Semantic Technologies (INFO116)

By Lieselotte De Cock

Inhoudsopgave

1 Role of Semantics and Metadata

Lecture 1 – Book Chapter 1

1.1 What is semantics?

- Semantics stands for capturing the meaning of data
- This is necessary because one thing can have various meanings in different databases

1.2 Why do we need semantics?

- There is a constantly growing amount of data, coming from different web resources
- We are not longer able to store all the data in the world
- But we do want to work with all of it, since it is very valuable
- Important things that we want to do with data are:
  - Search, browse, integrate
  - Gain insights
  - Discover new knowledge
  - …

1.3 Semantic web

- Make all the data in the world look like one big database
- Integrating all of the information about the same objects into a new modality
- Include objects, relationships and semantic searching (instead of text/keyword search)

2 Types and Models of Semantics

Lecture 2 – Book Chapter 2

2.1 Types

- Implicit Semantics:
  - Implicit in the patterns in data
  - Not represented explicitly in any strict machine processable syntax
  - The knowledge about patterns in data may yet be machine processable
  - Examples:
    - Two documents belonging to categories that are siblings of each other in a concept hierarchy
    - Automatic classification of a document
- Formal Semantics:
  - Describing the semanctics of language in purely symbolic terms that is accessible to machines
  - The semantics is described on the syntax of a language
  - The logical consequence relation is formalized in a proof system in terms of axioms and proof rules
  - Features that make a language and its semantics formal:
    - The notion of model theoretic semantics (primitive symbols)
    - The principle of compositionality (meaning of something defined in terms of the meaning of its immediate parts and of their combination)
  - Examples:
    - Subsumption in DL
    - Partonomy
    - FOL

2.2 Models

A semantic model is a way to abstract and relate different pieces of information. This consists of object and concepts, and relationships among them.

2.2.1 Prescriptive Approaches

- Enable specification of formal semantics of symbols from scratch
- The meaning of the symbols representing concepts and relationships is captured through the sntnces in the language of logic

Boolean Logic

- Truth tables
- AND, OR, NAND, NOR, NOT, XOR
- Example: "Cats AND Dogs", "Cats OR Dogs",…

First-Order Logic

- Objects, functions and relationships
- Boolean operators (see above)
- Universal and existential operators
- Example: "All things that are cats which love dogs"

Modal Logic

- Expressions with operators "necessarily" or "possibly"
- Logics for mentalistic notions (BDI – beliefs, desires and intentions)
- Example: "Cats which might like dogs"

Relational Algebra/Calculus

- Finitary relations & joins
- Five primitive closure operators:
    - Selection
    - Projection
    - Cartesian product
    - Set union
    - Set difference
- Serves as a basis for database query languages

Description Logics (DL)

- Family of restricted FOL languages
- To discuss worlds that are organized into objects

Ontology Web Language (OWL)

- Family of knowledge representation languages for modelling ontologies
- Versions: Full, DL, Lite
- Uses RDF and RDFS
-

Resource Description Framework (RDF)

- Standard data model for defining unary and binary relation instances
- Describing metadata for web resources
- RDF statements can be used to define class instances, property instances, collections and properties (through reification)
- A RDF triple exists of:
  - Subject (denotes the resource)
  - Predicate (denotes traits or aspects of the resource and expresses relationship to the object)
  - Object
- RDF example: “Sky – hasColor – Blue”
- Triples are serialized as XML, Turtle, N3 or N-triples

2.2.2 Descriptive Approaches

- Captures contemporary usages of terms (to eventually promote consensus)
- Provides a notation type (see below) that captures the existing use of terms

Folksonomy

- Bottom up classification
- Created by folks (ordinary users)
- Consists of freely selectable keywords or tags, which can be liberally attached to any information resource

Controlled Vocabulary, Thesaurus and Taxonomy

- Provide a way to organize knowledge by grouping together terms corresponding to a concept
- Controlled vocabulary is used to index and catalogue all of this (example: library)
- A thesaurus groups different words by the similarity of their meaning
  (example: Wordnet – lexical database to group words by synset)
- A taxonomy is a hierarchical classification and nomenclature (example: classifying organisms)

2.2.3 General Methods for Creation and Publication of Semantic Models

Linked Open Data (LOD)

- Use URI’s for identification
- Use http URI’s to refer to them and let them be looked up (“dereferenced”)
- Provide information about the thing when it is dereferenced using RDF/XML
- Incluse links to other, related URI’s

3 Ontology and Ontology Development

Lecture 5 – Book Chapter 2

3.1 What is an Ontology?

- An ontology is an explicit (formal) specification of a conceptualisation
- For AI systems: what “exists” can be “represented”
- 2 important properties:
  - Ontologies represent agreement within a domain of discourse (ontology commitment)
  - An ontology provides a terminological agreement or a support for nomenclature which allow for better resolution of terminological differences

3.2 How do we develop an Ontology?

Step 1: Defining the domain and scope

- Questions to answer:
- What domain will we cover?
- For what are we going to use the ontology?
- For what types of quetions should the ontology provide answers? (“competency questions”)
- Wo will use and maintain the ontology?

