



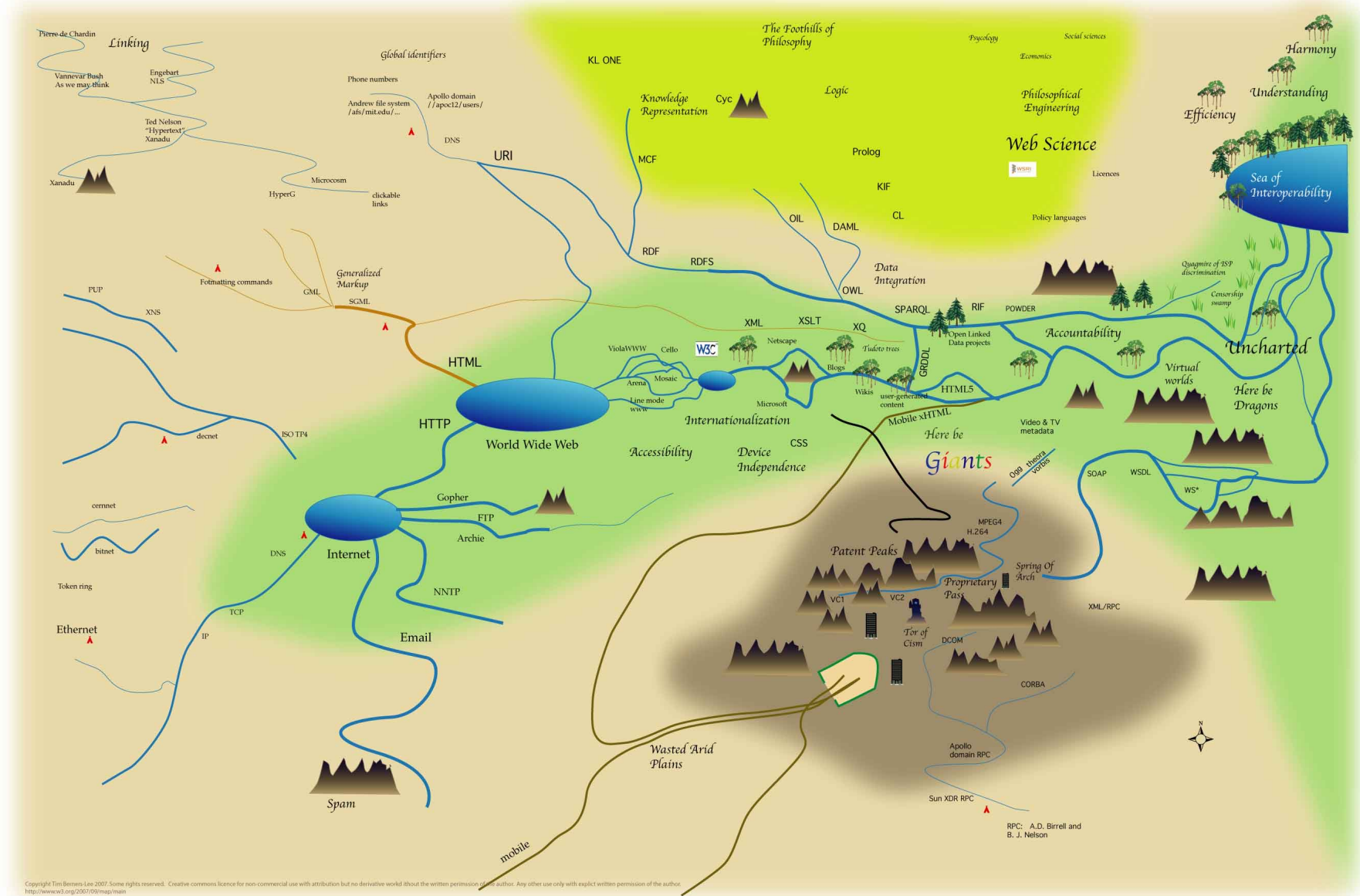
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Semantic Web and Knowledge Graphs

