159907
9	Press releases	None	0.151161

<http://annif.info/>

Evaluate the subject recommendations looking for any have a confidence score above 0.5

```
# check if the subjects dataframe has any values above 0.5
# and if so, add the subject to a new column in the nosubjectsdf and the subject
if subjectsdf['score'].iloc[0] > 0.5:
    nosubjectsdf['subject'] = subjectsdf['uri'].iloc[0]
    nosubjectsdf['label'] = subjectsdf['label'].iloc[0]
else:
    print("No subject found")
```

Get the brief record from Alma for the title so that we can add the subject to the MARC record

```
# use for statement in https://api-na.hosted.exlibrisgroup.com/almaws/v1/bibs/:mm:
# to download the marc records
# create a for loop to iterate through the mmsid list
# and download the marc record for each mmsid
# and save it to a folder on the desktop

# create a list of mmsids from the nosubjectsdf
mmsids = nosubjectsdf['MMS_ID'].tolist()
print(mmsids)

# create a for loop to iterate through the mmsid list
# and download the marc record for each mmsid
# and save it to a folder on the desktop

for mmsid in mmsids:
    url = 'https://api-na.hosted.exlibrisgroup.com/almaws/v1/bibs?mms_id='+ mmsid
    print(url)
    r = requests.get(url)
    # parse the response as xml
    root = ET.fromstring(r.content)
    #print(root)
    #print(ET.tostring(root, pretty_print=True))
    #we only want the record
    record = root.find('.//record')
    #print(record)
    #print(ET.tostring(record, pretty_print=True))

    #save the record as a file
    filename = mmsid + '.xml'
    print(filename)
    with open(filename, 'wb') as f:
        f.write(ET.tostring(record, pretty_print=True))
        f.close()
```



Before we can add the subject to the MARC record we need to know the type of subject we are working with here

In [...]

```
# we need to check the type of subject
# we can lookup FASTAll/lookup/nt to see the type of subject
# If the subject is found in FASTChronological.nt it is a Chronological subject
# If the subject is found in FASTCorporate.nt it is a Corporate subject
# If the subject is found in FASTEvent.nt it is an Event subject
# If the subject is found in FASTForm.nt it is a Form subject
# If the subject is found in FASTGeographic.nt it is a Geographic subject
# If the subject is found in FASTNamedEvent.nt it is a Named Event subject
# If the subject is found in FASTPersonal.nt it is a Personal subject
# If the subject is found in FASTTopic.nt it is a Topic subject
# If the subject is found in FASTUniformTitle.nt it is a Uniform Title subject

#query the FASTAll/lookup/nt/ folder to see if the subject is there
# if it is, then add the subject type to the nosubjectsdf
# open the nt and read it
# if the subject is found in the nt, then add the subject type to the nosubjectsdf

# create a list of subjects from the nosubjectsdf

#make the FASTAll/lookup/FASTTopical-nt.csv into a dataframe
fasttopicaldf = pd.read_csv('./FASTAll/lookup/FASTTopical-nt.csv', dtype=str)

#make the FASTAll/lookup/FASTChronological-nt.csv into a dataframe
fastchronologicaldf = pd.read_csv('./FASTAll/lookup/FASTChronological-nt.csv', dtype=str)

#make the FASTAll/lookup/FASTCorporate-nt.csv into a dataframe
fastcorporatedf = pd.read_csv('./FASTAll/lookup/FASTCorporate-nt.csv', dtype=str)

#make the FASTAll/lookup/FASTEvent-nt.csv into a dataframe
fasteventdf = pd.read_csv('./FASTAll/lookup/FASTEvent-nt.csv', dtype=str)

#make the FASTAll/lookup/FASTFormGenre-nt.csv into a dataframe
fastformdf = pd.read_csv('./FASTAll/lookup/FASTFormGenre-nt.csv', dtype=str)

#make the FASTAll/lookup/FASTGeographic-nt.csv into a dataframe
fastgeographicdf = pd.read_csv('./FASTAll/lookup/FASTGeographic-nt.csv', dtype=str)

#make the FASTAll/lookup/FASTEvent-nt.csv into a dataframe
fastnamedeventdf = pd.read_csv('./FASTAll/lookup/FASTEvent-nt.csv', dtype=str)

#make the FASTAll/lookup/FASTPersonal-nt.csv into a dataframe
fastpersonaldf = pd.read_csv('./FASTAll/lookup/FASTPersonal-nt.csv', dtype=str)

#make the FASTAll/lookup/FASTTitle-nt.csv into a dataframe
fastuniformtitledf = pd.read_csv('./FASTAll/lookup/FASTTitle-nt.csv', dtype=str)
```

```

#we're going to be using pymarc to read the marc records and add a new subject fr
#convert the MARCXML at ./alma.marc/ to MARC using pymarc
mmsids = nosubjectsdf['MMS_ID'].tolist()
print(mmsids)

# create a for loop to iterate through the mmsid list
# and download the marc record for each mmsid
# and save it to a folder on the desktop

for mmsid in mmsids:
    with open(mmsid + '.marc', 'rb') as fh:
        reader = MARCReader(fh)
        for record in reader:
            record.add_field(
                #evaluate the type of topic from the nosubjects, if topical use 650:
                if nosubjectsdf['subject_type'].iloc[0] == 'Topical':
                    record.add_field(
                        Field(
                            tag = '650',
                            indicators = [' ', '7'],
                            subfields = [
                                'a', nosubjectsdf['label'].iloc[0],
                                '2', 'fast',
                                '0', nosubjectsdf['subject'].iloc[0]
                            ]
                        )
                    )
                #evaluate the type of topic from the nosubjects, if chronological use
                elif nosubjectsdf['subject_type'].iloc[0] == 'Chronological':
                    record.add_field(
                        Field(
                            tag = '648',
                            indicators = [' ', '7'],
                            subfields = [
                                'a', nosubjectsdf['label'].iloc[0],
                                '2', 'fast',
                                '0', nosubjectsdf['subject'].iloc[0]
                            ]
                        )
                    )
                #evaluate the type of topic from the nosubjects, if corporate use 610
                elif nosubjectsdf['subject_type'].iloc[0] == 'Corporate':
                    record.add_field(
                        Field(
                            tag = '610',
                            indicators = [' ', '7'],
                            subfields = [
                                'a', nosubjectsdf['label'].iloc[0],
                                '2', 'fast',
                                '0', nosubjectsdf['subject'].iloc[0]
                            ]
                        )
                    )
                #evaluate the type of topic from the nosubjects, if event use 611:
                elif nosubjectsdf['subject_type'].iloc[0] == 'Event':
                    record.add_field(
                        Field(
                            tag = '611',
                            indicators = [' ', '7'],
                            subfields = [
                                'a', nosubjectsdf['label'].iloc[0],
                                '2', 'fast',
                                '0', nosubjectsdf['subject'].iloc[0]
                            ]
                        )
                    )
                #evaluate the type of topic from the nosubjects, if form/genre use 651:
                elif nosubjectsdf['subject_type'].iloc[0] == 'Form/Genre':
                    record.add_field(
                        Field(
                            tag = '655',
                            indicators = [' ', '7'],
                            subfields = [
                                'a', nosubjectsdf['label'].iloc[0],
                                '2', 'fast',
                                '0', nosubjectsdf['subject'].iloc[0]
                            ]
                        )
                    )
            )

