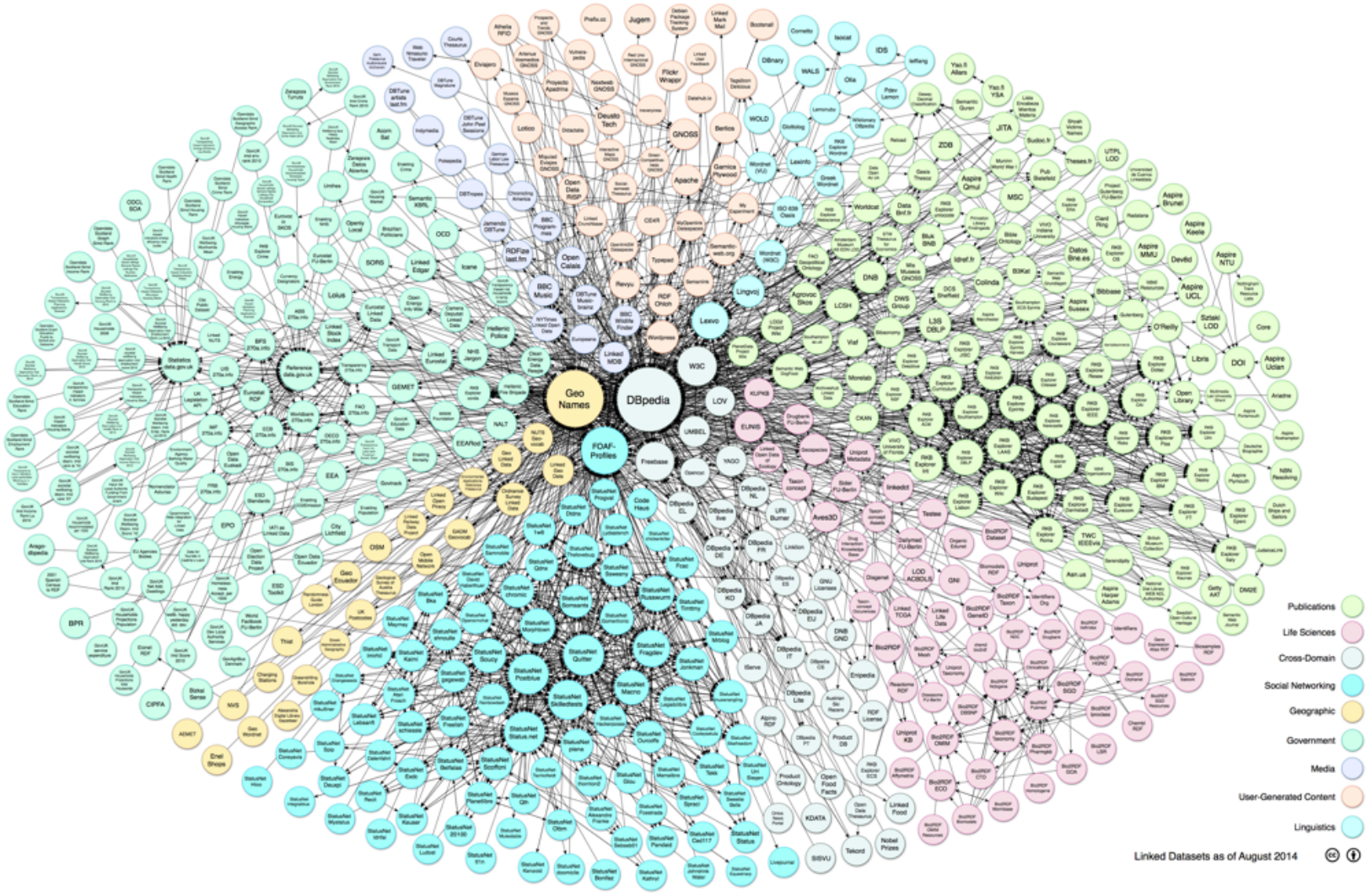
The Semantic Web:
A Lightweight Data Integration Platform