Federico Chesani

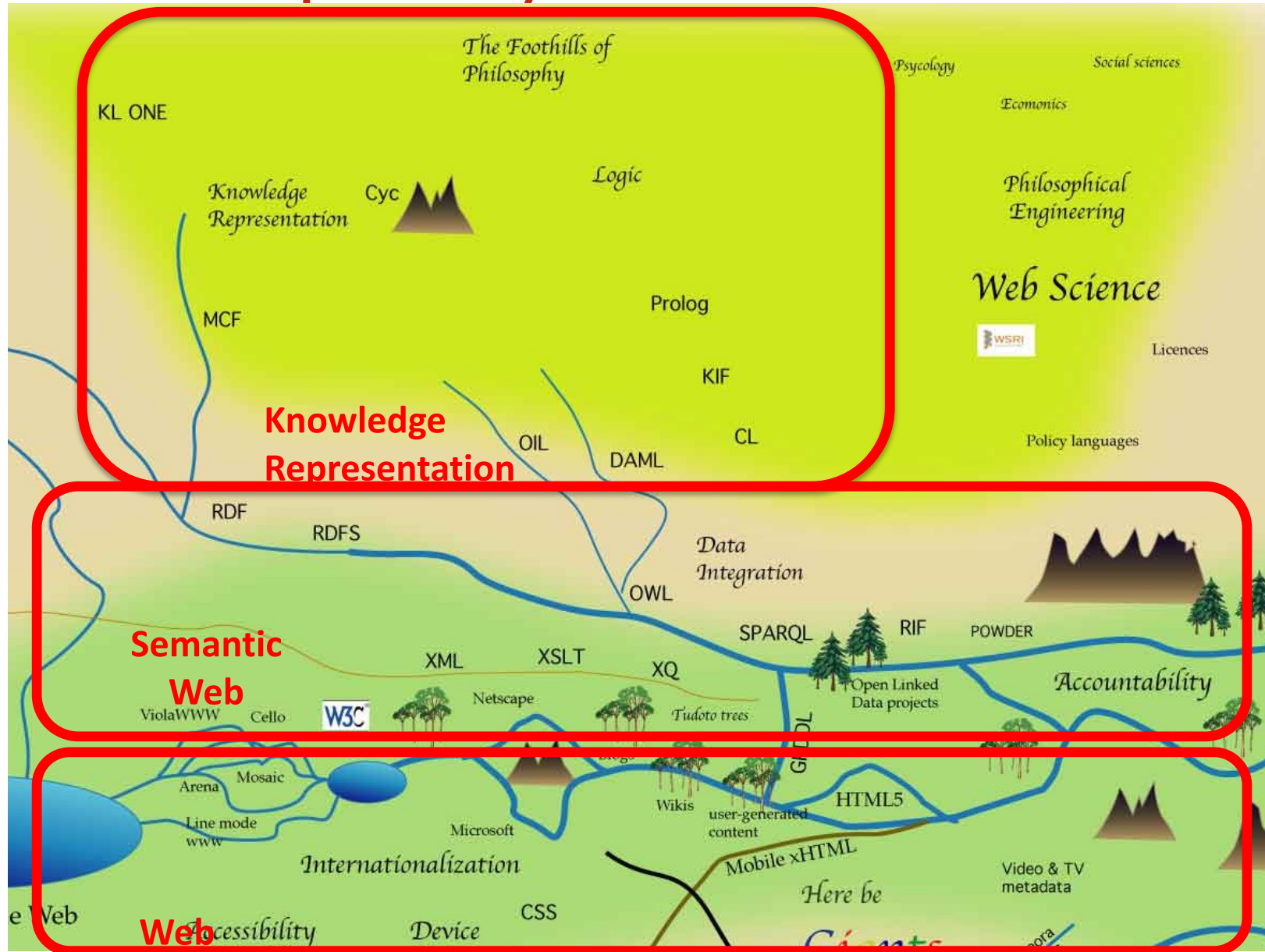
Semantic Web

The Web as depicted by Berners-Lee in 2007



©Tim Berners-Lee, <http://www.w3.org/2007/09/map/main.jpg>

The Web as depicted by Berners-Lee in 2007



©Tim Berners-Lee, <http://www.w3.org/2007/09/map/main.jpg>



The Web 1.0

Information represented by means of:

- Natural language
- Images, multimedia, graphic rendering/aspect

Human Users easily exploit all this means for:

- Deducting facts from partial information
- Creating mental associations (between the facts and, e.g., the images)
- They use different communication channels at the same time (contemporary use of many primitive senses)



The Web 1.0

The content is published on the web with the principal aim of being “human-readable”

- Standard HTML is focused on *how* to represent the content
- There is no notion of *what* is represented
- Few tags (e.g. <title>) provide an implicit semantics but ...
 - ... their content is not structured
 - ... their use is not really standardized



The Web 1.0

Web pages contain also links to other pages, but ...

- No information on the link itself ...
 - ... what does a link represent?
 - ... what does the linked page/resource represent?
- E.g.: in my home page there are links to other home pages ...
 - Which ones link to colleagues?
 - Which ones link to friends?



The Web 1.0

Actual Web = Layout + Routing

The problem: it is not possible to
automatically reason about the data

But the web is indeed an immense knowledge base... it's there, it's free...



The Web 1.0

The web is *global*

- Any page can link to anything
- Approximatively, anyone can publish anything on the web, about any topic
 - *Distribution* of the information
 - *Inconsistency* of the information
 - *Incompleteness* of the information
- Some recent attempts to limit such freedom (with mixed results)



Semantic Web

Goal: “*use*” and “*reason upon*” all the available data on the internet *automatically*

How? By *extending* the current web with *knowledge* about the content (*semantic information*)



Semantic Web

“The Semantic Web is about two things. It is about common formats for integration and combination of data drawn from diverse sources, where on the original Web mainly concentrated on the interchange of documents. It is also about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing.”

Source: W3C Semantic Web Initiative



Semantic Web

Principles SW would like to preserve:

- Globality
- Information distribution
- Information inconsistency
 - Content inconsistency
 - Link inconsistency
- Information incompleteness
 - ... of contents
 - ... of routing information (links)



Adding information about the content...

