Module 13
Semantic Technology and Linked Open Data: Basics, Tools, and Applications

Semantic Technology and Linked Data Annotation
About This Tutorial

- Understand RDF, linked data and SPARQL
- See the semantic technology in practice
- Create semantic annotations
- Index and search semantic annotations with MIMIR
# Module 13 Outline

<table>
<thead>
<tr>
<th>Time</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00-10.45</td>
<td>1. Introduction - (35 minutes)</td>
</tr>
<tr>
<td></td>
<td>2. Processing RDF Data - (35 minutes)</td>
</tr>
<tr>
<td></td>
<td>3. Linked Data (30 minutes)</td>
</tr>
<tr>
<td></td>
<td>4. SPARQL Query Language (10 minutes)</td>
</tr>
<tr>
<td>11:00-12:45</td>
<td>5. Query SPARQL Endpoint and Serialize Data (15 minutes)</td>
</tr>
<tr>
<td></td>
<td>6. Populate Gazetteer from LLD Endpoint (20 minutes)</td>
</tr>
<tr>
<td></td>
<td>7. Semantic Annotations and Linked Data (40 minutes)</td>
</tr>
<tr>
<td></td>
<td>8. Query MIMIR and LLD (30 minutes)</td>
</tr>
</tbody>
</table>
Ontotext Company Profile
XYZ announced profits in Q3, planning to build a $120M plant in Bulgaria, and more and more and more and more and more and more and more text...
If It Works, It's Not AI: A Commercial Look at Artificial Intelligence Startups

One can think of “Semantic Technologies” like as AI, made less abstract and more robust, predictable and manageable.
“Semantic technologies” (ST) is a general term for any software that involves some kind and level of understanding the meaning of the information it deals with.

Examples:

A search engine that can match a query for “bird” with a document mentioning “eagle”

A database that will return Ivan as a result of a query for “?x relativeOf Maria”, when the fact asserted was “Maria motherOf Ivan”

A navigation system that is more intelligent than what we are already used to
Ontotext Positioning

Leading semantic technology provider

- Top-5 core semantic technology developer
- Supplying engines and components to vendors and solution developers

Unique technology portfolio:

- **Semantic Databases**: high-performance RDF DBMS, scalable reasoning
- **Semantic Search**: text-mining (IE), Information Retrieval (IR)
- **Web Mining**: focused crawling, screen scraping, data fusion

Good recognition in the SemTech community

Ontotext pages are ranked #1 for “semantic annotation” and “semantic repository” at GYM
RDF Introduction
Types of Data

- Structured
  - DBMS
  - XML
  - Linked Data
  - Catalogues
- None
  - Text
  - HTML

Formal Semantics
Formal Knowledge

ontotext
GATE
So Why No Just Use XML?

So Why No Just Use XML?

No agreement on:
Structure
  is country a:
    object?
    class?
    attribute?
    relation?
  what nesting mean?

Vocabulary
  is country same as nation?

Are the above XML documents the same?
Do they convey the same information?
Is that information machine-accessible?

<country name="UK">
  <capital name="London">
    <areacode>20</areacode>
  </capital>
</country>

<nation>
  <name>United Kingdom</name>
  <capital>London</capital>
  <capital_areacode>20</capital_areacode>
</nation>
Resource Description Framework

• A simple data model for
  • describing the semantics of information in a machine accessible way
  • representing meta-data (data about data)

• A set of representation syntaxes
  • XML (standard) but also N3, Turtle, …

• Building blocks
  • Resources (with unique identifiers)
  • Literals
  • Named relations between pairs of resources (or a resource and a literal)
Data representation: XML vs. RDF

XML Documents

```
<document>
  <person>
    <name>Maria</name>
    <gender>F</gender>
    <relList>
      <rel type="child">Ivan</rel>
    </relList>
  </person>
</document>
```

RDF Representation

```
myData: Maria
ptop:Person

ptop:Male

myData: Ivan
rdf:type

ptop:childOf

ptop:Woman
```
Data representation: RDBMS vs. RDF

### Relational Tables

<table>
<thead>
<tr>
<th>Person</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>Maria P.</td>
</tr>
<tr>
<td>2</td>
<td>Ivan Jr.</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parent</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ParID</td>
<td>ChiID</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Spouse</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S1ID</td>
<td>S2ID</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

