Semantic Web and Image Information Mining

Jose Emilio Labra Gayo
University of Oviedo, Spain
http://www.di.uniovi.es/~labra

The current Web

Current Web = the biggest repository of information ever compiled by Humanity
Designed for direct human consumption
Lots of information available in:
- Natural Language in HTML
  - English, Spanish, Chinese, Italian, etc.
- More and More multimedia
  - Images, audio, video, etc.

Too much data, not enough knowledge
Multimedia on the Web

Large collections of multimedia assets
- Data integration problem
- Most of them driven by stand-alone databases
- Data isolated syntactically and semantically

Need for Interoperability
- Syntactic level
- Semantic level

Syntactic Interoperability

Data formats that we can share
- XML technologies
- Web Services and mashups
Levels of Interoperability

Semantic interoperability
Share meaning / Concepts
Finding and representing semantic links
Standard ways to provide meta-data
Automatically process the content

The Syntactic Web

Pages and links

Unit of information = HTML Page
Links are syntactic = href
The Semantic Web

Data and semantic links

Unit of information = Data

Links are semantic = Properties identified by URI

Towards the Semantic Web

Query: SPARQL
Ontologies OWL
Rules RIF
RDF Schema
Unifying Logic
Proof
Trust
Digital Signature

Semantic web layer cake, by Tim Berners Lee
RDF


Description of resources
- Resources = entities identified by URI
- Binary Relationships between resources
  - Property = global name of the relationship (URI)
  - Subject → Predicate → Object

RDF Triples

Subject
- A resource identified by URI
- Can also be a blank node (bNode)

Predicate
- Global Property identified by URI

Object
- Value of property
- Can be URI, Literal or bNode
RDF Graph Model

Can be represented in N-Triples

@prefix r: <http://example.org#> .

<http://euitio.uniovi.es>  r:name "School of Computer Engineering".
<http://pictures.org/p1.jpg> r:subject r:Building.

RDF is Compositional

RDF graphs can be composed
RDF is Compositional

RDF graphs can be composed

http://euitio.uniovi.es r:name School of Computer Engineering

http://pictures.org/p1.jpg

http://euitio.uniovi.es r:hasPicture


http://uniovi.es r:contains University of Oviedo

http://chemistry.uniovi.es r:hasPicture

Faculty of Chemistry

RDF graphs can be composed

graph1.rdf + graph2.rdf + graph3.rdf

Blank Nodes in RDF

Blank nodes are used to identify things that have no URI!
Example: The author of a web page is a person, not a URI!

http://euitio.uniovi.es r:author John Smith

http://euitio.uniovi.es r:author Mary Jordan

Blank nodes are represented as _:number in N-Triples

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>_:1</td>
<td>r:name</td>
<td>&quot;John Smith&quot; .</td>
</tr>
<tr>
<td>_:1</td>
<td>r:age</td>
<td>&quot;23&quot;^^xsd:positiveInteger .</td>
</tr>
<tr>
<td>_:2</td>
<td>r:name</td>
<td>&quot;Mary Jordan&quot; .</td>
</tr>
</tbody>
</table>
RDF/XML

RDF/XML = serialization of RDF in XML format

Several abbreviations
Difficult to integrate with other XML technologies

```
<rdf:RDF xmlns:s="http://subjects.org#"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns="http://example.org#">
  <rdf:Description rdf:about="http://pictures.org/p1.jpg">
    <subject rdf:resource="http://subjects.org#Building"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://euitio.uniovi.es">
    <name>School of Computer Engineering</name>
    <hasPicture rdf:resource="http://pictures.org/p1.jpg"/>
  </rdf:Description>
</rdf:RDF>
```

RDF as an Integration Language

A lot of information is currently published in RDF

Example:

DBPedia offers RDF triples of more than 80,000 persons, 293,000 places, 62,000 music albums, 36,000 films, etc.