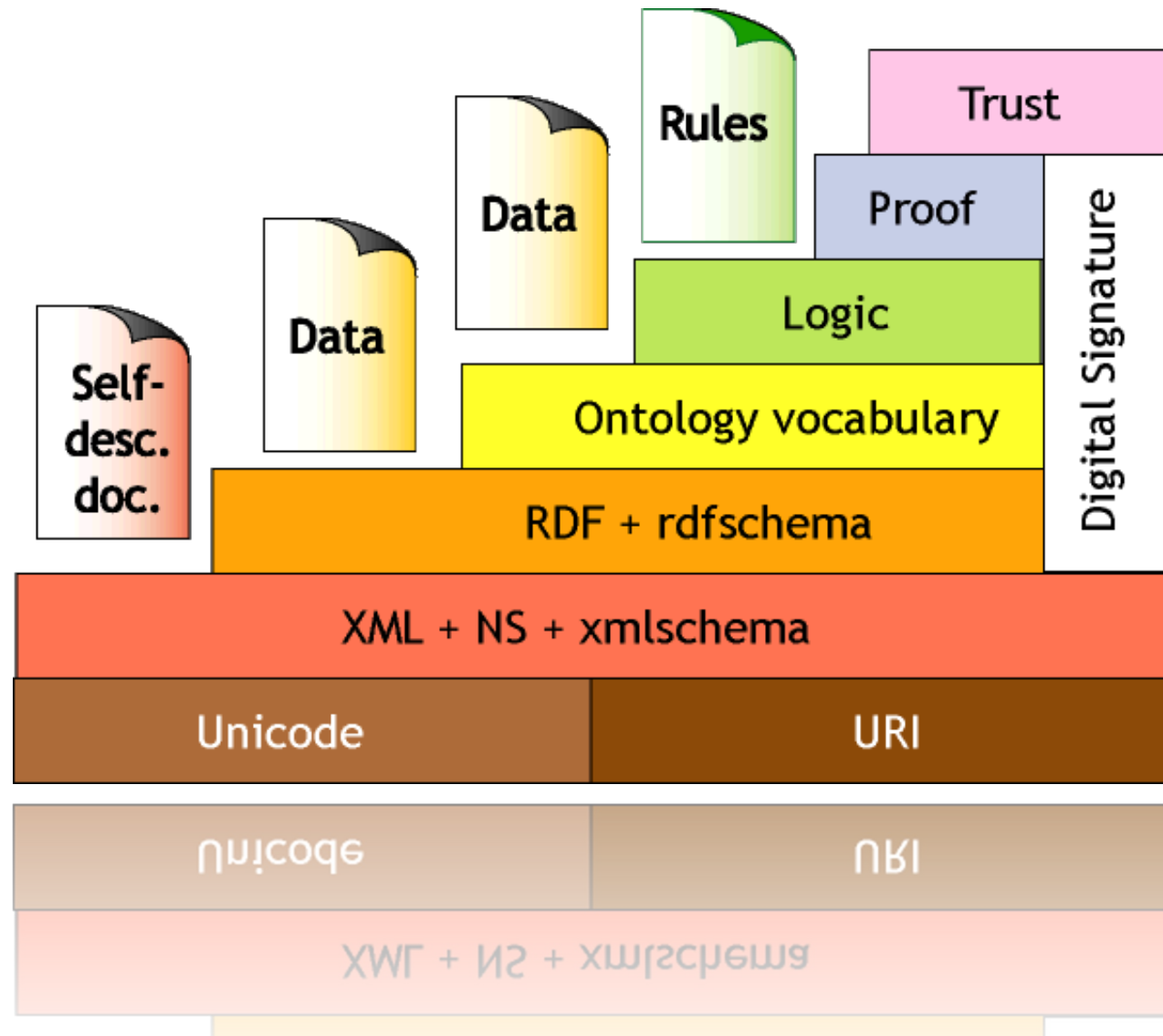
Adding information is not enough

- Information should be structured (e.g., Linneo classification for the living world)
 - *Ontologies!!!!*
- There is the need of some inference mechanism (e.g., sillogism, FOL, DL)
 - *Logic!!!! (and DL in particular)*
- We should be able to infer new knowledge
 - We need the *proofs* that originated such new knowledge



Semantic Web Tools

Recalling the Semantic Web Cake



A unique way for identifying concepts

- How to uniquely identified concepts?
 - > by means of a name system ...
- SW exploits an already available name systems, URIs (*Uniform Resource Identifier*)
 - By definition, URI guarantees unicity of the names
 - To each URI corresponds *one and only* one concept ...
 - ... but more URI can refer to the *same* concept!
 - NOTE: differently from the web, it is not necessary that to each URI corresponds some content!

Examples:

<http://www.repubblica.it>

federico.chesani@unibo.it

ISBN 88-7750-483-8

eXtensible Markup Language - XML

- Created for supporting data exchange between heterogeneous systems (hardware and software)
 - No presentation information
 - Human readable and machine readable
- Extensible, so that anyone can represent any type of data
- Hierarchically structured by means of *tags*
- An XML document can contain, in a preamble, a description of the grammar used in such document (optional) (self-describing document!!!)
- Very mature technology!

Resource Description Framework (RDF/RDFS)

- Standard W3C
- XML-based language for representing “knowledge”
- A design criteria: provide a “minimalist” tool
- Based on the concept of triple:

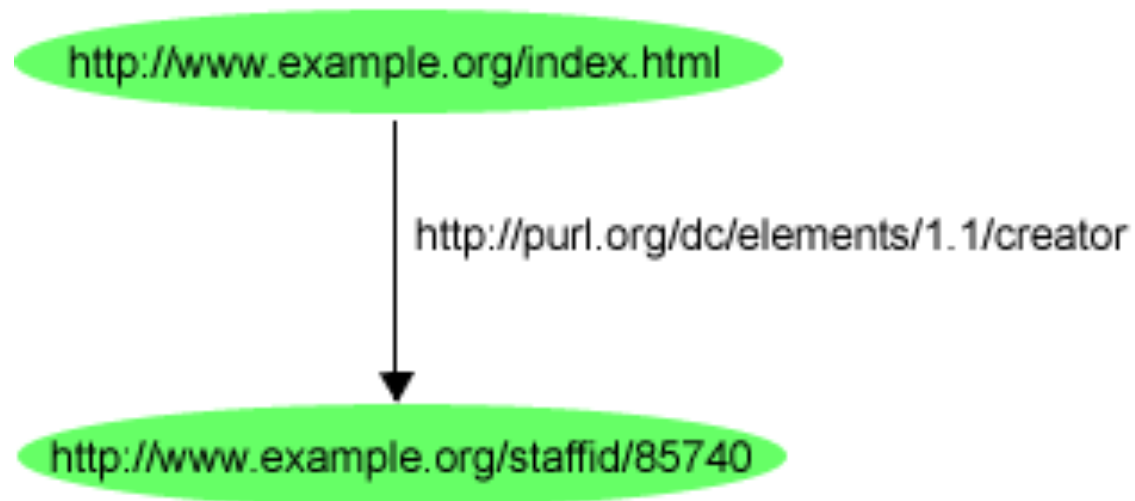
< subject, predicate, object >

< resource, attribute, value >

- Some different representations (N3, Graph, RDF/XML)

RDF – Graph Representation

- A node for the subject
- A node for the object
- A labeled arc for the predicate



`http://www.example.org/index.html` has a creator
whose value is John Smith

RDF – Graph Representation



RDF – XML Representation

```
<rdf:RDF
```

```
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