Step 2: Considering reusing existing ontologies

Step 3: Listing Terms

- What terms do we want to talk about?
- What properties doe these terms have?

Step 4: Defining class hierarchy

- Top-down (danger: starting with too few general concepts)
- Bottom-up (danger: too much specific concepts)
- Combination

Step 5: Defining classes & Subclasses

- How to decide if something has to be seen as “self-standing” or through properties:
  - **Sortal term**: the nouns in natural language (corresponding to types or roles)
    ➔ Self-standing entities
  - **Non-sortal term**: adjectives and verbs in natural language
    ➔ Attached to sortal terms as relationships or attributes
- But, sometimes this distinction is not sufficient, so we introduce **subclasses**:
  - Additional properties that the superclass doesn’t have
  - Can also participate in different relationships than the superclass

Step 6: Defining Properties

- Relations between individuals (example: “color”, “flavour”,…)
- Values of a property:
    - Cardinality (min, max, exact)
    - Type:
        - String, Number, Boolean,…
        - Literal value (“red”) or Instance (object “red” from class “colours”)?

Step 7: Defining Instances

4 Adding Semantics to Data

Lecture 6 – Book Chapter 3

4.1 Different Forms of Data and their Semantics

4.1.1 Unstructured Data

Grammatical Text

- Newspaper, journal, magazine,…
- Full sentences satisfying grammatical constraints
- No technology can understand this "human reader level"
  - ➔ Named Entity Extraction (understanding a restricted set of inputs)

Application Specific, User-Generated Content

- Twitter, Facebook
- Nonstandard spellings (spatial-temporal-thematic context)
  - ➔ Requires background knowledge (in the form of ontologies)
  - ➔ TEchniques and tools to summarize and visualize these contents

4.1.2 Structured Data

- Formal syntax & associated data model
- We need to associate real-world objects/entities and their relationships with the structured data on the basis of their schema

4.2 Approaches to Adding Semantics to Data

4.2.1 Microformats

- We reuse existing HTML/XHTML tags to incorporate metadata
- We embed en encode semantics within the attributes of markup tags
  - ➔ Human readable and machine processable
  - ➔ Semantics are extended to specified and precise domains
- Example: hCard, rel-tag

4.2.2 RDF – in – Attributes (RDFa)

- Attribute-level extensions to XHTML
- Embeds metadata within the Web
  - ➔ Improved traceability and minimized information duplication

4.2.3 Microdata

- HTML5 specification to embed semantics within the Web
- Example: Google, schema.org (ontology for web resources)
- RDFa Lite: minimal subset of RDFa with 5 attributes (vocab, typeof, property, resource & prefix), designed for schema.org

5 Semantics for Social Data

Lecture 7 – Book Chapter 7

5.1 Semantic Social Web (SSocW)

- Key component of Web 3.0
- Organize and analyze social data (ontologies & document level metadata)
- Social Web:
  - User Generated Content
  - Links to other web resources, people and to social networks
- Social Web Data perspectives (about an event):
  - Theme (“what is being said?”)
  - Spatial (“where is it being said?”)
  - Temporal (“when is it being said?”)
    - ➔ Necessary to perform **analysis** of social data
    - ➔ Natural Language Processing (NLP), but this can go wrong so we need other techniques

5.2 Semantic techniques

- Understanding Natural Language is difficult, so we introduce new possibilities
- Ways to improve techniques from NLP (more information below):
  - Named entity identification and disambiguation
  - Normalization and linking of extracted entities/concepts
  - Elimination of spam and off-topic items
  - Aggregation of sentiments
- We need some tools to be able to do this:
  - Knowledge about language and the world
  - A way to combine knowledge sources
    - ➔ Building prohabilistic models from language data

5.3 Creation of Semantic Metadata, Models and Annotations

5.3.1 Disambiguating Entity Mentions

- Deciding on what entities from a text are of interest
- A domain model will indicate certain information to do this (for example the music genre of an artist)

5.3.2 Identifying Entities (based on the entities of interest)

5.3.3 Robustness with Respect to Off-Topic Noise

- We need a focus on related words
- Measures association strength between words, based on a corpus, with the addition of a knowledge base of computer software