```

- Adding the 650 with the FAST heading from Annif.

```

        )
    )
    print(record)
    writer = XMLWriter(open('./alma_marc_output/' +
    writer.write(record)
    writer.close()
    fh.close()

['9979063291703681']
=LDR 00374nam#a2200121#u#4500
=001 9979063291703681
=008 230330s2022###xx#####r####000#0#eng#d
=005 20230330162416.0
=100 1\\$aCartarescu, Mircea
=245 10$aPremio FIL de Literatura en Lenguas Romances /
=983 \\$a87508-14$g1$hlatt-app
=984 \\$a14.9000000000000003552713678800500929355621337890625$d87508
=985 \\$astor$dLAP
=650 \\$aLiterary prizes$2fast$0http://id.worldcat.org/fast/999945

```

### Prepare the enriched MARC record for export to Alma

```

In [...] # modify the xml for Alma upload needs <bib> root element
# add the <bib> root element to the xml
marc_file = glob.glob("./alma_marc_output/*.xml")
print(marc_file)

for file in marc_file:
    with open(file, 'r') as f:
        xml = f.read()
        #strip out the xml declaration
        xml = re.sub('<\\?xml.*?\\?>', '', xml)

        xml = '<bib>' + xml + '</bib>'
        f.close()
    with open(file, 'w') as f:
        f.write(xml)
        f.close()
    print('done')

alma_marc_output/9979063291703681.xml']

```



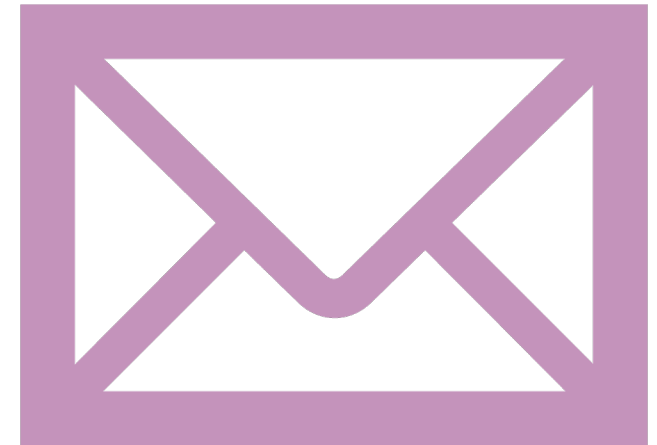
# Concluding thoughts

- SVDE data has been useful for several non-traditional uses for Linked Data experimentation and for hybrid linked data discovery.
  - **Subject Indexing**
  - **Knowledge Panels** for author agents – bringing in related works and biographical assertions



# Thank you!

- [jimhahn@upenn.edu](mailto:jimhahn@upenn.edu)



# Acknowledgement

This presentation contains information from [FAST \(Faceted Application of Subject Terminology\) Data](#) which is made available by OCLC Online Computer Library Center, Inc. under the [ODC Attribution License](#).

