

デジタルアーカイブと その社会的活用

第10回

立命館大学 映像学部講義
福田一史

<https://scrapbox.io/fukudakz/21デジタルアーカイブとその社会的活用>



manabaRにもリンクがあります

次回講義は休講です

- 2021-06-21は講師の都合（本務校都合）により休講になります。
- 補講は調整中ですが、「2021-07-11」に行う予定です。
 - レポート作成のための相談会のような形式での開催を想定しています。

コンテンツ

1. メタデータの検索と利用：演習

メタデータの検索と活用

SPARQLを用いたRDFデータの活用

SPARQL

- RDFで記述されたメタデータは、その問い合わせ言語である「**SPARQL** (SPARQL Protocol and RDF Query Language)」を用いることで、リッチな検索や識別やデータ分析を可能となる。
 - ブラウザでアクセスする**ウェブのGUIでは達成できない**、もしくは**非常にコストがかかる機能**を、数多くかつ容易に**達成**できる。
- 2008年よりVer. 1.0が、2013年にVer. 1.1がW3C勧告
 - [SPARQL 1.1 Query Language \(W3C\)](#), [日本語版](#)
- SPARQLはRDFに出現する**パターン**の組み合わせや、フィルタリング、文字列指定などで必要とするデータの指定が可能。
- PHP, JavaScript, Perl, Python, Ruby, Rなど**複数のプログラム言語**でSPARQLを実装するための**ライブラリが公開**されている。

RDFストア

- RDFストアとは、**膨大なRDFデータ**（数十万～数億トリプル）を**登録し検索する**ためのデータベース。**トリプルストア**とも呼ばれる。
- 複数のRDFストアが公開されている
 - e.g. [Apache Jena](#), [Virtuoso](#)
 - [Comparison of triplestores – Wikipedia](#)

SPARQLエンドポイント

- SPARQLによるRDFデータの検索や分析の機能を提供するインターフェイス。
- ウェブでは数多くのSPARQLエンドポイントが公開されており、これらからSPARQLを用いたデータ検索が可能となっている。
 - e.g. [Wikidata Query Service](#)
 - e.g. [Snorql for Japan Search](#)