5.3.4 Analyzing User Comments

- Example: system that mined music-artist popularity from user comments on MySpace artist pages
- Consisting of:
  - Annotator to spot entities (example artists, albums, tracks)
  - A sentiment annotator to detect sentiment expressions and measure their polarities
- Called “attention metadata” (example above, but also page views, star ratings,…)

5.4 Improving SSocW

5.4.1 Friend of a Friend (FOAF)

- Ontology to describe people and their relationships

5.4.2 Semantically-Interlinked Online Communities (SIOC)

- Fully describing content an structure of social websites
- Facilitate creation and integration of online communities

5.4.3 Semantic MediaWiki

- Extenstion of MediaWiki (which powers Wikipedia)
- Allows users to add structured information to pages

6 Semantics for Enterprise Data

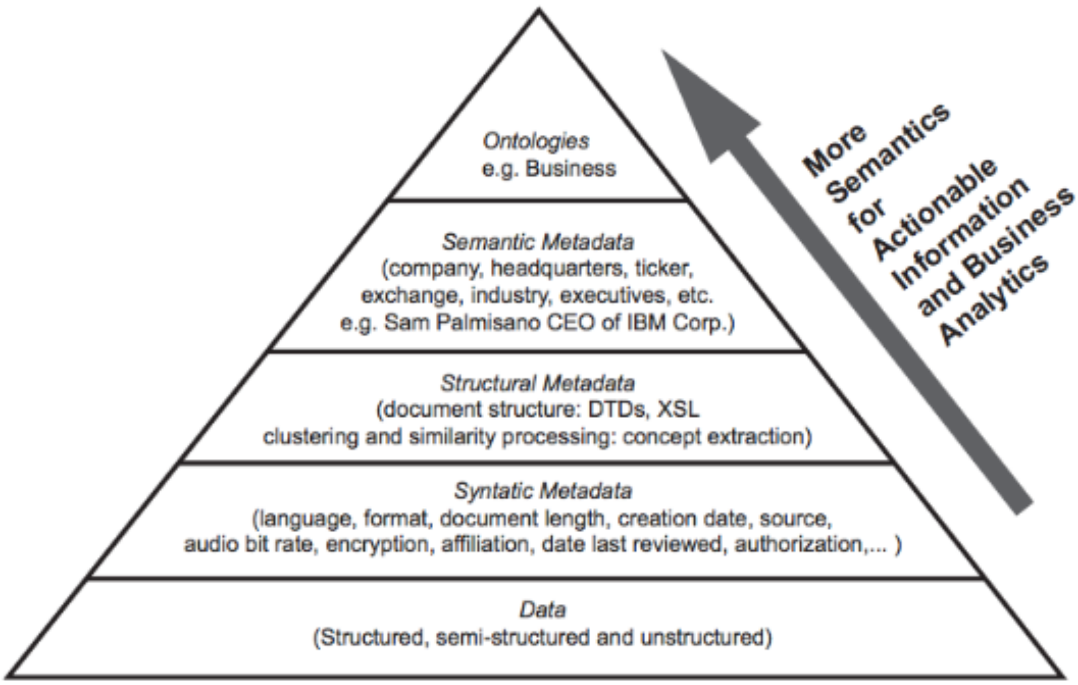
Lecture 8 – Book Chapter 4

6.1 Enterprise System

- Tasks that modern system should include:
  - Extract, organize and standardize information from many sources
  - Identify interesting and relevant knowledge
  - Discover new relationships between documents and entities
  - Fast and high-quality querying, browsing and analyzing
- Main challenges:
  - Excel/PDF
  - Paper forms
  - Departmental Information
  - ….
- Role of semantics in an enterprise:
  - Semantic organization & using metadata
  - Semantic normalization
  - Semantic search
  - Semantic association

6.2 Metadata

6.2.1 Types



6.2.2 Creation

- Find, analyse and tag relevant information
  (documents usually mention references to instances, not semantic abstractions)
    - ➔ The semantic challenge is to infer implicit metadata from the available context
- Techniques used to extract the important data:
    - Dictionaires and thesauri (match words & phrases to recognise terms)
    - Aliases and acronyms
    - Analysing for various patterns and co-occurences (using extraction rules)
    - Ontologies (capturing domain specific knowledge)
- Ontology-driven metadata provides:
    - The basis to build tools to organize and provide a useful description of heterogeneous content
    - Cross-node horizontal relationships between entities
    - Flexibilty and comprehension
- Ontology-driven semantic application development lifecycle:
    - Creation of a schema that serves as the definitional component of the ontology
    - Population of the ontology at the instance level
    - Metadata extraction or semantic annotation od heteregeneous content
    - Blended Semantic Browsing & Querying (BSBQ), to let the user cross-navigate between related knowledge and content
- Semantic Content Organisation and Retrieval Engine (SCORE) System Architecture:
    - 3 different activities who cooperate through XML-based knowledge and metadata sharing:
        - Definition of the World Model and Knowledgebase
        - Content processing
        - Supporting semantic applications
    - ➔ Automatic classification, accurate metadata extraction and extensive mangement of the enhancement processes

