DM4T Project

Matt Thompson

What is DM4T?

- A project to manage the data produced by TEDDINET.
- We want people to be able to write queries across TEDDINET datasets.
- This way, we can visualise queries such as "All power usage in March for all TEDDINET homes"

First Attempt: Semantic Web Technologies

- We used a tool called "Grafter" to convert CSV data to RDF triples
- This allowed us to store our data in a "triple store" and query them with SPARQL

What is RDF?

"Appliance 2 is an appliance":

```
Subject:
<http://www.cs.bath.ac.uk/dm4t/enliten/appliance/2>

Predicate:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

Object:
<https://w3id.org/seas/Appliance>
```

Another Example

"Reading 15998574 came from house 10":

```
Subject:
<http://www.cs.bath.ac.uk/dm4t/refit/reading/15998574>

Predicate:
<http://purl.oclc.org/NET/ssnx/ssn#hasLocation>

Object:
<http://www.cs.bath.ac.uk/dm4t/refit/home/house_10>
```

SEAS Ontology

SEAS

Smart Energy Aware Systems

We use the Smart Energy Aware Systems (SEAS) Ontology to describe our RDF triples:

`http://ci.emse.fr/seas/`

SEAS also makes use of the Semantic Sensor Network Ontology:

`https://www.w3.org/2005/Incubator/ssn/ssnx/ssn`

SPARQL

SPARQL is a query language for RDF that looks a little like SQL. A basic example is:

```
SELECT ?subject ?predicate ?object
WHERE { ?subject ?predicate ?object }
LIMIT 20
```

Getting a Random Sample of Data

```
SELECT ?value ?time ?r
WHERE {{
  SELECT ?value ?time
  WHERE {
          ?uri rdf:type seas:RelativeHumidity;
                sear:value ?value;
                sear:measurementStart ?time .
  FILTER( year(?time) = 2014 && month(?time) = 1 )
        }}
        BIND ( rand() AS ?r )
        FILTER ( ?r < 0.001 )}
LIMIT 100
```

Federated Querying

```
 SELECT ?s ?p ?o {
   { SERVICE <http://mist.cs.bath.ac.uk/enliten/query>
       { SELECT ?s ?p ?o WHERE
                           { ?s ?p ?o }
                           LIMIT 20 }}
   UNION
   { SERVICE <http://mist.cs.bath.ac.uk/refit/query>
       { SELECT ?s ?p ?o WHERE
                           { ?s ?p ?o }
                           LIMIT 20 }}
}
```

RDF Drawbacks

- Takes many hours of processing to convert every reading in our CSV data to triples
- SPARQL queries can take a long time (> 10 minutes) or time out the triple store service
- Produces about 1.25 billion triples per dataset
- It is difficult to quickly get a random sample of the data readings for visualisation
- Constructing SPARQL queries often requires expert knowledge

File Sizes

Based on the ENLITEN and REFIT datasets:

ENLITEN:

- Original (TSV): 13.3GB
- RDF triples (turtle): ~113GB
- In triple store (TDB): ~107GB

REFIT:

- Original: 6.2GB
- RDF triples (turtle): ~167GB
- In triple store (TDB): ~140GB

New Approach: CSV Streaming

Our new approach:

- Use metadata to describe the contents of each *column* of a CSV, rather than every row
- SPARQL queries then return the CSV file and columns with the relevant data
- The CSV files are then streamed in with irrelevant data ignored

MetaMaker: A Tool for Authoring Metadata

We are building this tool to allow TEDDINET project managers to easily add metadata to their sensor readings

- It infers datatypes and certain metadata from a given CSV path
- The user can change / add metadata as they wish
- Metadata describes both the file as a whole and each column in the file
- Output is converted to RDF triples and added to our triple store service to be queried with SPARQL

MetaQuery: A Tool for Querying Data Readings

This is the next tool we will create

- Given a list of remote CSV files, it will allow for querying and visualisation of data
- The metadata stored in our triple store service tells the tool which CSV information to stream in
- All the heavy work is done in the browser

Summary

- Our goal is to be able to query across TEDDINET datasets.
- Though we have achieved that with our SPARQL / RDF approach, we are now building tools to make the process more efficient