### RDF Representation

<table>
<thead>
<tr>
<th>Statement</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>Predicate</td>
</tr>
<tr>
<td>myo:Person</td>
<td>rdf:type</td>
</tr>
<tr>
<td>myo:gender</td>
<td>rdfs:type</td>
</tr>
<tr>
<td>myo:parent</td>
<td>rdfs:range</td>
</tr>
<tr>
<td>myo:spouse</td>
<td>rdfs:range</td>
</tr>
<tr>
<td>myd:Maria</td>
<td>rdf:type</td>
</tr>
<tr>
<td>myd:Maria</td>
<td>rdf:label</td>
</tr>
<tr>
<td>myd:Maria</td>
<td>myo:gender</td>
</tr>
<tr>
<td>myd:Maria</td>
<td>rdf:label</td>
</tr>
<tr>
<td>myd:Ivan</td>
<td>myo:gender</td>
</tr>
<tr>
<td>myd:Maria</td>
<td>myo:parent</td>
</tr>
<tr>
<td>myd:Maria</td>
<td>myo:spouse</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Processing RDF data
Hands-on Sessions Today

- There are 5 practical examples to be completed
- Hands-on could be downloaded from:
  - ...
- The used software is
  - Sesame (LGPL)
  - Gate Developer (LGPL)
  - MIMIR (GPL v2)
  - OWLIM (Commercial, free for research)
  - Linked Life Data service (free and public)
  - Talend Open Studio (GPL v2)
• IDE for development of data transformation jobs
• No programming skills are required
• All used software is integrated as components
• The task will be to select, configure and connect components
TalenD Open Studio Basics

components

<table>
<thead>
<tr>
<th>Subject (string)</th>
<th>Predicate (string)</th>
<th>Object (String)</th>
<th>Graph (String)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

schema
data
data type
Linked Data
The Web Of Data

- Give to all entities in a data source an URL
- Give to all relationships in a data source an URL
- Link between items in different data sources
- Link between terms from different vocabularies
DBpedia

- Linked Data version of Wikipedia
- 3.5 million entities, incl. 410K places, 310K persons, 146K species, 140K organisations, 95K music albums, 50K films, 33K buildings, 15K videogames, 5K diseases
- Descriptions available in 90 languages
- 1 billion triples, 10 million links to external RDF datasets
- Ontology – 260 classes, 1200 properties, 1.5 million instances

http://www4.wiwiss.fu-berlin.de/dbpedia/dev/ontology.htm
Linked Data evolution – Sep 2008
Linked Data evolution – Jul 2009
Linked Data Design Principles

- Unambiguous identifiers for objects (resources)
  - Use URIs as names for things
- Use the structure of the web
  - Use HTTP URIs so that people can look up the names
- Make is easy to discover information about an object (resource)
  - When someone lookups a URI, provide useful information
- Link the object (resource) to related objects
  - Include links to other URIs
Reason-able Views to the Web of Data

- **Reason-able views** represent an approach for reasoning and management of linked data
  - Integrate selected datasets and ontologies in one dataset
    - Clean up, post-process and enrich the datasets if necessary
  - Load the compound dataset in a single RDF repository
  - Perform inference with respect to tractable OWL dialects
  - Define sample queries against the integrated dataset
Link to the Linked Life Data Service.

- **Linked Life Data** is a public RDF warehouse service.
- Integrates more than 25 popular biomedical data sources.
- Specifies many cross data sources semantic mappings.
- Exposes massive amounts of linked data.
Spring of the data Web

Data Integration Levels

**Semantics**
- Generalization/specialization (Nexium vs. Esomeprazole)
- Homonyms, synonyms

**Structure**
- Different metric units
- Aggregation (full name with initials vs full name)
- Schema mismatch and internal path discrepancy

**Syntax**
- File format (CSV, XML, flat file)
- Character encoding (ASCII, UTF-8, UTF-16)
Syntax and Structure Ambiguity

- RDF data model resolves all syntax level ambiguities
  It helps you express all data in a common data model

ID: GRAA_HUMAN_STANDARD; PRT; 262 AA.
AC: P12544; DT: 01-OCT-1989 (Rel. 12, Created)
DT: 01-OCT-1989 (Rel. 12, Last sequence update)
DT: 15-JUN-2002 (Rel. 41, Last annotation update)

DE: Granzyme A precursor (EC 3.4.21.78) (Cytotoxic T-lymphocyte proteinase 1) (Hanukkah factor) (H factor) (HF) (Granzyme 1) (CTL tryptase)

DE: (Fragmentin 1). GN: GZMA OR CTLA3 OR HFSP.
OS: Homo sapiens (Human).