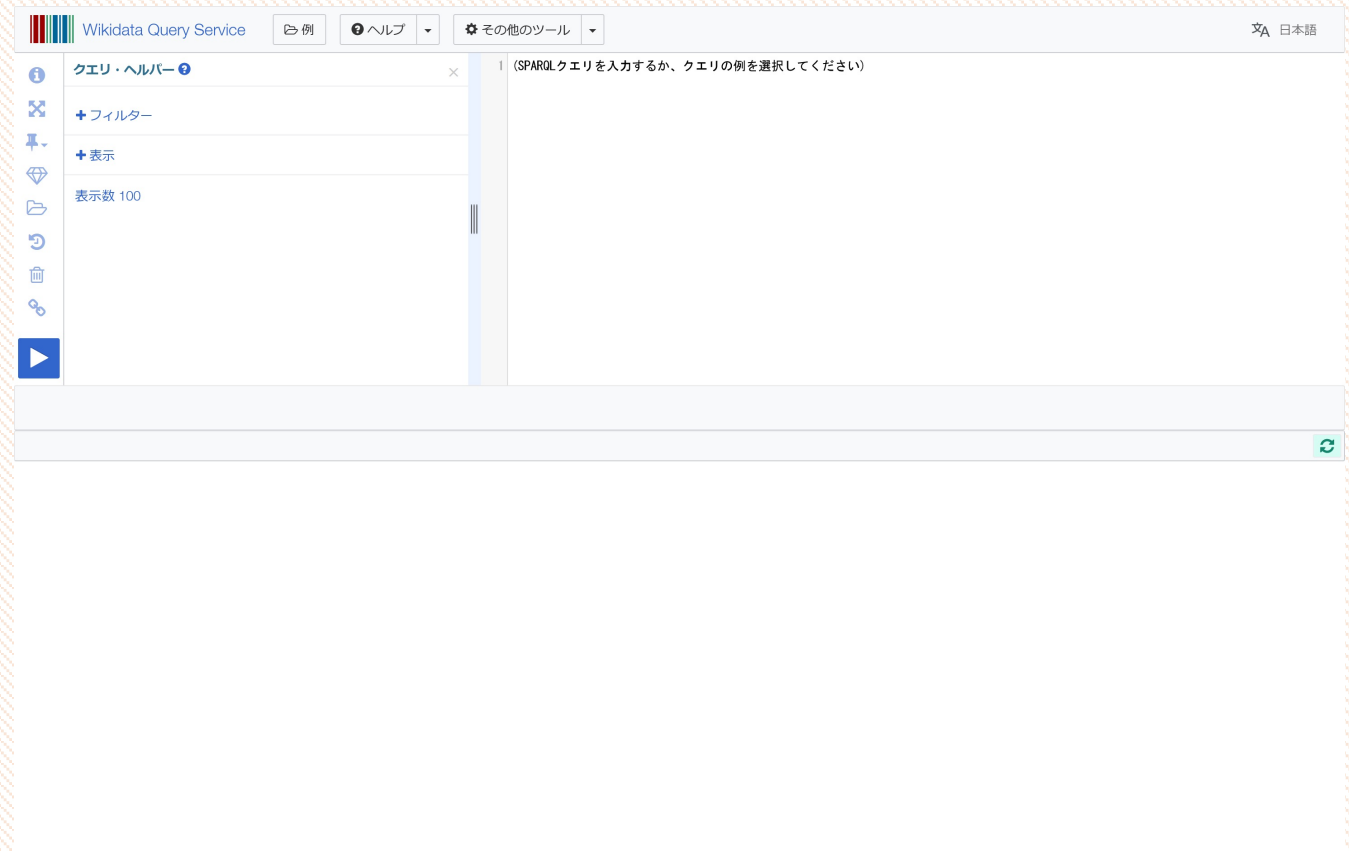


図. Wikidata Query Service

SPARQLを用いた検索サンプル

- SPARQLを用いて、高度な検索が可能となる
- 様々なSPARQLエンドポイントのクエリ（問い合わせ）のサンプルがウェブ上で公開されている。
 - e.g. [Wikidata:SPARQL query service/queries/examples](https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/queries/examples)
 - e.g. [RCGS SPRQLクエリサンプル](#)

SPARQLを用いたデータの指定

- 右のクエリは指定のURIを主語とするリソースのRDFグラフを取得する
 - DESCRIBEは特定のリソースのRDFグラフを応答する
 - 上 : [ジャパンサーチでの問い合わせ](#)
 - 下 : [Wikidataでの問い合わせ](#)

```
DESCRIBE <https://jpsearch.go.jp/data/michi-D0004990094_00000>
```

```
DESCRIBE <http://www.wikidata.org/entity/Q24862683>
```

SPARQLを用いた検索

- 右はSPARQLの基本的な検索パターンに基づくサンプルクエリ ([ジャパンサーチのSPARQLエンドポイント](#)での検索)
 - PREFIXはURIを省略するための接頭辞を定義する
 - ジャパンサーチのeasySPARQLでは、これらの接頭辞はすでに登録されているため入力する必要はない。
 - **SELECT**は変数を定義する
 - **WHERE**はRDFの記述パターンを指定する
 - LIMITは結果の件数の上限の指定

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <https://jpsearch.go.jp/term/type/>
```

