



トリプルストアの導入

山本 泰智

データサイエンス共同利用基盤施設
ライフサイエンス統合データベースセンター

2017.8.3 @ COREDO日本橋

FUSEKI

Apacheプロジェクトにより開発されている実装

Apache Jenaを用いたRDFストア
&SPARQLエンドポイント実装

SPARQL1.1 Query / Update / Graph
Store protocolに準拠

<http://jena.apache.org/>

インストールと立ち上げ

- Qiitaのサイトを参考にしてインストール
 - Windows編
<http://tinyurl.com/fuseki-win10>
 - Mac OS編
<http://tinyurl.com/fuseki-MacOSX>
- Qiitaのサイトを参考にしてデータのロード
 - <http://tinyurl.com/fuseki-intro>

SPARQL更新系クエリ

- 参照系とは異なりGRAPHの概念が反映される
- Fusekiに様々なデータを異なるグラフにロード
- 参照系のGRAPHの扱いを確認
- 以降、PREFIX宣言が省略されていることが
ありますのでご注意ください

CREATE / LOAD

グラフの生成とデータのロード

取得対象データがLinked Dataの原則3を満たしている場合は当該URIを指定することで適当なデータを取得できる

PREFIX : <http://example.net>

PREFIX dbpedia-ja: <http://ja.dbpedia.org/resource/>

CREATE GRAPH :myGraph ;

LOAD dbpedia-ja:富士山 INTO
GRAPH :myGraph

残念ながらdbpedia側サーバーの文字コード処理に起因する問題により、現状上記のクエリをfusekiで実行するとエラーになる。例えば、dbpedia-ja:駒ヶ岳などはok。

CLEAR / DROP

グラフやグラフ内のデータを削除

CLEAR GRAPH :myGraph

DROP GRAPH :myGraph

トリプルのないグラフは自動的に消される実装もあり、その場合、両者は同じ

INSERT DATA / DELETE DATA

グラフ内のデータをトリプル単位で挿入や削除
更新対象は具体的なトリプルで指定

INSERT DATA

```
{ GRAPH :myGraph2  
  { :sample foaf:homepage :index.html .} }
```

DELETE DATA

```
{ GRAPH :myGraph2  
  { :sample foaf:homepage :index.html .} }
```

INSERT / DELETE

グラフ内のデータをトリプル単位で挿入や削除
更新対象はパターンで指定

INSERT

```
{ GRAPH :myGraph2 { ?s ?p ?o .} }
```

WHERE

```
{ GRAPH :myGraph1 {
```

```
  ?s ?p ?o .
```

```
  MINUS {
```

```
    ?s owl:sameAs |
```

```
      <http://dbpedia.org/ontology/wikiPageWikiLink> ?o }
```

```
  } }
```


INSERT DELETE

グラフ内のデータをトリプル単位で一度に削除と挿入
更新対象トリプルはパターンで指定

```
WITH :myGraph1
```

```
DELETE { ?s owl:sameAs ?o }
```

```
INSERT { ?s rdfs:seeAlso ?o }
```

```
WHERE
```

```
{ ?s owl:sameAs ?o .
```

```
  FILTER( ! strstarts(str(?o),
```

```
    "http://www.wikidata.org/"))
```

```
}
```

ADD / COPY / MOVE

グラフ内のデータをグラフ単位でコピーもしくは移動

ADD GRAPH :myGraph1 TO :myGraph2

COPY GRAPH :myGraph1 TO :myGraph2

MOVE GRAPH :myGraph1 TO :myGraph2

それぞれの違いは、データの追加元と先のグラフに、処理実行前に存在したデータの扱い

	追加元	追加先
ADD	変化なし	変化なし
COPY	変化なし	削除
MOVE	削除	削除

他のトリプルストア

- Virtuoso
オープンソース版あり
<https://virtuoso.openlinksw.com/>
- Stardog
推論エンジンやデータの検証機能が充実
<http://www.stardog.com/>
- Blazegraph
Wikidataで利用されている
<https://wiki.blazegraph.com/>
- AllegroGraph
Prologを用いた推論機能や高機能クライアントを提供
<https://franz.com/agraph/allegrograph/>
- RDF4J
オープンソース
<http://rdf4j.org/>

関連ツール

- YASGUI
クエリ構築を支援
<http://legacy.yasgui.org/>
- Sparklis
クエリを明示的に書かずにデータを閲覧
<http://www.irisa.fr/LIS/ferre/sparklis/osparklis.html>
- Fuseki
すでに紹介済みですが、手元からSERVICEを用いて好きなエンドポイントにクエリを発行できる
- SPARQLES
様々なエンドポイントを複数の見地から定期的に調査
<http://sparqles.ai.wu.ac.at/>

curlでアクセス

SPARQLの仕様はHTTPを用いたAPIも含む

HTTP GETによるアクセス例

```
$ curl -gLH 'Accept: text/tab-separated-values'  
' http://ja.dbpedia.org/sparql?  
query=select+distinct+*+where+{+%3Chttp%3A%2F%2Fja.dbpedia.org%  
2Fresource%2F%E6%9D%B1%E4%BA%AC%E9%83%BD%3E+%3Fp+%3Fo+.++}+LIMI  
T+100'
```

仕様ではクエリ無しでHTTP GETするとエンドポイントのメタデータが得られることになっているが...

curlでアクセス

HTTP POSTによるアクセス例

```
$ echo ¥  
'select distinct * where { <http://j a. dbpedi a. org/resource/東  
京都> ?p ?o .   } LIMIT 100' ¥  
| curl -gLH 'Accept: text/tab-separated-values' ¥  
--data-urlencode query@- 'http://j a. dbpedi a. org/sparql '
```

課題例

- 人口の多い順に列挙
- 各国の首相を列挙
- 国ごとの最高峰とその高さや場所を列挙
- DBpedia Japaneseの山に関する情報でWikidataにも情報があるものを列挙する
- さらにWikidataからの情報も収集する
エンドポイントは <http://query.wikidata.org/sparql>

国内のエンドポイントを探す

- <http://ja.dbpedia.org/>
- <http://wp.lodosaka.jp/tool/searchdata/>
- <https://data.city.osaka.lg.jp/api/>
- <https://data.e-stat.go.jp/lodw/sparql/>
- <http://id.ndl.go.jp/information/sparql/>
- <http://data.allie.dbcls.jp/>
- <http://lsd.dbcls.jp/>
- <http://integbio.jp/rdf/>

生命科学分野の話題

- 比較的早いうちからRDF化が行われている
 - 参考: <http://id.nii.ac.jp/1001/00163747/>
(情報処理学会員以外は2018年6月まで有料)
- 例のLODクラウドでもそれなりのボリューム
- 国内でも最近活発に研究・開発
- 筆者所属のDBCLSはハッカソンを定期的を開催
 - 参加自由ですので、ぜひ!
 - <http://2017.biohackathon.org/>
 - <http://wiki.lifesciencedb.jp/mw/SPARQLthon>

参考文献