<MedlineCitation Owner="NLM" Status="In-Process" Version="1">21500419</PMID> <DateCreated>2011-04-15</DateCreated> <Article PubModel="Print"> <Journal Issue="Internet"> <ISSN IssnType="Electronic">1520-6882</ISSN> <Volume>82</Volume> <Issue>20</Issue> <PubDate>2010-Oct-15</PubDate> </JournalIssue> </Journal> </MedlineCitation>
Linked Data Mapping

- How well interlinked is the linked data cloud?
  - Many interesting queries are difficult to be expressed in SPARQL
  - String functions could not be index
  - Often there are misplaced identifiers
Linked Data Mappings

- Identified 6 linked data integration patterns
- Define meta-rules to connect resources with various predicates
- Manually controlled process

The blue lines and the blue text of the captions (used either as part of the URI or literals) designate the criteria for linking the information
## Instance Level Identify Alignment

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Semantics</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact match</td>
<td>Transitive equivalence</td>
<td><img src="#" alt="Diagram" /></td>
</tr>
<tr>
<td>Close match</td>
<td>Equivalent only for search purposes</td>
<td><img src="#" alt="Diagram" /></td>
</tr>
<tr>
<td>Broader match</td>
<td>Generalization of a concept</td>
<td><img src="#" alt="Diagram" /></td>
</tr>
<tr>
<td>Narrower match</td>
<td>Specialization of a concept</td>
<td>Inverse of broader match</td>
</tr>
<tr>
<td>Related</td>
<td>Unspecified relation (no real semantics)</td>
<td><img src="#" alt="Diagram" /></td>
</tr>
</tbody>
</table>
What is the Relation with Text Mining?

The molecular basis for the renal compensation to respiratory acidosis and specifically the role of pendrin in this condition are unclear. Therefore, we studied the adaptation of the proximal tubule and the collecting duct to respiratory acidosis. Male Wistar-Hannover rats were exposed to either hypercapnia and hypoxia [8% CO(2) and 13% O(2) (hypercapnic, n = 6) or normal air (controls, n = 6)] in an environmental chamber for 10 days and were killed under the same atmosphere.

Hypoxia

CSP: reduction of oxygen supply to tissue below physiological oxygen. NCI: Having too little oxygen. NCI: A decrease in the amount of oxygen in the blood. Symptoms range from mild (impairment of cognitive function) to severe (seizures and coma). NCI: Status of decreased oxygen in blood, or tissues. -- 2003

View as Triples  Download in JSON  RDF  N3/Turtle  N-Triples
Simple Knowledge Organisation Schema (SKOS)

- SKOS is a common linked data vocabulary
- Serialized as RDF graph
- Published on the web in a to be shared between applications
- Efficient structuring of terms in thesauri

```
llid:C0035204
  skos:broader
    llid:C0004096
      skos:broader
        llid:C0004096
          skos:prefLabel
            Chronic Obstructive Asthma

llid:C0004096
  skos:broader
    llid:C0035204
      skos:prefLabel
        Asthma

llid:C0035204
  skos:broader
    llid:C0035204
      skos:prefLabel
        Respiratory Disease
```
SPARQL Query Language
SPARQL Protocol and RDF Query Language (SPARQL)

- SQL-like query language for RDF data
- Simple protocol for querying remote databases over HTTP
- Query types
  - *select* – projections of variables and expressions
  - *construct* – create triples (or graphs) based on query results
  - *ask* – whether a query returns results (result is true/false)
  - *describe* – describe resources in the graph
Anatomy of a SPARQL a SELECT query

- List of namespace prefixes
  - PREFIX xyz: <URI>
- List of variables
  - ?x, $y
- Graph patterns + filters
  - Group, alternative, optional
- Modifiers
  - ORDER BY, DISTINCT, OFFSET/LIMIT
Querying SKOS Data

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX lld: <http://linkedlifedata.com/resource/>

SELECT DISTINCT ?label ?concept ?top
WHERE {
    ?top skos:prefLabel "Respiration Disorders".
    ?concept skos:broader ?top.
}

Return all “Respiration Disorder” concepts in LLD and all their IDs and labels.