1G3-OS-08b-1

JavaScript による SPARQL 検索結果の可視化ライブラリ

JavaScript visualization library for SPARQL results

片山 俊明[*1]
Toshiaki Katayama

[*1] ライフサイエンス統合データベースセンター
Database Center for Life Science

In Semantic Web applications, SPARQL query is used to search SPARQL endpoints and results can be obtained as a SPARQL Query Results JSON Format, which is essentially tabular structured data. The D3.js JavaScript library is getting popularity as a generic framework for creating and controlling dynamic graphical representation of data based on the widely accepted Web standards such as SVG, JavaScript, HTML5 and CSS. Therefore, a JavaScript library which combines these two Web technologies, by transforming SPARQL Query Results JSON Format into JSON data structures consumed by the D3.js, would be beneficial to develop dynamic and lightweight application on the Web. The d3sparql.js is a generic JavaScript library to query SPARQL endpoints and to provide various callback functions for visualizing the obtained results. The d3sparql.js library is freely available at https://github.com/ktym/d3sparql.

1. はじめに

セマンティック・ウェブでは、SPARQL によってオープンデータを検索し、結果を可視化するまでをクライアントサイドの JavaScript で行うことができるため、動的で軽量なウェブアプリケーションを容易に開発できる。これを支援するため、D3.js を利用して SPARQL 検索結果の可視化を行う汎用ライブラリを作成した。ここ数年で急速に蓄積されつつある生命情報科学の RDF データへの適用例と合わせて紹介する。

2. ウェブ標準技術に基づくアプリケーション開発

これまでのウェブアプリケーションでは、データを関係データベース (RDB) や NoSQL データベースに格納し、ミドルウェアを利用して、サーバ側で Java や Ruby 言語などにより記述されたロジックを適用した上で、レンダリングされた結果をウェブブラウザに表示するものが多かった。

一方、セマンティック・ウェブによるアプリケーション開発では、ウェブの標準技術だけを用いて、クライアント側の JavaScript から SPARQL によってデータを検索し、その結果をウェブブラウザに表示するまでを完結させられる。

2.1 セマンティック・ウェブ

セマンティック・ウェブでは、URL (IRI) を ID とした RDF データや OWL オントロジーに対し、HTTP プロトコルによる SPARQL 検索を行って、その結果を JSON 形式で得ることができる。これらの全てがウェブ標準技術に基づいているため、ウェブアプリケーションの開発で広く用いられている JavaScript 言語との親和性は高い。ただし、AJAX コールによって得られた SPARQL の検索結果をウェブブラウザ上でどのようにユーザに提示するかは、アプリケーション側で実装する必要がある。このため、SPARQL の結果を可視化できる汎用的なライブラリが利用できれば、アプリケーション開発が容易になると考えられる。

連絡先：片山俊明，ライフサイエンス統合データベースセンター，〒277-0871 千葉県柏市若柴 178-4-4 東京大学柏の葉キャンパス駅前サテライト 6 階，電話 04-7135-5508，Fax 04-7135-5534，ktym@dbcls.jp

2.2 可視化ライブラリ

D3.js (http://d3js.org) は JavaScript のライブラリで、ウェブアプリケーション上でのデータの可視化に広く使われるようになってきている。主に JSON 形式のデータを入力とし、必要に応じて変換・集計などを施したのち、SVG による画像生成を行うことができる。また標準的なウェブ技術にもとづき、HTML/CSS によるデザインと JavaScript によるインタラクティブな機能を作りこむことができる。ただし、現状では可視化パターンごとに前提としている JSON のデータ構造が異なっているため、既存の可視化例を応用する場合にはデータ構造の変換が必要となる。

2.3 SPARQL と可視化の統合

SPARQL 検索の結果を取得する際には、いくつかのデータ形式が指定できるが、その1つが SPARQL Query Results JSON Format である。本質的にはキーと値のセットが配列になった表形式の JSON データであるが、D3.js ではこのデータ構造を直接利用することはできない。そこで、JavaScript から SPARQL 検索を AJAX によって非同期に行う機能と、その結果として得られた JSON 形式を D3.js で利用する際に必要な JSON 形式に変換し、ウェブブラウザ上に可視化する機能を持つ、汎用的なライブラリ d3sparql.js を作成した。

3. データ構造の変換

以下に、例として SPARQL クエリと、結果として得られた SPARQL Query Results JSON Format の JSON のデータ構造、さらに D3.js 用に変換した JSON のデータ構造を示す。

3.1 SPARQL クエリ

下記の SPARQL クエリは、生物種系統のデータベースに対し、クマムシの一群である Hypsibiidae に属する生物種の系統関係を全て取得するものである。

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX up: <http://purl.uniprot.org/core/>
SELECT ?root_name ?parent_name ?child_name
WHERE {
  VALUES ?root_name {"Hypsibiidae"}
  ?root up:scientificName ?root_name .
```