```
  xmlns:contact=http://www.w3.org/2000/10/swap/pim/contact#
```

```
>
```

```
  <contact:Person   rdf:about="http://www.w3.org/People/EM/contact#me">
```

```
    <contact:fullName>Eric Miller</contact:fullName>
```

```
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
```

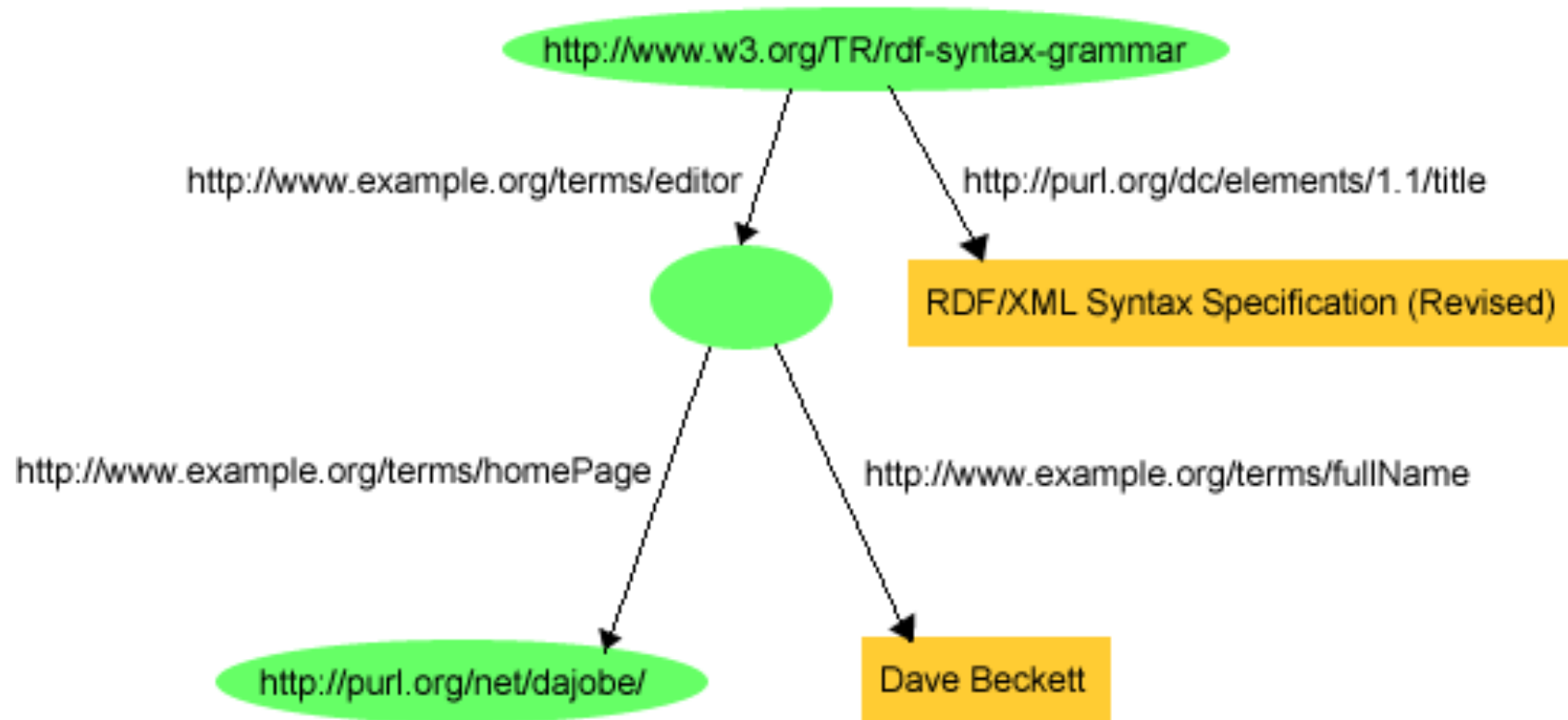
```
    <contact:personalTitle>Dr.</contact:personalTitle>
```