```
SELECT ?katana ?label
```

```
WHERE {
  ?katana rdf:type type:刀剣 .
  ?katana rdfs:label ?label .
}
```

```
LIMIT 100
```

Snorql for Japan Search

Description of <<https://jpsearch.go.jp/data/cobas-74743>>

rdf:type	type:刀剣 (刀剣)	
rdfs:label	"太刀 銘 来国光 嘉暦二年二月日"	
schema:name	"太刀 銘 来国光 嘉暦二年二月日"@ja	
schema:creator	chname:来国光 (来国光)	
schema:temporal	time:1327 (1327年)	
schema:dateCreated	"1327"	
schema:category	keyword:国宝 (国宝)	
schema:description	"ID: 74743" "公開: 公開" "分類: 刀剣" "員数: 1口" "新ID: 12f08f3c06a62af80737925634848303/52653" "言語ID: 2"	
schema:image	< https://colbase.nich.go.jp/media/tnm/F-21/image/slideshow_s/F-21_W082G101a.jpg >	
jps:agential	jps:relationType	role:制作 (制作)
	jps:value	chname:来国光 (来国光)
	schema:description	"作者: 来国光"
jps:temporal	jps:relationType	role:制作 (制作)
	jps:value	time:1327 (1327年)
	schema:description	"時代世紀: 鎌倉時代・嘉暦2年(1327)"
	jps:era	time:鎌倉時代
jps:accessInfo	< https://jpsearch.go.jp/data/cobas-74743#accessinfo >	
	rdf:type	type:アクセス情報 (アクセス情報)
	schema:itemLocation	place:東京 (東京都)
	schema:license	< http://creativecommons.org/licenses/by/4.0 > (CC BY 4.0)
	schema:provider	chname:東京国立博物館 <small>(fetched via neo proofs of chname:東京国立博物館)</small>

図. 刀剣のデータサンプル

SPARQLを用いた検索

- 全ページのクエリで指定した変数の値はテーブル形式で取得できる（右図）
- エンドポイントの機能にもよるが、JSONやCSVなどの構造化データのファイル形式でデータを取得することも可能

Snorql for Japan Search

SPARQL results:
10 results

s	label
https://jpsearch.go.jp/data/cobas-12159	"太刀 銘 来国光"
https://jpsearch.go.jp/data/cobas-159108	"太刀 銘 吉房 号 岡田切"
https://jpsearch.go.jp/data/cobas-159109	"太刀 銘 吉房"
https://jpsearch.go.jp/data/cobas-159110	"太刀 銘 一 群鳥文兵庫鎖太刀 (号上杉太刀)"
https://jpsearch.go.jp/data/cobas-159140	"太刀"
https://jpsearch.go.jp/data/cobas-159141	"短刀 銘 行光"
https://jpsearch.go.jp/data/cobas-159144	"太刀 銘 助真"
https://jpsearch.go.jp/data/cobas-159147	"太刀 銘 三条 (名物 三日月宗近)"
https://jpsearch.go.jp/data/cobas-74731	"太刀 銘 定利"
https://jpsearch.go.jp/data/cobas-74743	"太刀 銘 来国光 嘉暦二年二月日"

データ検索演習（1）

1. 前々ページのクエリを試して、レスポンスを取得してみよう
 2. 1) を元にジャパンサーチの「**rdf:type**」（述語）が「**type:映画**」（目的語）のリソースと、「**rdfs:label**」（述語）＝ラベルのリストを取得してみよう。
- Scrapboxの[本講義ページ](#)にリンクがある「[講義用コンテンツ：SPARQL](#)」より、SPARQLエンドポイントのURLと、クエリを取得できます。
 - 各コードの名称（赤にマークされている箇所）の右のファイルボタンを押下するとコピーできます。

SPARQLを用いた検索

- 資料種別 (rdf:type) ではなく、著者 (作成者) 「**schema:creator**」を指定し、その主語 (?novel) とラベル (?label) を取得するクエリ。
 - 結果からは、海外の出版が多いことがわかる。

```
SELECT ?novel ?label WHERE {  
  ?novel schema:creator cname:村上春樹 .  
  ?novel rdfs:label ?label .  
}
```

SPARQLを用いた検索

- より（ちょっとだけ）高度なクエリ
 - OPTIONALは値がある場合のみデータを返す
 - 「;」（セミコロン）で Turtleのように同じ主語のトリプルを繰り返す場合、省略できる
 - 亀甲括弧 [] で、空白ノードを用いた構造的記述の値を指定できる

```
SELECT ?subject ?label ?thumb ?provider
WHERE {
  ?subject rdf:type type:刀剣 .
  ?subject rdfs:label ?label .
  OPTIONAL {
    ?subject schema:image ?thumb .
    ?subject jps:accessInfo [ schema:provider ?provider ]
  }
}
LIMIT 100
```


SPARQLを用いた分析

- 検索結果の件数をリスティング、ソートした結果を示す。
 - countは件数をカウントする
 - order by は表示順序のルールを指定する

```
SELECT (count(?provider) as ?number) ?provider
WHERE {
  ?subject rdf:type type:刀剣 .
  ?subject jps:accessInfo [ schema:provider ?provider ] .
}
order by desc(?number)
```

データ検索演習（2）

- 「**type:絵画**」の所蔵者のリストと各自の所蔵数をレスポンスとして取得してみよう。

データ検索演習（3）

- 演習（3）は**グループワーク**として実施します。
 - 3-4名程度のグループにわかれます。それぞれに名称を付与します。
- これまでに得た知識とサンプルクエリを参考にして、グループで独自のジャパンサーチのSPARQLのクエリを作ってみよう。
 - サンプルクエリはジャパンサーチのクエリ入力欄右にリストがあります。
 - 演習（3）で作成したクエリについて、**1) グループで作成したクエリ結果が生成されたページのURLと、2) そのクエリの説明（どのような機能を持つか）**を、コミュニケーションノートの「グループで作成したクエリ（URL）」と「クエリの説明」から投稿してください。