Marcelo Arenas

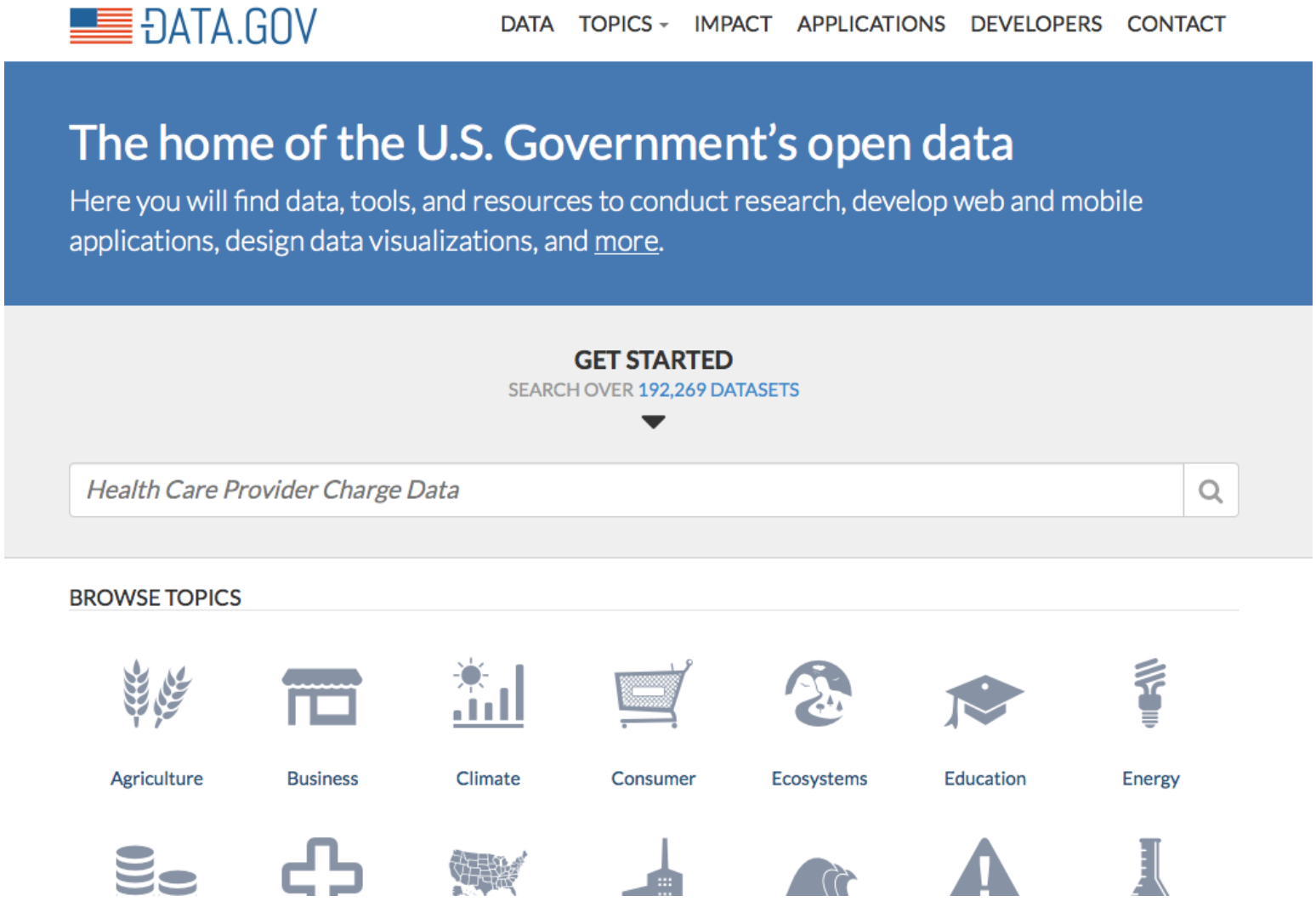
PUC Chile & Center for Semantic Web Research

