The Semantic Web and its Relevance for Cultural Heritage (Benjamins et al. 2004)[1]

Text Mining for Historical Documents

Peter Stahl

23th February 2012

Outline

- Introduction
  - What is the Semantic Web?
  - Its benefits and how they are accomplished
- Two technologies in a bit more detail
  - Extensible Markup Language (XML)
  - Resource Description Framework (RDF)
- Cultural Heritage and the Semantic Web
  - Ontology of Humanities
  - Semantic Annotation Tool Support

What is the Semantic Web?[2]

- main purpose
  - to convert unstructured Web of documents into structured Web of linked data
  - Tim Berners-Lee*: "[...] data that can be processed directly and indirectly by machines"
  - to find, share, combine, and reuse information more easily

* inventor of internet; director of World Wide Web Consortium (W3C)

Limitations of the Current Web

- web mainly based on documents written in *Hypertext Markup Language* (HTML)
  - describes elements such as titles, paragraphs, lists, simple URLs and their positions on a website
  - no way to represent arbitrary real world objects and their properties described in those elements
- recent HTML standards encourage use of semantic tags for formatting
  - <*em*> instead of <i> (*emphasis* instead of *italic*)

Semantic Web Technologies

- development of languages specifically designed for data
  - *Extensible Markup Language* (XML)
  - *Resource Description Framework* (RDF)
- technologies are combined to supplement or replace Web content, e.g.
  - *Extensible HTML* (XML + HTML = XHTML)
  - RDF + XML

Extensible Markup Language[3]

- provides rules and elemental syntax for content structure, but associates no semantics with content itself
- superset of HTML
- design goals
  - simplicity
  - generality
  - usability over internet
  - readable by both humans and machines
- widely used in other software as well, e.g. in office applications

XML Properties

- XML document is divided into *markup* and *content*
  - strings constituting markup begin and end with characters **<** and **>**
  - strings which are not markup are content
- most common markup construct is *tag*
  - start-tags, e.g. *<section>*
  - end-tags, e.g. *</section>*
  - empty-element tags, e.g. *<line-break />*

XML Properties

- *attribute* is markup construct within tag
  - consists of name/value pair
  - serves as additional information for a tag
  - **
- content is placed between start- and end-tags
  - may itself contain markup and other nested content → resulting in tree-like structure

```
The university is in Saarbrücken.
```

Example: Plain Text

- sentence in plain format has no particular meaning to computer

```
<sentence>
  The

  <institution>university</institution>

  is in

  <location>Saarbrücken</location>.
</sentence>
```

Example: XML markup

- computer knows that
  - *The university is in Saarbrücken* is a sentence
  - *university* is an institution
  - *Saarbrücken* is a location
- computer doesn‘t know what‘s
  - a sentence
  - an institution
  - a location

```
<sentence>
  The

  <institution
    href="http://www.uni-saarland.de/">
    university
  </institution>

  is in

  <location type="city"
    href="http://de.wikipedia.org/wiki/Saarbrücken/">
    Saarbrücken
  </location>.
</sentence>
```

Example: XML markup with attributes

- problem: *institution* and *location* are common words and can be used differently in markups from other people
- computer must uniquely identify markup elements

```
<sentence
  xmlns="http://example.org./xml/documents/"
  xmlns:pns="http://peter.example.net/xmlns/">

  The

  <pns:institution
    p:href="http://www.uni-saarland.de/">
    university
  </pns:instution>

  is in

  <pns:location
    pns:href="http://de.wikipedia.org/wiki/Saarbrücken/">
    Saarbrücken
  </pns:location>.
</sentence>
```

Example: XML markup with namespaces

- solution: use *uniform resource identifiers* (URI) that identify namespaces with their own tags

Interlude: URIs

- anything can have a URI
- URLs (L = *locator*) both identify and locate web resources, URIs only identify them
  - resource may or may not be accessible over internet
- convention: let Web page describe resource to be identified and let page URL be URI for that resource
  - problem: URI now represents both resource and web page describing it → recurring point of discussion

XML: Pros & Cons

+ high flexibility and legibility
+ can be easily adapted to different domains
+ mainly self-describing (i.e. no previous knowledge of specification or design required to understand what‘s going on)

− high rate of verbosity and complexity
− difficult to map tree model to type systems of programming languages
− does not provide advanced semantics on its own (e.g. relations between objects)

Resource Description Framework[4]