RDF enables better integration of data
Transform the Web from fileserver to database
RDF as an integration language

Example: Flickr Wrappr
http://www4.wiwiss.fu-berlin.de/flickwrappr/photos/Frascati

RDF Schema

Extends RDF with a Schema vocabulary
- Class, Property, Resource,…
- type, subClassOf, subPropertyOf,…
- range, domain,…

RDF Schema enables simple inferences
RDF Schema

Example

```
\( r:hasPicture \) \\
\( rdfs:range \) \( r:Picture \)

Meaning:

if \( x \ r:hasPicture y \wedge r:hasPicture rdfs:range r:Picture \) 
then
\( y \ rdf:type r:Picture \)
```

```
\( r:Teacher \) \\
\( rdfs:subClassOf \) \( r:Person \)

Meaning:

if \( x \ rdf:type r:Teacher \wedge r:Teacher rdfs:subClassOf r:Person \) 
then
\( x \ rdf:type r:Person \)
```

RDFS Inference

```
\( r:hasPicture \) \\
\( rdfs:range \) \( r:Picture \)

\( r:Teacher \) \\
\( rdfs:subClassOf \) \( r:Person \)

\( http://euitio.uniovi.es \) \\
\( r:hasPicture \)
```

```
\( http://pictures.org/p1.jpg \) \\
\( r:hasPicture \)
```

```
\( Mary Jordan \) \\
\( r:name \)

\( r:Building \) \\
\( r:subject \)
```

```
\( r:Teacher \) \\
\( rdf:type \)
```

```
\( http://euitio.uniovi.es \) \\
\( rdf:type \)
```

```
\( http://pictures.org/p1.jpg \) \\
\( rdf:type \)
```

```
\( Mary Jordan \) \\
\( r:name \)
```

```
\( r:Building \) \\
\( r:subject \)
```
SPARQL

Simple Protocol and RDF Query Language

Query language for the semantic web
Graph matching language
A protocol
Defines a way of invoking a service
WSDL description file
HTTP and SOAP bindings
It also defines XML vocabulary for results

Example

```
prefix r: <http://example.org#>
select ?n where {
  ?p r:subject r:Building.
  ?x r:hasPicture ?p .
  ?x r:name ?n .
}
```

"Find names of resources who have a picture whose subject is Building"
SPARQL example

```sparql
select ?n
where
{
  ?p r:subject r:Building.
  ?x r:hasPicture ?p.
  ?x r:name ?n.
}
```

School of Computer Engineering

Faculty of Chemistry

SPARQL & Inference

The RDF Graph may be obtained by inference

```sparql
select ?n
where
{
  ?x r:subject r:Building.
  ?x r:hasPicture ?p.
  ?x r:name ?n.
}
```

Results

Mary Jordan

Mary Jordan
SPARQL

More features
- Limit the number of returned results; remove duplicates, sort them, ...
- Optional subpatterns (match if possible...)
- Specify several data sources within the query
- Construct a graph combining a separate pattern and the query results, or simply ask whether a pattern matches
- Use datatypes and/or language tags when matching a pattern

Obtaining RDF

SPARQL Endpoints offer an integration mechanism
- Big RDF datasets accessible to applications
  - Example: DBPedia

Nowadays Data is mostly in Databases
- It is not feasible to convert all databases to RDF
- More practical to convert on the fly
- Several systems: Oracle 11g, Sesame, ...
RDF and HTML

Problems to embed RDF/XML in (x)HTML
  It can be linked from an HTML page
There are some “scrapers” to extract the structure of web pages and dynamically generate RDF
  Can be a solution for legacy web content
  Not very elegant

2 proposals for a more systematic way:
  GRDDL
  RDFa

---

GRDDL

```
<html xmlns="http://www.w3.org/1999/">
  <head profile="http://www.w3.org/2003/g/data-view">
    <title>University of Oviedo</title>
    <link rel="transformation" href="http://.../dc-extract.xsl"/>
    <meta name="DC.author" content="Mary Jordan"/>
    ...
  </head>
  ...
  <span class="date">2008-01-02</span>
  ...
</html>
```
RDFa

RDFa defines attributes to add meta-data to HTML elements
Similar to microformats

```html
<html xmlns:cal="http://www.w3.org/2002/12/cal/ical#"> ...
<body>
<p class="cal:Event" about="#meetingBcn">
    I will attend a meeting in
    <span property="cal:location">Barcelona</span>, on
    <span property="cal:dtstart" content="20070508T1000+0200">May 8th at 10am</span>
</p> ...
</body>
</html>
```