From the Web of documents to the Web of data

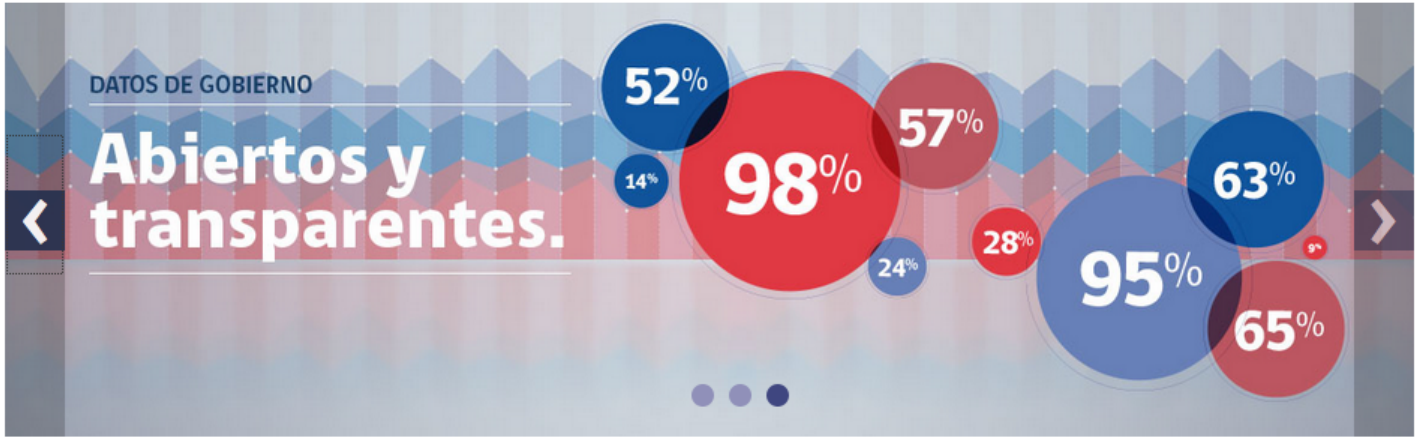
From the Web of documents to the Web of data



