Semantic Web and Information Extraction Technologies
Raphaël Troncy

Lecture 1: Introduction

Web 2.0: anyone without computer science knowledge can contribute to the web (social network, wiki, etc.)

History of web is divided into 4 decades:

1. The Web: connects information
2. The Social Web: connects people
3. The Semantic Web: connects knowledge
4. The Ubiquitous Web: connects intelligence

RDF: knowledge on the web
RDFS, SKOL, OWL: build our own vocabulary
SPARQL: query the web of data

I. The Web History

Vannevar Bush is a librarian who invented in 1945 a machine called "Memex" (Memory Index) used to locate objects. This was the beginning of hypertexts.
Douglas Engelbart has invented the mouse in 1960.
In the 70's: the email and spam.
In the 80's: first emoticon, the Domain Name System, and the World Wide Web in 1989.
In the 00's: first social media websites, e.g. Facebook, YouTube, Twitter.

II. The Web of Documents

HTML5 in 2014.
XML is a textual format for exchanging structured data using defined tags.

- Well-formed: compiles with the XML format.
- Valid: well-formed which compiles with a DTD or a XML schema (constraints).

Namespaces (URIs) are used to avoid name collision.

III. The Web of Data

We need to provide an explicit representation of the content for machines to understand the current web. Ontology is something that goes beyond XML.

The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. Data should be related as documents and accessed using URIs.

URI (Uniform Resource Identifier) gives a unique identifier to a resource (one URI → one object).
URLs (Uniform Resource Locators) are a subset of URIs, used for resources that can be accessed on the web.

RDF: Resource Description Framework

- Resource: pages, images, videos, etc. anything with URI.
- Description: attributes, features, and relations.
- Framework: model, language, and syntax.

Every piece of knowledge is broken down into (subject, predicate, object). For example (image.png, creator, Raphael). An RDF document is a collection of unordered triples.

Notation3 or N3 is the most known non-XML serialization of RDF. Example:

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
<http://en.wikipedia.org/wiki/Tony_Benn>
   dc:title "Tony Benn";
   dc:publisher "Wikipedia".
```

JSON-LD: JSON for Linked Data

```
{
"@context": "http://json-ld.org/contexts/person",
"@id": "http://dbpedia.org/resource/John_Lennon",
"name": "John Lennon",
"birthday": "10-09",
"member": "http://dbpedia.org/resource/The_Beatles"
}
```

IV. The 4 Linked Data Principles

Tim Berners Lee [2006]

1. Use URIs to identify things.
2. Use HTTP URIs so that people can look up those names.
3. Provide useful data in RDF.
4. Include RDF links to other URIs to discover related information.

For every resource, we define 3 URIs: one for the real object, one for the webpage and one for the RDF description.

Lecture 2:

I. What is an Ontology?

Ontology is the theory of what exists.

In computer science, it is a vocabulary used to describe some domain. It is a formal, explicit specification of a shared conceptualization.

Why? To share common understanding of the information structure among people or software agents, to enable reuse of domain knowledge.

Kind of (is a) & part of (has) relationships!

(more!)

II. Methodologies for Building Ontologies

Ontology life cycle: needs – design – diffusion – use – evaluate – evolution

Building Ontology

- Define classes
- Arrange in taxonomic hierarchy
- Define properties
- Create instances

Step of building an Ontology:

1. Domaine & scope
2. Re-use existing ontologies
3. Enumerate terms
4. Define classes and hierarchy
5. Properties of classes
6. Define slots / attributes / properties
7. Create instances

III. Tools for Building Ontologies

Protégé is a good tool to use among many others.

Linked Open Vocabularies: search existing ontologies.

Lecture 3: Lab Session

Build an ontology.

Lecture 4: SPARQL Basics

SPARQL: RDF query language and data access protocol (implemented in many programming languages).

Query Language + Result Format (XML) + Access Protocol

I. Query Language

There are 4 types of SPARQL queries: SELECT, ASK, CONSTRUCT, DESCRIBE.

PREFIX ... SELECT ... FROM ... WHERE ... ORDER BY ... LIMIT ... OFFSET ...

Example

```
PREFIX ex: <http://www.eurecom.fr/schema#>
SELECT ?person ?name
WHERE {
      ?person rdf:type ex:Person .
      ?person ex:name ?name .
      OPTIONAL {?person ex:age ?age}
      FILTER (?age > 17)
}
```

a can replace rdf:type

We can use CONSTRUCT to complete a graph.

II. Access Protocol

```
GET /sparql/?query=<encoded query> HTTP/1.1
Host: www.eurecom.fr
User-agent: my-sparql-client/0.1
```

Lecture 5: NLP & Information Extraction

Federated search: query many distributed RDFs.

Knowledge Graph: representation of the world entities.

Named Entity Recognition: a task that aims to locate and classify the name of an entity in a textual document.
Named Entity Linking: link a given named entity from a source document to an existing Knowledge Base.
Problems: capitalization, punctuation, shortening, etc.

Language preprocessing:

1. Text Normalization: emoticons and HTML tags, words with low entropy e.g. "and", "or", "a", etc., capitalization, punctuation, shortening, etc.
2. Language Identification: use supervised algorithms, Wikipedia is used as training data.
3. Tokenization: split the text to atomic elements.
4. Part-of-Speech Tagging: attach a grammatical tag to each token.

Named Entity Recognition

1. Grammar-based approach: based on language rule e.g. after preposition + adjective we can find location.
2. Statistical-based approach: supervised classification.

Named Entity Linking

Linguistic pipeline: Named Entity Recognizer (NER) + KB indexing

Lecture 6: NLP & Language Models; Sentiment Analysis; Topic Modeling

Lemma: rough semantics (bank) vs. Wordform: word as it appears in speech (banks)
One lemma can have many meanings (senses)
Homonymy: Homographs (bat/bat) and Homophones (write/right) → Problems for NLP applications
We can introduce the notion of similarity:

- between senses: nearly synonyms (bank and slope)
- between words: related (car and bicycle)

path-based similarity: length of the shortest path between 2 concepts in the hypernym graph
$\text{pathlen}(c_1, c_2) = 1 + \min_p |p|$ where p is a path between c_1 and c_2

$$\text{simpath}(c_1, c_2) = 1/\text{pathlen}(c_1, c_2)$$

Drawback: all the edges have the same weights
→ We define a word as a vector: similar words are somehow nearby
More common: word-word matrix: frequency of other words in a fixed-length window (e.g. 5).
Cosine similarity: $cosine(\vec{v}, \vec{u}) = \frac{\vec{v}.\vec{u}}{|\vec{v}||\vec{u}|}$
Drawback: frequency is useful (sugar appears a lot near apricot), but frequent words like the, it or they will be similar to all words.
→ normalize with the frequency of the word in the document.

Word2vec: instead of using the frequency of words, we use the likelihood of one word to appear in the context of others. Train a binary classifier: is w likely to appear near "apricot"?

Lecture 7: Advanced Semantic Web

Simple Knowledge Organization System (SKOS) is a language used to share vocabulary on the web.
We define concepts (skos:Concept) identified by URIs.

Many label types to refer to the concept, e.g. skos:prefLabel (1 per language), skos:altLabel, skos:hiddenLabel

Ontology Matching: Silk is a tool used to discover links between data items within different Linked Data src. Linkage rules:

- Select values to be compared
- Normalize the values (transform to common formats, e.g. case normalization)
- Compare using similarity measures, e.g. Levenshtein distance.
- Aggregate the result of multiple comparisons, e.g. minimum, average, etc.