6.3 Linking Enterprise Data

- Linked Enterprise Data (LED) helps businesses to solve isolated information silos by using Semantic Web Technologies
  (by cross-organizational data integration and sharing)
- A linked data enterprise is an organization in which the act of information creation in intimately coupled with the act of information sharing
- Linked data is using the Web to connect related data that wasn't previously linked, or to lower the barriers to data linked by other methods:
  - URI's as names for things
  - http URI's so that people can look up these names
  - WHen somebody looks up the URI, provide useful information using RDF & SPARQL
  - Include links to other URI's
- Linked data is "rated" by using star data (1-5 stars)
- Provenance:
  - Rules and regulations for companies, that apply to their internal operations and the products and services they provide
  - 3 categories: Content, management & use
  - Example: Where does the data come from? How did it get there? Can it be trusted, is is machine interpretable?

7 Semantics for Sensor Data

Lecture 9 – Book Chapter 6

!Example application SemSOS on page 92

7.1 Sensor Data

- Sensors used for many purposes (materology, traffic management, satellite images…)
- Generally proprietary, binary data which is difficult to quickly look up
- Very little available communication between sensor streams
    - ➔ Semantic Sensor Web (SSW):
        - o Annotation with Semantic Metadata
        - o Provides meaning to sensor observations
    - ➔ Sensor metadata:
        - o Magnitude, unit of measurement
        - o Location of sensor
        - o Time of measurement

7.2 Sensor Web Enablement (SWE)

- Standardized description files for sensors:
    - All sensors report position
    - All connected to the web
    - All registered with metadata
    - All remotely readable
    - Some remotely controllable
- These standards provide annotations fort he expression of simple metadata (location, timestap, specification…):
    - **Observations & measurements** (O&M – Standard models and XML Schema for encoding observations and measurements from a sensor)
    - **SensorModelLanguage** (SensorML – Standard models and XML Schema for describing sensor systems and processes)
    - **Transducer Model Language** (TML – describing trandsducers and supporting real-time data streaming)
    - **Sensor Observations Service** (SOS – Standard Web service interface for requesting, filtering and retrieving information)
    - **Sensor Planning Service** (SPS – Standard Web service interface for requesting user-driven information)
    - **Sensor Alert Service** (SAS – publishing and subscribing to alerts from sensors)
- Sensor Fusion
    - Multiple sensor data to build evidence for a real-world event
    - Can cut down unnecessary data by tasking only contextually relevant sensors
- Domain general and specific ontologies (spacial, temporal, thematic…):
    - National Institute of Standards and Technology
    - W3C (Geo Markup Language Ontology)
    - OGC SWE (Open Geospatial Consortium)
- Rule-based reasoning to include the Semantic Web Rule Language (SRWL)
  Example: Temperature < 0 degrees -> roads are potentially icy

8 Semantics for Cloud Computing

Lecture 10 – Book Chapter 8

8.1 Cloud Services

- Problem of interoperability, caused by heterogeneity and vendor lock-ins
- National Institute of Standards and Technology (NIST) provides 3 service models:
  - Cloud Software as a Service (SaaS)
    - Consumer can use the application running on a cloud infrastructure
    - Applications accessible from various client devices
  - Cloud Platform as a Service (PaaS)
    - Consumer can deploy consumer-created or acquired applications onto the cloud infrastructure
    - Example: Google Cloud Platform – App Engine
  - Cloud Infrastructure as a Service (IaaS)
    - Consumer can provise processing, storage, networks, etc where he is able to deploy and run arbitrary software
    - Can include operating systems and applications
    - Example: Google Developer Console

8.2 Semantics for Cloud Services

- Challenge is that changing providers and services involves heterogeneity
  - ➔ Core data needs to follow the same semantic concepts:
    - Types:
      - Data (definitions on data structures and relations)
      - Logic/process (core functionality/business logic of an application)
      - Non-functional (QoS, access control, logging…)
      - System (deployment, runtime, service…)
    - Level of abstraction
    - Software life cycle state
- Clouds provide Web services to manipulate resources
- Semantic Annotations for Web Services Description Language (WSDL) standardizes semantic annotations in WSDL service descriptions
- Example: embedded SLA