Different scenarios: transparency in government data



Different scenarios: transparency in government data



Different scenarios: transparency and open data

Different scenarios: open data

Different scenarios: open data

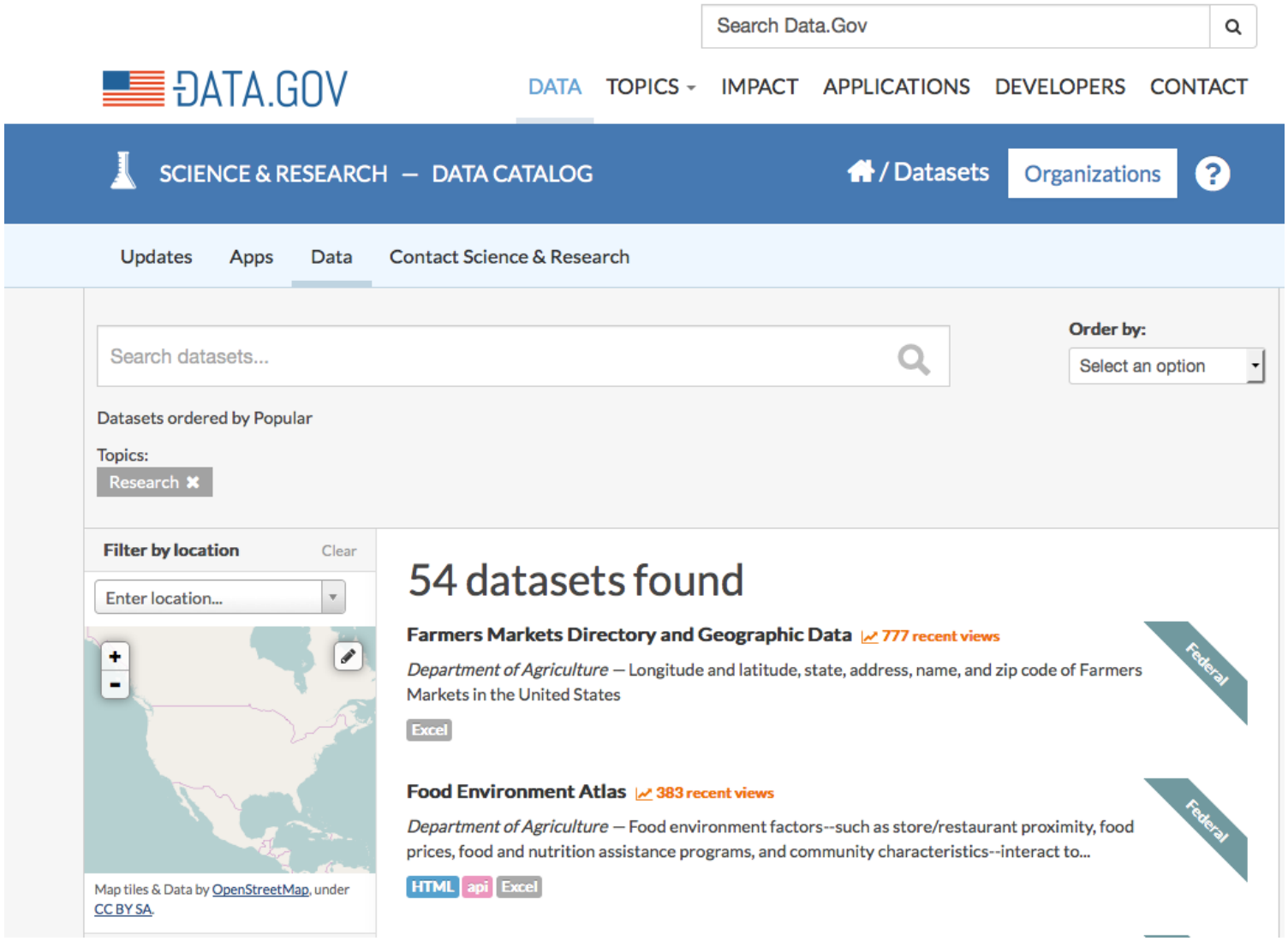
COLUMBIA UNIVERSITY LIBRARIES

Latest update: 2016-08-19

Agency	Article Solution (A)	Maximum Embargo Period	Data Solution (D)
ACL/ NIDILRR	PubMed Central (PMC)	12 months	TBD
AHRQ	PMC	12 months following publication date	Commercial repository, yet to be named* [DMP guidance]
ASPR**	PMC	12 months	Scientific data repositories, data.gov data registry* [DMP guidance]
CDC**	CDC Stacks, using NIHMS submission system	12 months	Multiple solutions + data registry [DMP guidance]
DOD	Defense Technical Information Center (DTIC)	12 months	No specific solution* [DMP guidance]
DOE	Public Access Gateway for Energy and Science (PAGES)	12 months	Varies by office* [DMP template]
DOT	DOT National Transportation Library (NTL)	N/A	To be released [DMP guidance]
FDA**	PMC	12 months	Disciplinary data repositories, where available* [DMP guidance]