```
  </contact:Person>
```

```
</rdf:RDF>
```

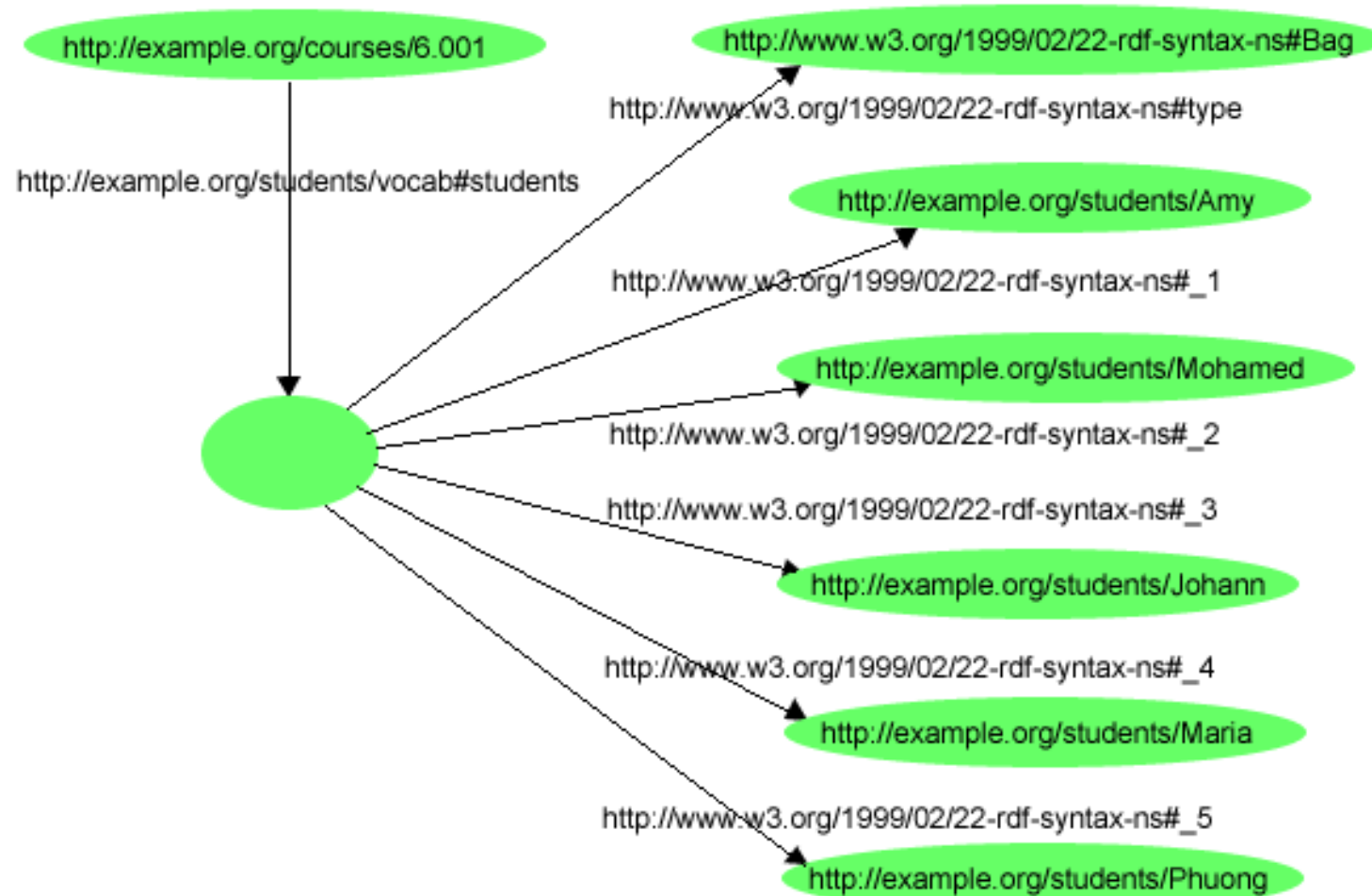
I can query for the mailbox of Eric Miller, without knowing a priori if he uses mailbox or email ...
... if Eric Miller will change mailbox, search result will be coherent!

RDF - Examples



Empty Nodes

RDF – Examples



Bags/Sets

RDF – Expressive Power

RDF supports:

- **Types** (classes) by means of the attribute **type** (that assume as value an URI)
- Subject/object of a sentence can be also **collections** (bag, sequence, alternative)
- **Meta-sentences**, through *reification* of the sentences (“Marco says that Federico is the author of web page xy”)

RDF Schema

- RDF can be intended also as a description of resource attributes and of the values of such attributes
- RDFS allows to describe classes and relations with other classes/resources
 - *type*
 - *subClassOf*
 - *subPropertyOf*
 - *range*
 - *domain*

RDF and E/R Models

- Many similarities with E/R models ...
 - ... RDF is more expressive
- RDF to be intended as the “E/R” for the web
- Relations in RDF are “first class entities”
- In RDF the list of properties of an entity is not:
 - A priori determined by the developer
 - Centralized (DB)
 - Consequence of the fact that any one can assert anything about any one else

RDF and Relational Databases

There is a direct mapping with relational db

- A record is viewed as a RDF node
- The name of a table column is viewed as `rdf:propertyType`
- The corresponding field value is intended as the value of the property
- RDF aims to integrate different databases with different underlying model
 - Traditional DBMS are optimized for creating new data models within the same db or within a restricted set of dbs

RDF Tools

Many tools already available ...

Only in the W3C wiki there are citations for:

- 38 Frameworks/reasoners
- 47 RDF Triple Stores

Have a look to

<http://www.w3.org/2001/sw/wiki/Tools>

RDFa

- RDFa is a specification for attributes to express structured data in XHTML.
- The rendered, hypertext content of XHTML is reused by the RDFa markup
 - publishers don't need to repeat significant data in the document.

Source: RDFa Primer

<http://www.w3.org/TR/2008/NOTE-xhtml-rdfa-primer-20081014/>

RDFA

```
...  
All content on this site is licensed under  
<a href="http://creativecommons.org/licenses/by/3.0/">  
    a Creative Commons License  
</a>.
```

```
...  
All content on this site is licensed under  
<a rel="license" href="http://creativecommons.org/licenses/by/3.0/">  
    a Creative Commons License  
</a>.
```

This page has a **relation** of type **license** with the page at creative commons...

Source: RDFA Primer

<http://www.w3.org/TR/2008/NOTE-xhtml-rdfa-primer-20081014/>

RDFA

```
...  
<div>  
    <h2> The trouble with Bob </h2>  
    <h3> Alice </h3>  
    ...  
</div>
```

```
<div xmlns:dc="http://purl.org/dc/elements/1.1/">  
    <h2 property="dc:title"> The trouble with Bob </h2>  
    <h3 property="dc:creator"> Alice </h3>  
    ...  
</div>
```

Note the reference to the DC namespace, i.e. the Dublin Core initiative
<http://dublincore.org/>

SPARQL

- SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as [RDF](#) or viewed as [RDF](#) via middleware.
- SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions.
- Supports extensible value testing and constraining queries by source [RDF](#) graph.
- The results of SPARQL queries can be results sets or [RDF](#) graphs.

Source: SPARQL W3C Working group

<http://www.w3.org/2001/sw/wiki/SPARQL>

<http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>

SPARQL

Data:

```
<http://example.org/book/book1>  
  <http://purl.org/dc/elements/1.1/title>  
  "SPARQL Tutorial" .
```

Query:

```
SELECT ?title  
  WHERE { <http://example.org/book/book1>  
          <http://purl.org/dc/elements/1.1/title>  
          ?title .  
        }
```

Source: SPARQL W3C Working group

<http://www.w3.org/2001/sw/wiki/SPARQL>

<http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>

Ontology Web Language (OWL 1.0)

- Standard W3C
- Based upon/extend RDF/RDFS
- Formal Semantics (*Description Logic Fragments*)
- Three level of expressivity/complexity
 - OWL Lite
 - OWL DL
 - OWL Full
- OWL 2.0 is already a standard

OWL – Features

- **Classes (categories)**: subClassOf, intersectionOf, unionOf, complementOf, enumeration, equivalence, disjoint
- **Properties (Roles, Relations)**: symmetric, transitive, functional, inverse Functional, range, domain, subPropertyOf, inverseOf, equivalentProperty
- **Instances (Individuals)**: sameIndividualAs, differentFrom, allDifferent

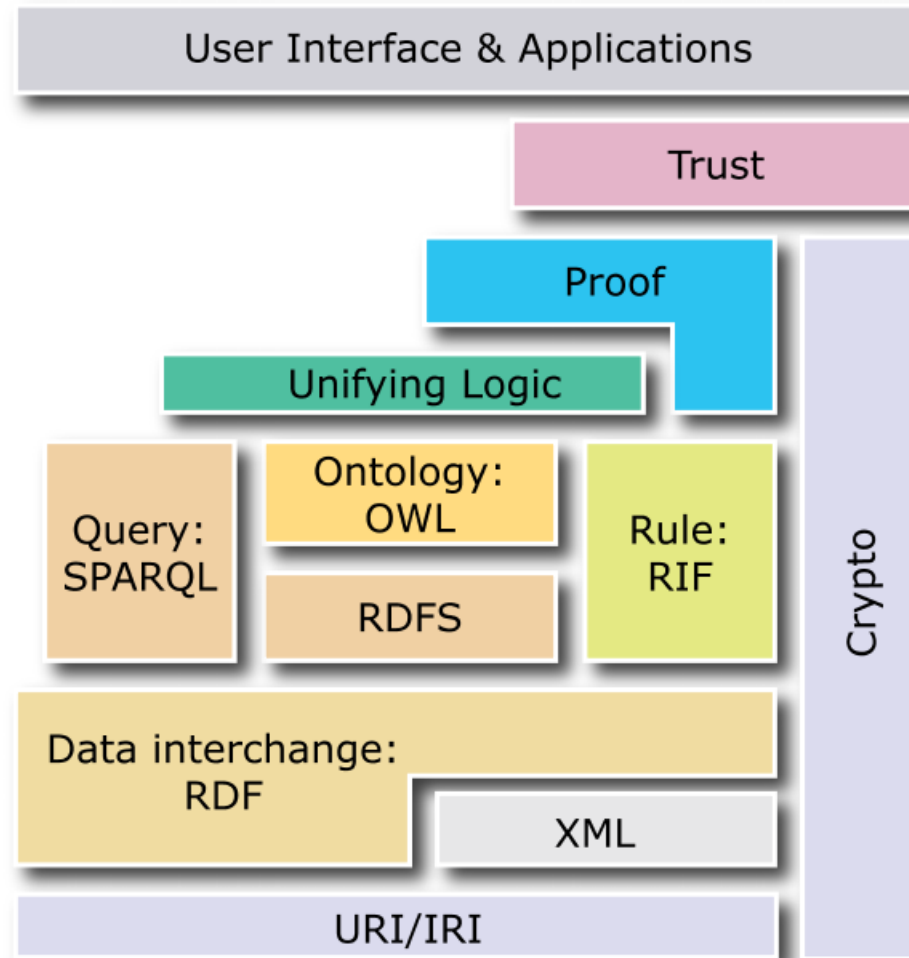
OWL Tools

- Many tools for OWL
 - Editors (37 listed at <http://www.w3.org/2001/sw/wiki/Category:Editor>)
 - Reasoners (39 listed at <http://www.w3.org/2001/sw/wiki/Category:Reasoner>)
- Quite often integrated in a comprehensive framework

A well known (but not necessarily the best one) ontology editor:

Protégé <http://protege.stanford.edu/>

The Semantic Web Cake



Knowledge Graphs

Knowledge Graphs

Suggested Readings:

1. Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. **Industry-scale knowledge graphs: lessons and challenges**. Commun. ACM 62, 8 (July 2019), 36–43. DOI: <https://doi.org/10.1145/3331166>
2. Dieter Fensel, Umutcan Simsek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, Alexander Wahler: **Knowledge Graphs - Methodology, Tools and Selected Use Cases**. Springer 2020, ISBN 978-3-030-37438-9, pp. 1-147



Knowledge Graphs – technological push

- 2012: At Google they understand the need of overcoming search approaches based on statistical data retrieval only...
- ... they need something able to represent and reason upon knowledge...
- The Semantic Web stack seems too complex, and not so fast for the Google needs
 - Reasoning on the T-Box is computationally expensive, and might even not terminate
- They mentioned the term "Knowledge Graphs" in 2012



Knowledge Graphs – business push

- Early 2000 have shown the win of statistical methods over KB methods in accessing the web

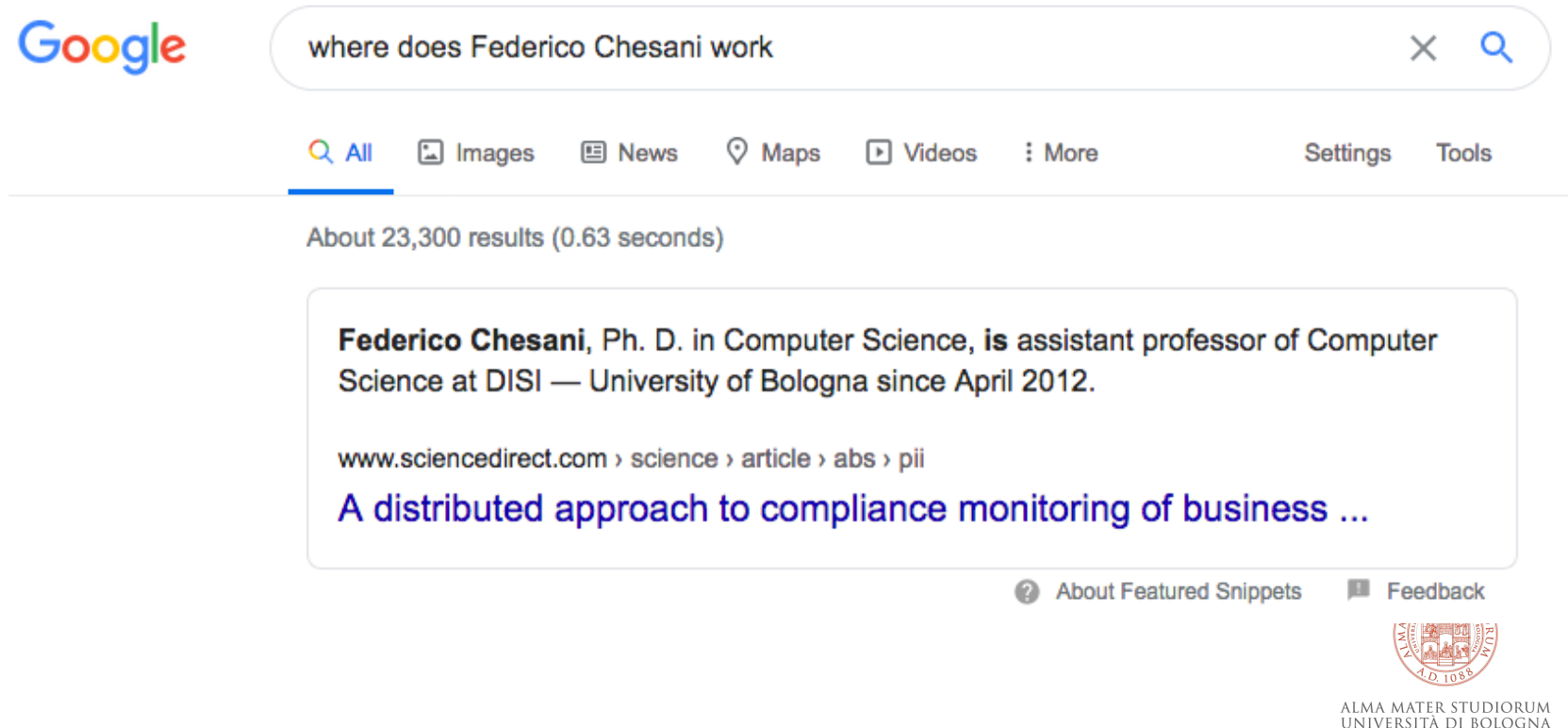
The problem was that people were clicking urls and go away from google, so none as watching the time. So to increase the time of people stay on the site. Google starts to give responses

- Google business model:
 - People make a search
 - Google replies with urls and **ads**
 - People click on urls, and leave
- Limit of this model: users arrive and leave quite early



Knowledge Graphs – business push

- Transformation of Google, since 2011: from a **search engine**, to a **query-answering engine**
- Provide the users with the answer, and keep them in google pages....



Knowledge Graphs – the Google approach

How Google answer the two previous questions?

- Create a common, simple vocabulary...
schema.org
- Create a **simple**, but **robust** corpus of types, properties, etc.
- Push the web to adopts these standards (well, after all it is Google!)

Based on annotations, Google KG is reported to contains 100B facts about 1B entities



Knowledge Graphs

Basic idea: just store the information in terms of nodes and arcs connecting the nodes.

Logical formulas are missing...

...T-Box and A-Box are stored at the same "level".

- Billions of factual knowledge, in form of "triplettes"
- No conceptual schema: data from various sources, with different semantics can be mixed freely (avoids the *knowledge acquisition bottleneck*)
- **No reasoning** Google put inside the database both Francesco is the son of Federico and Fedeco is the father of Francesco. Beacuse putting both in the DB we do not need to reason so the reaserach will go faster
- Graph algorithms used to fast traverse the graph, looking for the solution
- "Embedding" allows to represent such graphs in terms of n-grams, and then apply ML techniques



Knowledge Graphs

Definition of Fensel and colleagues:

"Knowledge Graphs are very large semantic nets that integrate various and heterogeneous information sources to represent knowledge about certain domains of discourse."



Knowledge Graphs – "open" examples

- **DBpedia** (<http://dbpedia.org/>), defined as the "de facto central dataset on the Semantic Web"
 - October 2016 release counts 13B RDF triples
- **Freebase** (<http://www.freebase.com/>) started as a collaborative KB, then it was acquired by Google, and dismissed in 2016. Its knowledge was used to improve/extend Google's KG
- **YAGO** (<https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/>), also based on Wikipedia
- **NELL** (<http://rtw.ml.cmu.edu/rtw/resources>)