```
  ?child rdfs:subClassOf+ ?root .
  ?child rdfs:subClassOf ?parent .
  ?child up:scientificName ?child_name .
  ?parent up:scientificName ?parent_name .
}
```

ここで、?root は指定した生物系統を、?parent と?child は下位の生物系統分類の親子関係を表す。

3.2 SPARQL Query Results JSON Format

上記の SPARQL クエリを UniProt の提供する SPARQL エンドポイント (http://beta.sparql.uniprot.org/sparql) 等に対して検索すると、下記のような結果が得られる。

```
{
  "head" : {
    "vars" : [ "root_name", "parent_name", "child_name" ]
  },
  "results" : {
    "bindings" : [ {
      "root_name" : {
        "type" : "literal",
        "value" : "Hypsibiidae"
      },
      "parent_name" : {
        "type" : "literal",
        "value" : "Isohypsibius"
      },
      "child_name" : {
        "type" : "literal",
        "value" : "Isohypsibius elegans"
      }
    },
      : (以下繰り返し)
```

注:実際には、現状 UniProt のエンドポイントは JSON 形式での結果取得をサポートしていなかったため、同じデータを用いた自前のサーバで検索を行った。

3.3 D3.js の JSON 形式

樹形図を D3.js で描画する際には、上記の JSON を下記のようなネストしたデータ構造に変換する必要がある。

```
{
  "name": "Hypsibiidae",
  "children": [
    {
      "name": "Isohypsibius",
      "children": [
        {
          "name": "Isohypsibius elegans",
          "size": 1
        },
          : (以下繰り返し)
```

他に、グラフ構造を描画する際には、ノードのリスト (nodes) とエッジのリスト (links) をもつデータ構造への変換が必要となる。

4. d3sparql.js ライブラリ

d3sparql.js では、d3sparql.query(endpoint, sparql, callback) を実行することで、指定した SPARQL エンドポイント (endpoint) に対し SPARQL クエリ (sparql) を AJAX により検索し、結果を指定したコールバック関数 (callback) に渡すことができる。

コールバック関数では、d3sparql.js の提供する可視化パターンを選び、樹形図の場合 d3sparql.sunburst(json) のように関数呼び出しするだけで、前節で例示した JSON のデータ構造変換を行った上で SVG 画像を生成することができる(図)。また、インタラクティブな操作を作りこむことも可能である。

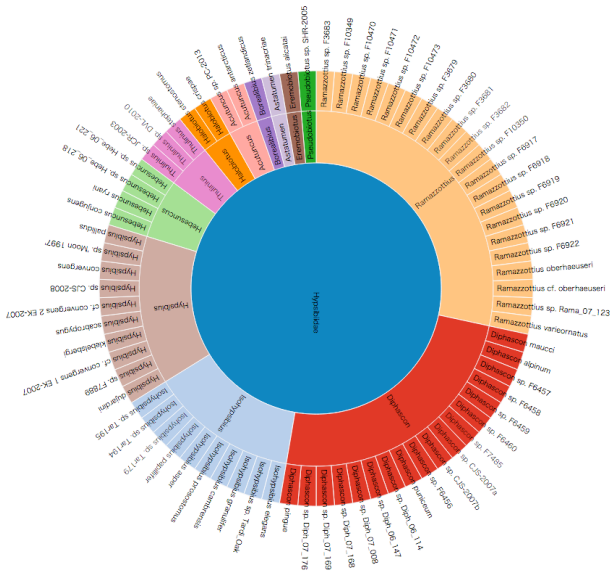

図: d3sparql.js の sunburst による木構造データの可視化例

4.1 可視化パターン

現在、d3sparql.js では、下記の可視化パターンをサポートしている。

パターン	d3sparql.js での関数名
チャート	barchart, piechart, scatterplot
グラフ構造	forcegraph, sankey
ツリー構造	roundtree, dendrogram, treemap, sunburst, circlepack
地図	coordmap, namedmap
表	htmltable

4.2 ダウンロード

d3sparql.js は GitHub にてオープンソースのライブラリとして公開している (https://github.com/ktym/d3sparql) ため、ユーザはダウンロードして自由に利用できるほか、機能改良などの提案を随時受け付けている。

4.3 利用方法

ダウンロードしたライブラリを <script> タグで D3.js ライブラリとともに読み込み、d3sparql の提供する関数を呼び出すことで、さまざまウェブアプリケーションで利用することができる。

```
d3sparql.query(endpoint, sparql, function(json) {
  var config = { ... }              // 描画オプションの設定
  d3sparql.xxx (json, config)  // 可視化パターンの指定
})
```

5. 参考文献

[Bostock 2011] Michael Bostock, Vadim Ogievetsky, Jeffrey Heer: D3: Data-Driven Documents, *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011