Different scenarios: scientific open data

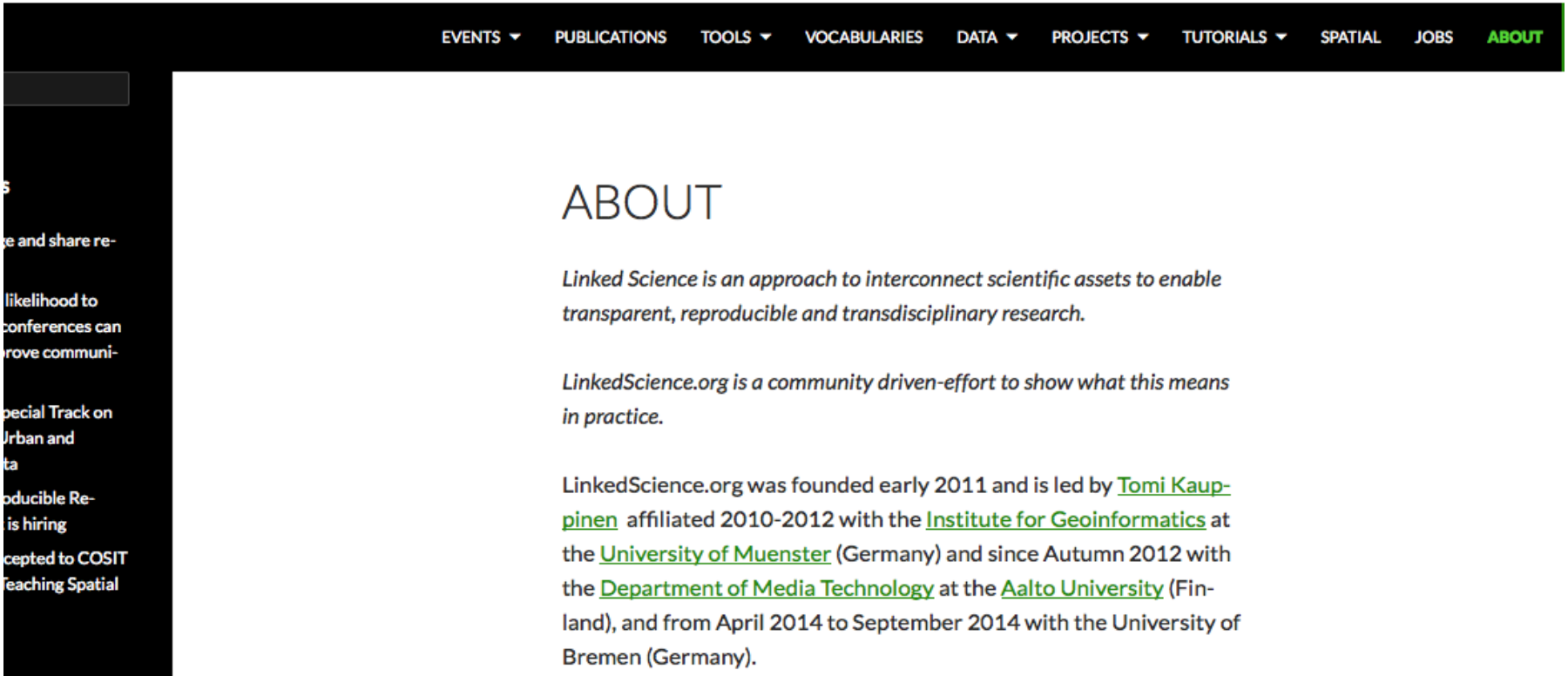
Different scenarios: scientific open data



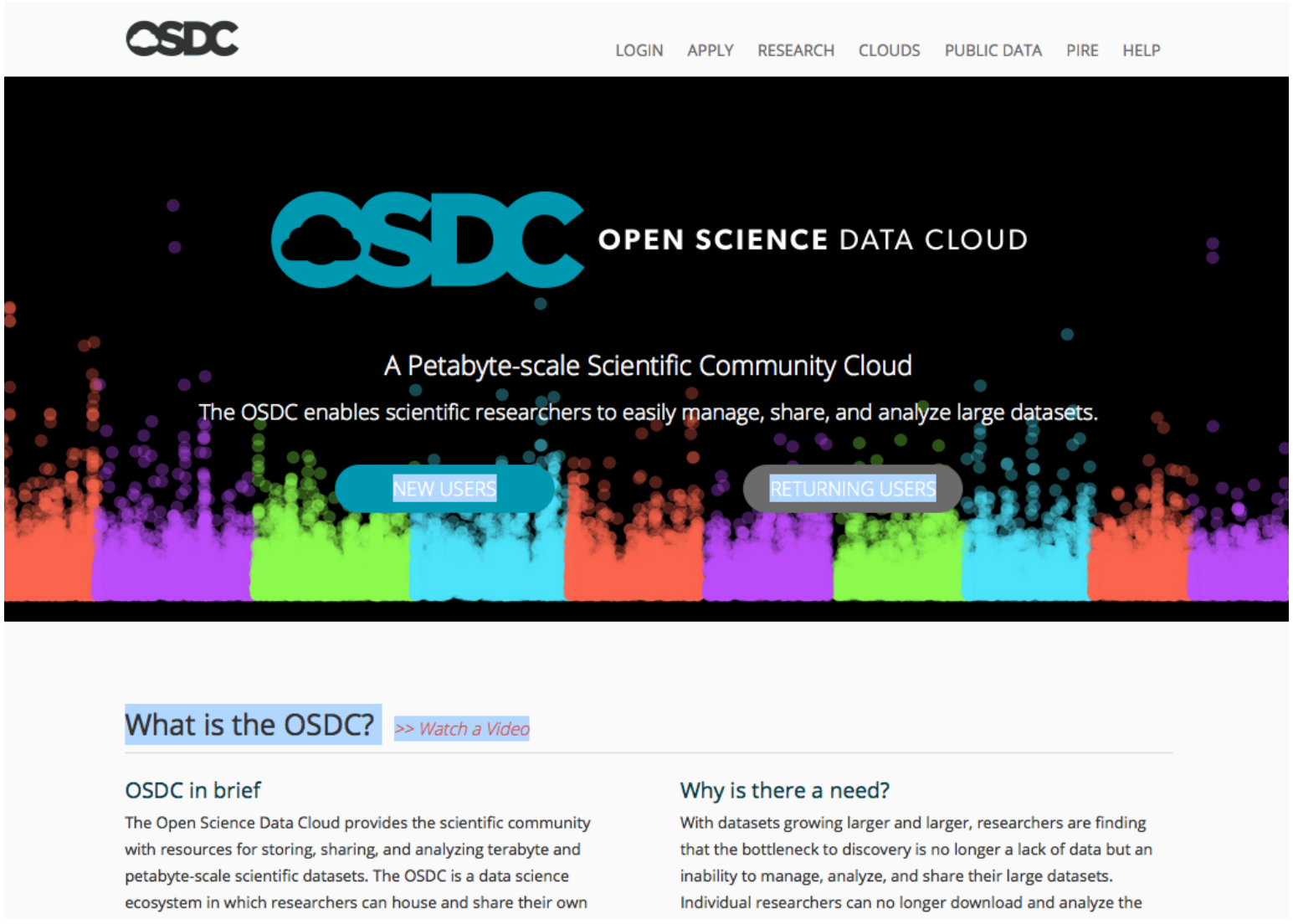
Different scenarios: scientific open data