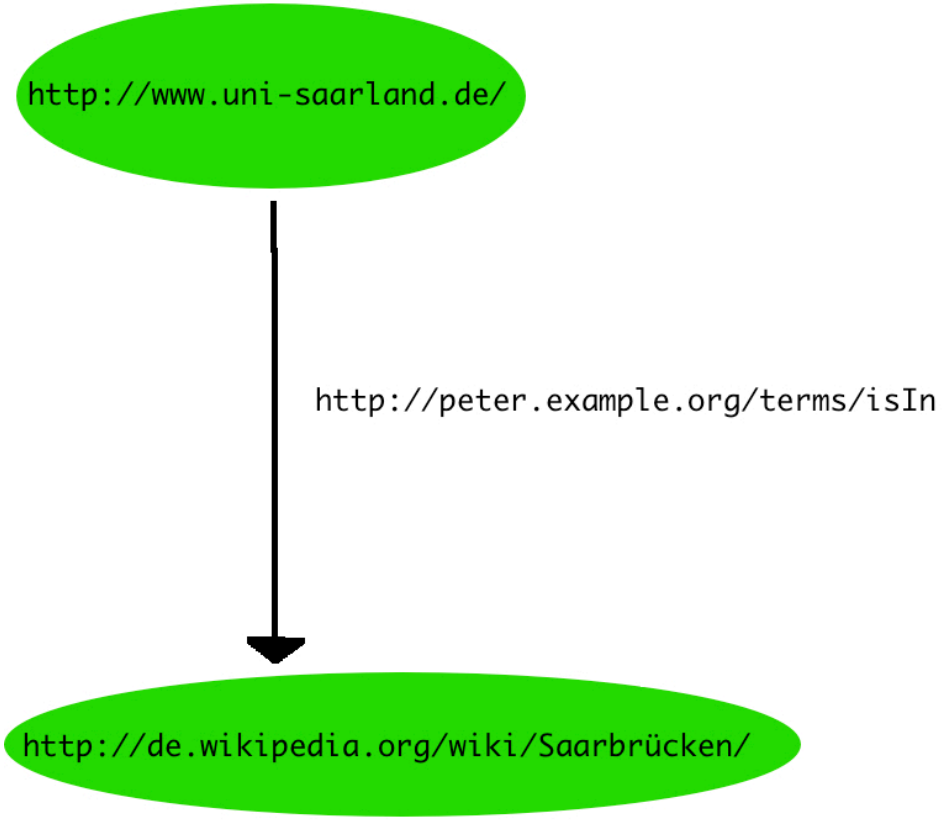
- problem
  - now data is *machine-readable* but not yet *machine-processible*
  - computer cannot yet deal further with data such as doing inference and other post-processing
- solution: RDF
  - statement like simple sentence, almost all words are URIs
  - “anything can say anything about anything“

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:pns="http://peter.example.org/terms/">