Ontologies

RDFS does not solve all requirements
Applications need more expressivity and reasoning
In RDFS, it is not possible to create new subclasses

OWL (Web Ontology Language) offers a common language to define ontologies

Ontology = specification of a conceptualization
Specifies classes and their relationships
Shared by different agents
**OWL**

OWL enables the description of new classes

- By enumeration
- Through intersection, union, complement
- Through property restrictions

It is based on **Description Logics**

- Well defined semantics
- A subset of Predicate Logic
  - Limited use of variables
  - Binary predicates = Relationships
  - Unary predicates = Classes

**OWL: Classes by enumeration**

Example: “The European Union is formed by Italy, France, Germany, etc.”

```
EUCountry = { Italy, France, Germany, … }
```

![Diagram of EUCountry and its members]
**OWL: Set theoretic definitions**

Example: *A person is a man or a woman*

Person = Man \(\cup\) Woman

It also has **IntersectionOf** and **complementOf**

**OWL: Property restrictions**

It is possible to define new classes by restricting the property values of some class

Reasoner acts as a classifier
**OWL: Property restrictions**

**Value constraints**
- **some** ($\exists$) value must be from a class
- **all** ($\forall$) values must be from a class

Example: “An european is a person who has nationality from a European Union country”

$$\text{European} = \text{Person} \cap \exists \text{hasNationality EUCountry}$$

$$\forall x (\text{European}(x) \leftrightarrow \text{Person}(x) \land \exists y (\text{hasNationality}(x,y) \land \text{EUCountry}(y)))$$

**Cardinality constraints** (ie, how many times the property is used on an instance?)
- exact cardinality
- minimum cardinality
- maximum cardinality

$$\text{Person} \subseteq \text{hasFather} = 1$$

$$\forall x (\text{Person}(x) \rightarrow \exists y (\text{hasFather}(x,y) \land \forall z (\text{hasFather}(x,z) \rightarrow z = y)))$$
**OWL: Property characterizations**

It is possible to characterize the behaviour of properties:
- Symmetric, transitive, functional, inverse functional, ...

```
transitive(isPartOf)
```

```
isPartOf(#Rome, #Italy)
isPartOf(#Italy, #Europe)
inference
```

**OWL: Datatype properties**

Properties whose range are typed literals

**Example: age**

```
datatypeProperty(#age)
range(#age, xsd:positiveInteger)
CanVote = Person ∩ age > "18"
```

```
Person(#Sergio)
age(#Sergio, "20")
inference
```

```
CanVote(#Sergio)
inference
```
OWL and Unique Name Assumption

Web = Open System

2 different URIs could identify the same object
OWL does not support Unique Name Assumption

Person ⊆ hasFather = 1
hasFather(#peter, #william)
hasFather(#peter, #bill)
Person(#peter)

There is no error in the model
It infers that "#william" and "#bill" are the same

OWL: Open World Assumption

Traditional systems used Closed World Assumption
OWL uses the Open World Assumption

Singleton = ¬∃ isMarriedWith Person
Married = ∃ isMarriedWith Person

It infers: Married(#Mary)
It does not infer:
Married(#Peter)
Singleton(#Peter)
It also infers that James
is married with someone...
but it does not know with whom
OWL Layers

OWL was defined in 3 layers:
  * OWL Full:
    - No constraints
    - Superset of RDFS
    - Undecidable
  * OWL DL (DL comes from Description Logics)
    - Classes and individuals are separated
    - No characterization of datatype properties
    - Decidable
  * OWL Fragments
    - Subsets of OWL DL more tractable
    - Examples: OWL Lite, DLP, EL++, etc.

OWL 1.1

An extension of OWL (*in development*)
It is based on more expressive DL
More property characterization possibilities:
  - Reflexive, Irreflexive, Antisymmetric
Increased datatype expressivity
  - N-ary datatypes
  - User-defined datatypes
Annotations and meta-logical statements
The name of the game

S often used for ALC extended with transitive roles (R+)
Additional letters indicate other extensions, e.g.:

H for role hierarchy (e.g., hasDaughter ⊆ hasChild)
O for nominals/singleton classes (e.g., {Italy})
R for reflexive properties (e.g., knows)
I for inverse roles (e.g., isChildOf = hasChild–)
N for number restrictions (e.g., ≥2 hasChild, ≤3 hasChild)
Q for qualified number restrictions (e.g., ≥2 hasChild.Doctor)
F for functional number restrictions (e.g., ≤1 hasMother)