Different scenarios: scientific open data

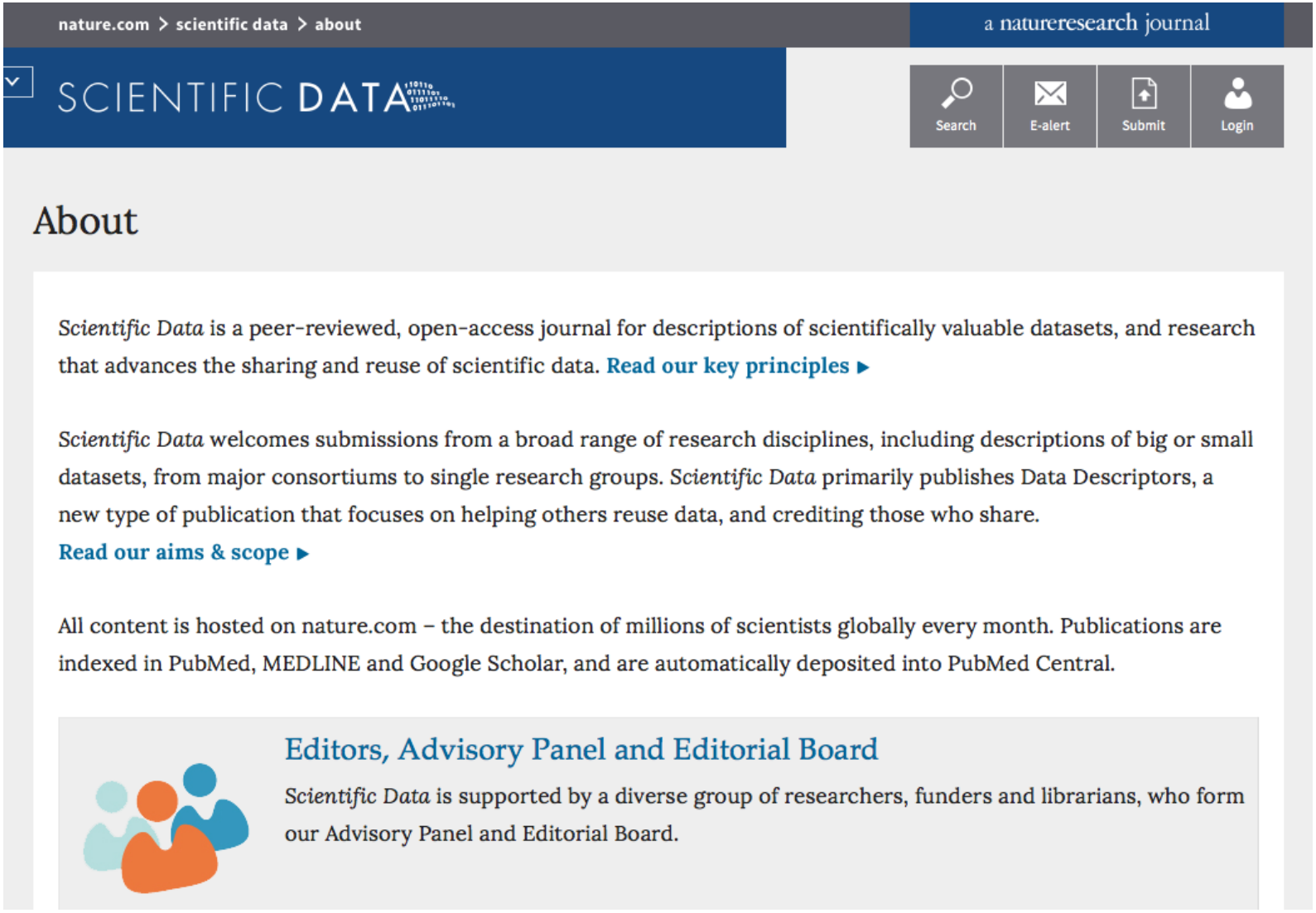
Different scenarios: scientific open data