- **Wikidata** (https://www.wikidata.org/wiki/Wikidata:Main_Page), at August 2019 it counts 7B triples
- **KBpedia** (<http://www.kbpedia.org/>), offers a bridge towards any other KB (open and commercial)
- **Datacommons.org**, launched by Google in 2018, and integrates knowledge about geographic and administrative areas, demographics, and other public available. Accessible through a browser interface. Each fact brings along the *provenance*.



Knowledge Graphs – commercial examples

- **Cyc** (<http://www.cyc.com/>) "*one of the longest-living AI projects*", provides a common sense knowledge base.
- **Facebook's Entities Graphs**
(<http://www.facebook.com/notes/facebook-engineering/under-the-hood-the-entities-graph/10151490531588920>), used internally by Facebook, stores data about users, interests, and connections. Available through Facebook Graph API
- **Google's Knowledge Graph**
(<https://developers.google.com/knowledge-graph/>), in 2016 Google claimed it holds 70B facts. It is not known how information is stored, but it uses schema.org for terms. Accessible through the Google knowledge Graph API.



Examples of (fairly large!!!) Industry-scale Knowledge Graphs (from [1])

Common characteristics of the knowledge graphs.

	Data model	Size of the graph	Development stage
Microsoft	The types of entities, relations, and attributes in the graph are defined in an ontology.	~2 billion primary entities, ~55 billion facts	Actively used in products
Google	Strongly typed entities, relations with domain and range inference	1 billion entities, 70 billion assertions	Actively used in products
Facebook	All of the attributes and relations are structured and strongly typed, and optionally indexed to enable efficient retrieval, search, and traversal.	~50 million primary entities, ~500 million assertions	Actively used in products
eBay	Entities and relation, well-structured and strongly typed	Expect around 100 million products, >1 billion triples	Early stages of development and deployment
IBM	Entities and relations with evidence information associated with them.	Various sizes. Proven on scales documents >100 million, relationships >5 billion, entities >100 million	Actively used in products and by clients

Google's Knowledge Graph – an example

Google

trattoria boni

All Maps Images News Videos More Settings Tools

About 153,000 results (0.65 seconds)


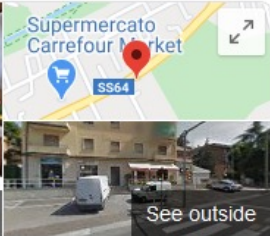
www.trattoriaboni.it ▾ [Translate this page](#)
Trattoria Boni Bologna: Home
Una delle più vecchie **trattorie** bolognesi dove poter riscoprire sapori e profumi del bel tempo che fu. Un tempo nel quale la "nouvelle cuisine" non era ancora ...

www.trattoriaboni.it/index.php > [menu](#) ▾ [Translate this page](#)
Menu - Trattoria Boni Bologna
Una delle più vecchie **trattorie** bolognesi dove poter riscoprire sapori e profumi del bel ...
Piatto Rustico di **Boni** (tigelle e crescentine con affettati) SOLO A CENA.

www.tripadvisor.com > ... > Bologna > Bologna Restaurants ▾
Trattoria Boni, Bologna - Menu, Prices & Restaurant Reviews ...
★★★★★ Rating: 4 - 857 reviews - Price range: \$\$ - \$\$\$
Trattoria Boni, Bologna: See 857 unbiased reviews of **Trattoria Boni**, rated 4 of 5 on Tripadvisor and ranked #210 of 1830 restaurants in Bologna.
You visited this page on 2/18/20.

ristadvisor.it > Trattoria-Boni_Bologna ▾ [Translate this page](#)
Trattoria Boni Bologna
Trattoria Boni - Bologna cucina tipica bolognese. in questo locale si va sul sicuro si mangia alla grande con piatti tipici bolognesi tortellini tagliatelle bollito e ...

it-it.facebook.com > Luoghi > Bologna > Ristorante italiano
Trattoria Boni - Bologna - Menù, prezzi, recensioni dei ...

 [See photos](#)  [See outside](#)

Trattoria Boni

[Sito web](#) [Indicazioni stradali](#) [Salva](#)

4.3 ★★★★★ 1,129 recensioni Google
€€ · Ristorante di cucina tradizionale

Indirizzo: Via Don Luigi Sturzo, 22, 40135 Bologna BO

Orari: **Closed** · Opens 12PM ▾
Orari suggeriti da un utente
Mon Closed
Tue-Sat 12–2:45PM and 7:30–10:30PM
Sun 12–2:45PM

Telefono: 051 615 4337

[Suggerisci una modifica](#) · Sei il proprietario di quest'attività?

Where does this information come from?



Some key questions when evaluating the *quality* of a Knowledge Graph

- **Coverage**

Does the graph have all the required information? If not, can we estimate how much of it?

Usually the answer to the former question is NO...

- **Correctness**

Is the information correct?

Correctness... it depends on the type of information (it can be objective or subjective), and it depends on the *intended use* of the information

- **Freshness**

Is the content up-to-date?

Keep in mind the change rate of the domain: it is different to represent stock quotes, and parenthood



Google KG API

An example from <https://developers.google.com/knowledge-graph> :