S + role hierarchy (H) + inverse (I) + QNR (Q) = SHIQ
SHIQ is the basis for W3C’s OWL Web Ontology Language

OWL DL = SHIQ extended with nominals (i.e., SHOIQ)
OWL Lite = SHIQ with only functional restrictions (i.e., SHIF)
OWL 1.1 = SROIQ

Rules

Rules based systems have a long tradition
They can extend OWL expressivity

Examples:

uncle(?x,?y) ← brother(?x,?z),parent(?z,?y)
older(?x,?y) ← age(?x,?a),age(?y,?b), ?a > ?b.

Proposals:

SWRL = Adds prolog-like rules to OWL
Problem: Adding rules to OWL ⇒ Undecidable
RIF Working group
Uncertainty

Uncertainty handling = critical in practical applications
   Specially in Image Information mining

Several approaches:
   Extend DL with temporal and modal operators
   Probabilistic Description Logics
   Fuzzy Description Logics

Some Applications

BOPA Project
   Ontology based search through governmental documents

WESONet Project
   Multimedia information search

MultimediaN E-Culture
   Art collections search & annotation
BOPA Project

Goal: create a “bridge” between citizens and juridical jargon
- We used semantic Web vocabularies and tools
- Applied to Administrative documents
- Large dataset
  - More than 35,000 legal documents
  - 150,000 different terms

Ontology based query expansion
- Pre-Query: Ask user to disambiguate meanings
- Post-Query: Sort results

Ontology based search

1. Match terms with concepts
2. Spread activation
3. Obtain list of words and weights
4. Apply enriched query

<table>
<thead>
<tr>
<th>Word</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>janitor</td>
<td>1.0</td>
</tr>
<tr>
<td>holiday</td>
<td>1.0</td>
</tr>
<tr>
<td>vacation</td>
<td>1.0</td>
</tr>
<tr>
<td>staff</td>
<td>0.5</td>
</tr>
<tr>
<td>collective agreement</td>
<td>0.5</td>
</tr>
<tr>
<td>work day</td>
<td>0.5</td>
</tr>
<tr>
<td>legal contract</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Ontology based search

Available at: http://bopa.fundacionctic.org

WESONet Project

WESO (WEb Semántica Oviedo) group
Annotation and search over Multimedia assets
We apply it to our University domain
…but it could be applied to other domains
WESONet Project

3 levels of definition

Resource → Feature Extraction
Creator → Low-level Descriptions
Users → High-level Descriptions

Core Ontologies → Domain Ontologies

Tag "A" → Tag "B" → Tag "C" → ...

Automatic Annotation

Automatic Process obtains low level descriptions
Objective values: Date of creation, resolution, ...
Feature extraction algorithms
Histogram, textural analysis, ...

Descriptions are linked to core-ontologies
Several MPEG-7 based ontologies
Example: http://comm.semanticweb.org/
Collaborative Tagging

Users provide tags to multimedia-assets
Tags are pseudo-free text
Tag recommendation systems improve quality
Emergent semantics: folksonomies
Users participate in the image tagging process
Tags are not logically consistent
Users have reputation levels

Experts Annotations

The creator of multimedia assets can give high-level descriptions
Descriptions link to concepts in high-level domain ontologies
Difficulty: Connecting different domain ontologies

We are developing/testing algorithms to combine these 3 levels of description
MultimediaN E-Culture

Searching and annotating cross-institutional heritage art collections
- Based on Semantic web technologies
- Interoperability between collections and vocabularies
  - Supports multiple distributed collections
- Works with a large dataset
  - Near 9,000,000 triples
  - 8 vocabularies

Query: "renaissance"
Conclusions

Semantic Web technologies = ready for deployment
  It is easy to publish something in RDF
  There are already huge amounts of data in RDF
  Linking to existing ontologies is already possible

Social barriers have to be overcome
  “Open door” policy
  Use standards
  Connect to others so others can connect to you

A little semantics can have a lot of impact

The End

Questions?

More information:
http://www.di.uniovi.es/~labra