  <rdf:Description rdf:about="http://www.uni-saarland.de/">
    <pns:isIn rdf:resource="http://de.wikipedia.org/wiki/Saarbrücken/" />
  </rdf:Description>
</rdf:RDF>
```

Example: XML markup with RDF

- RDF specification defines vocabulary
- can be combined and extended by other namespaces
- uses *subject-predicate-object* expressions (*triples*)
- triples mainly realized by URIs



Example: Minimal RDF graph

- collection of RDF statements represents *labeled directed multi-graph*

Information Sources for RDF

- RDF information usually taken from databases
- URIs given to everything in database
- intelligent programs link data to each other so that complex queries on data can be made

Schemas and Ontologies

- problems again
  - any hard-coded information will go out of date
  - since most terms are ambiguous, computer cannot figure out what main meaning of specific term is
  - restrictions on possible relations and meanings necessary
- solution: schemas and ontologies
  - ontology: explicit specification of abstract concepts and their relations within a domain
  - schema: similar to ontology (in RDF) / specifies document layout including elements, tags, attributes (in XML)

RDF: Pros & Cons

+ low rate of verbosity, high rate of expression
+ specification exists for main vocabulary
+ can be extended to new domains and vocabularies
+ RDF data can be embedded into HTML pages

− which concepts to consider as resources and predicates?
− meaning of existing resources may change over time
− difficult handling of ontology complexity
− resource identification problem (URIs vs URLs)

Semantic Web & Cultural Heritage (Benjamins et al. 2004)

- huge amount of literature works scanned to provide better access
  - information overload: too many databases available with their own attributes and search facilities
- keyword-based search not sufficient, relations between artists and their works necessary
  - → use Semantic Web technologies

Ontology of Humanities

- Basic idea:
  - build ontology of Humanities by involving professionals
  - use it to semantically annotate cultural content
  - support annotation process by “intelligent“ editor
  - publish results on the Web, with semantic navigation, search and visualization
  - provide methodology for helping others publish and exploit their content

Concept	Competency Question
Person	Who wrote 'Cráter'? Every member of PNV Editors of the Gaceta Literaria journal Painters that participated any exposition in Barcelona in 1923
Works	Which works is about Rafael Laffón? What another name is known for work X? Where did X study? Which publications came out from the congress X?

Creation of Ontology

- Competency Question Methodology
  - experts should come up with questions that ontology should provide answers for
  - answers provide concrete inputs for resources and relationships to include in ontology

Creation of Ontology

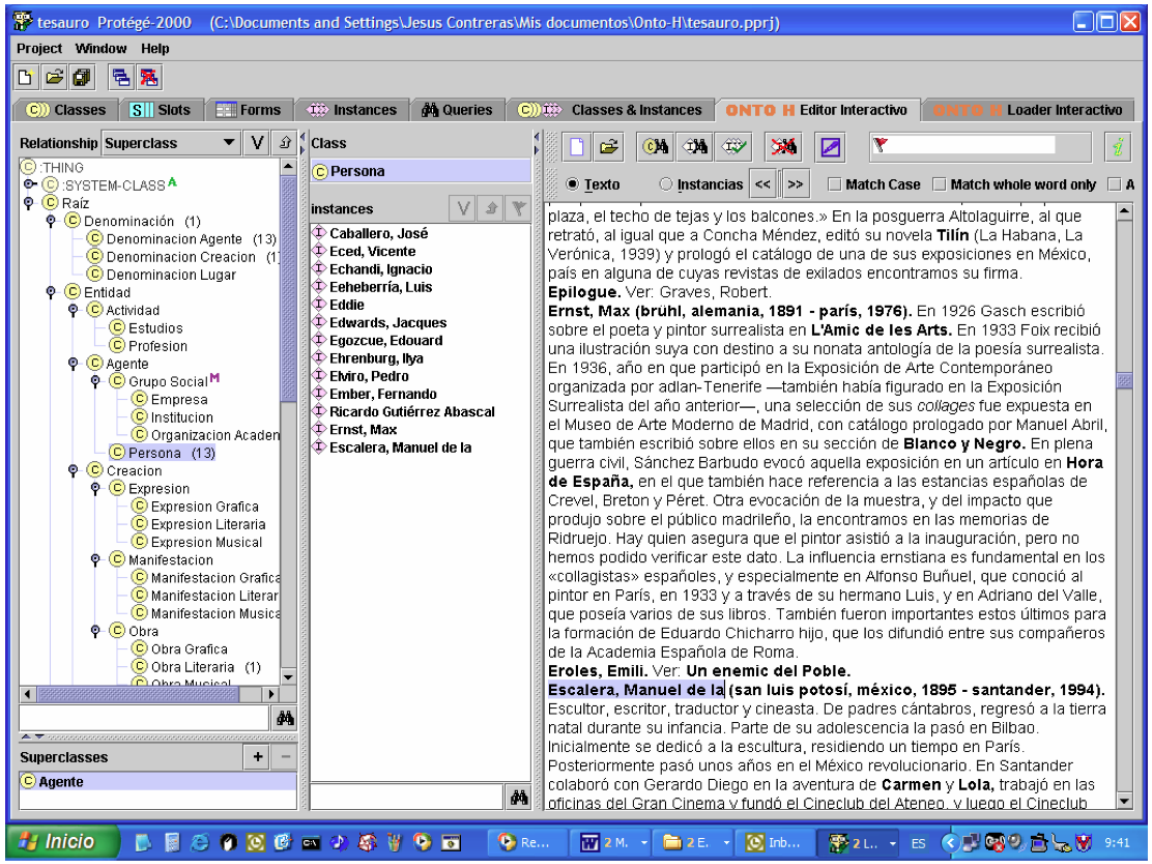
- concepts include:
  - studies
  - profession
  - company
  - institution
  - person
  - movement
  - work etc
- two types of existing ontologies used:
  - general ones to model persons, organizations, events etc
  - specific ones to model things in humanities domain and relations such as *studied_in* and *inspired_by*

Annotation Tool Support

- take structured, semi-structured or unstructured content as input
- provide same content with semantic annotation as output, providing pointers to ontologies
- annotation can be done manually, tool-assisted or fully automatic
- type of annotation depends on rate of content structure
  - more structure → more automation

Annotation Tool Support

- tools have recommendation functionalities for selected words and text parts
- possible actions:
  - add completely new instance to ontology
  - modify existing instance with new occurrence
  - discard any ontology modification
  - more instances in ontology → better recommendations
- tools useful for two kinds of users:
  - knowledge engineer: peforms major changes on ontology
  - annotator: introduces new instances in ontology and maintains existing ones



Ontology Creation with Protégé

- open source
- based on Java
- http://protege.stanford.edu/

Relevance for Cultural Heritage

- identify new relations between authors and their works, intentions, lives
- provide intelligent and convenient access to very complex information
- improve and simplify further research in cultural heritage

Thanks!

References

1) V. R. Benjamins, J. Contreras, M. Blázquez, J. M. Dodero, A. Garcia, E. Navas, F. Hernandez, C. Wert. *Cultural Heritage and the Semantic Web*. The Semantic Web: Research and Applications Lecture Notes in Computer Science, 2004, Volume 3053/2004, pages 433-444
2) http://www.w3.org/2001/sw/
3) http://www.w3.org/XML/
4) http://www.w3.org/RDF/