```
"""Example of Python client calling Knowledge Graph Search API."""
from __future__ import print_function
import json
import urllib

api_key = open('.api_key').read()
query = 'Taylor Swift'
service_url = 'https://kgsearch.googleapis.com/v1/entities:search'
params = {
    'query': query,
    'limit': 10,
    'indent': True,
    'key': api_key,
}
url = service_url + '?' + urllib.urlencode(params)
response = json.loads(urllib.urlopen(url).read())
for element in response['itemListElement']:
    print(element['result']['name'] + ' (' + str(element['resultScore']) +
    ')')
```



Google KG API

An example from <https://developers.google.com/knowledge-graph> :

```
{
  "@context": {
    "@vocab": "http://schema.org/",
    "goog": "http://schema.googleapis.com/",
    "resultScore": "goog:resultScore",
    "detailedDescription": "goog:detailedDescription",
    "EntitySearchResult": "goog:EntitySearchResult",
    "kg": "http://g.co/kg"
  },
  "@type": "ItemList",
  "itemListElement": [
    {
      "@type": "EntitySearchResult",
      "result": {
        "@id": "kg:/m/0dl567",
        "name": "Taylor Swift",
        "@type": [
          "Thing",
          "Person"
        ],
        "description": "Singer-songwriter",
        "image": {
          "contentUrl": "https://t1.gstatic.com/images?q=tbn:ANd9GcQmVDAhjhWnN2OWys2ZMO3PGAhupp5tN2LwF_BJmiHgi19hf8Ku",
          "url": "https://en.wikipedia.org/wiki/Taylor_Swift",
          "license": "http://creativecommons.org/licenses/by-sa/2.0"
        },
        "detailedDescription": {
          "articleBody": "Taylor Alison Swift is an American singer-songwriter and actress. Raised in Wyomissing, Pennsylvania, she moved to Nashville, Tennessee, at the age of 14 to pursue a career in country music. ",
          "url": "http://en.wikipedia.org/wiki/Taylor_Swift",
          "license": "https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License"
        },
        "url": "http://taylorswift.com/"
      },
      "resultScore": 4850
    }
  ]
}
```



A new contact point between KR and ML?

KGs have raised a lot of interest also from the ML side.

- Given the huge number of facts available, isn't it possible to learn something?

Few simple tasks:

- **Entity prediction:** given a source/target entity and a link, which is the most “probable” target/source entity?
- **Link prediction:** given two entities (interpreted as a source and a target of a triplete), which is/are the most probable links connecting them?



Graph embedding

- Triplettes come in the shape of (h, r, t)
 - h (head) and t (tail) are entities
 - r is a relation
- a KG is a set $S=\{(h, r, t)\}$

IDEA: let us represent entities and relations into a vectorial space.

- Would it be possible then to learn the relation function?
 - Input: (h, t) and r
 - Output: a ANN able to predict r , given some h and t .
- Would it be possible to predict some entity?
 - Input: (h, r) and t
 - Output: a ANN able to predict t , given some h and r .



Graph embedding

Usually, three steps:

1. Choose a representation space for entities and relations
2. Choose a scoring function
3. Learn a representation function

Many works have obtained important results:

- TransE
- TransH
- TransR
- TransD

- KG2E
- TransG

Further reading:

Wang, Q., Mao, Z., Wang, B. & Guo, L. (2017). Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Transactions on Knowledge and Data Engineering , 29, 2724--2743. doi:10.1109/TKDE.2017.2754499

Python Library: **OpenKE** <https://github.com/thunlp/OpenKE>