Different scenarios: scientific open data



Different scenarios: scientific open data



What are the common challenges?

What are the common challenges?

- Storing

What are the common challenges?

- Storing
- Publishing

What are the common challenges?

- Storing
- Publishing
  - Anonymizing

What are the common challenges?

- Storing
- Publishing
  - Anonymizing
- Linking

What are the common challenges?

- Storing
- Publishing
  - Anonymizing
- Linking
  - How can we relate pieces of data in different repositories? How can we make these relationships human and machine understandable? How can we identify the same element in different repositories?

What are the common challenges?

- Storing
- Publishing
  - Anonymizing
- Linking
  - How can we relate pieces of data in different repositories? How can we make these relationships human and machine understandable? How can we identify the same element in different repositories?
- Integrating

What are the common challenges?

- Storing
- Publishing
  - Anonymizing
- Linking
  - How can we relate pieces of data in different repositories? How can we make these relationships human and machine understandable? How can we identify the same element in different repositories?
- Integrating
  - Can we provide a common view of the data? How can we translate data from one repository to another?

What are the common challenges?

- Querying

What are the common challenges?

- Querying
- Ranking

What are the common challenges?

- Querying
- Ranking
- Updating

What are the common challenges?

- Querying
- Ranking
- Updating
- Visualizing

What are the common challenges?

- Querying
- Ranking
- Updating
- Visualizing
- Dealing with uncertainty

What are the common challenges?

- Querying
- Ranking
- Updating
- Visualizing
- Dealing with uncertainty

Is there a common framework where these problems can be studied and (hopefully) solved?

What are the common challenges?

- Querying
- Ranking
- Updating
- Visualizing
- Dealing with uncertainty

Is there a common framework where these problems can be studied and (hopefully) solved?

- The Semantic Web is a common framework to tackle these issues

The Semantic Web

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

[Tim Berners-Lee et al. 2001.]

The Semantic Web

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

[Tim Berners-Lee et al. 2001.]

Specific goals:

The Semantic Web

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

[Tim Berners-Lee et al. 2001.]

Specific goals:

- Build a resource description language with standard semantics

The Semantic Web

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

[Tim Berners-Lee et al. 2001.]

Specific goals:

- Build a resource description language with standard semantics
  - Make semantics machine-processable and understandable

The Semantic Web

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

[Tim Berners-Lee et al. 2001.]

