SIMPLE-ML

Towards a Framework for Semantic Data Analytics Workflows

Simon Gottschalk¹, Nicolas Tempelmeier¹, Günter Kniesel², Vasileios Iosifidis¹, Besnik Fetahu¹, Elena Demidova¹

¹ L3S Research Center, Leibniz Universität Hannover
² Smart Data Analytics Group (SDA), Universität Bonn
Motivation & Goal

- Data analytics demands
  - Data science expertise
  - Knowledge about data access, data integration, feature extraction, …
- Many concepts and operations are domain-specific
  - Map matching: connect position data with street segments
- Semantics can support these tasks and simplify data analytics for non-expert users

Adopt semantic technologies to support the
- efficient creation,
- configuration
- and reusability
  of robust data analytics workflows
A semantic data specification is generated based on a data catalog and user-selected operations
- Dataset selection
- Feature selection and extraction
- Dataset integration
- Sampling
- The actual data is materialised later
- Potential subsequent machine learning and result visualisation steps make use of semantic models
Data Catalog & Dataset Profiles

- A **dataset profile** represents dataset characteristics
  - statistics, license, data access information, ...
- A domain-specific **data catalog** contains dataset profiles
- The **attributes** of a dataset are **mapped** to a domain model
Domain Model — Example: Mobility

- description of relevant concepts, their properties and relations in the specific application domain

Example: Mobility

- **Locations**
  - Position representations
- **Traffic statistics**
  - Floating car data, …
- **Events**
  - Accidents, warnings, …
- **Weather**
  - Rainfall, temperature
- …

- Reuse of existing vocabularies if possible
Benefits of Semantics in Data Analytics

- The data catalog with domain-specific dataset profiles is used …
  - … to support the user in generating a data specification
    - extraction of domain-specific features
    - suggest semantically meaningful joins
  - … to materialise the data
    - data access information in the catalog schema
  - … to configure the machine-learning workflow
    - type checking based on the domain model
  - … to visualise results
    - domain-specific visualisations (e.g. rainfall vs. temperature)
Example Data Catalog in the Mobility Domain

sml:SimpleMLCatalog a dcat:Catalog;
dcat:dataset sml:FCDDataset .

sml:FCDDataset a dcat:Dataset;
dcterms:title "Floating Car Data";
dcterms:temporal [ 
  so:startDate "2017-08-01"^^xsd:date;
  so:endDate "2017-12-31"^^xsd:date ];
sml:hasAttribute sml:FCDDatasetAttribute1 .

sml:FCDDatasetFile a sml:TextFile;
dcterms:format "text/comma-separated-values";
csvw:separator ";" .

sml:FCDDatasetAttribute1 a sml:Attribute;
rdfs:label "vehicle id"@en;
sml:columnNumber "0"^^xsd:integer;
sml:hasMapping [ 
  sml:mapsToProperty sml:carId;
  sml:mapsToDomain sml:FloatingCarDataPoint ] .
Example: Traffic Speed Prediction (Mobility Domain)

- Task: Traffic speed prediction for a specific street segment at a given time
- The user selects two datasets:
  - Floating car data (historic positions and speed of vehicles)
  - OpenStreetMap (map data with street segments)
- The user selects and extracts a set of features
- Data materialisation based on the resulting data specification

### Example instances

<table>
<thead>
<tr>
<th>Type</th>
<th>Speed</th>
<th>Time (day)</th>
<th>Time (hour)</th>
<th>Type</th>
<th>Max Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truck</td>
<td>74</td>
<td>Sunday</td>
<td>23</td>
<td>motorway link</td>
<td>80</td>
</tr>
<tr>
<td>Car</td>
<td>84</td>
<td>Sunday</td>
<td>16</td>
<td>motorway</td>
<td>none</td>
</tr>
<tr>
<td>Truck</td>
<td>17</td>
<td>February</td>
<td>8</td>
<td>secondary</td>
<td>70</td>
</tr>
</tbody>
</table>
SIMPLE-ML

Towards a Framework for Semantic Data Analytics Workflows

Thank you!

Questions?
SELECT
?columnNumber ?attrName ?mapProperty ?mapDomain
WHERE {
  sml:FCDDataset sml:hasAttribute ?attribute .
  ?attribute dcterms:identifier ?attrName .
  ?attribute sml:columnNumber ?columnNumber .
  OPTIONAL {
    ?attribute sml:hasMapping [
      sml:mapsToProperty ?mapProperty ;
      sml:mapsToDomain ?mapDomain ;
    ] .
  }
}