Specific goals:

- Build a resource description language with standard semantics
  - Make semantics machine-processable and understandable
- Incorporate logical infrastructure to reason about resources

The Semantic Web

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

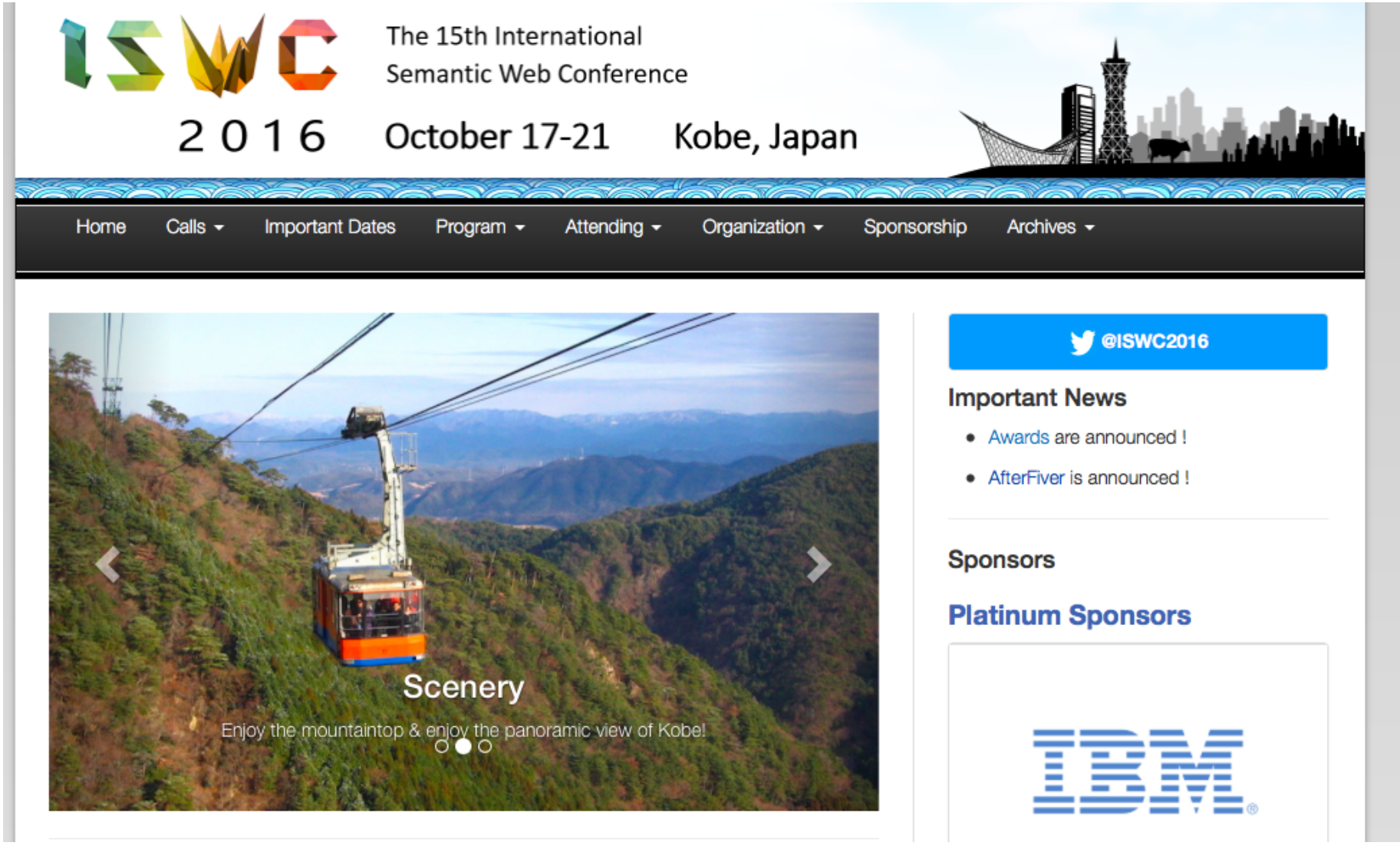
[Tim Berners-Lee et al. 2001.]

Specific goals:

- Build a resource description language with standard semantics
  - Make semantics machine-processable and understandable
- Incorporate logical infrastructure to reason about resources

W3C proposals: Resource Description Framework (RDF) and SPARQL

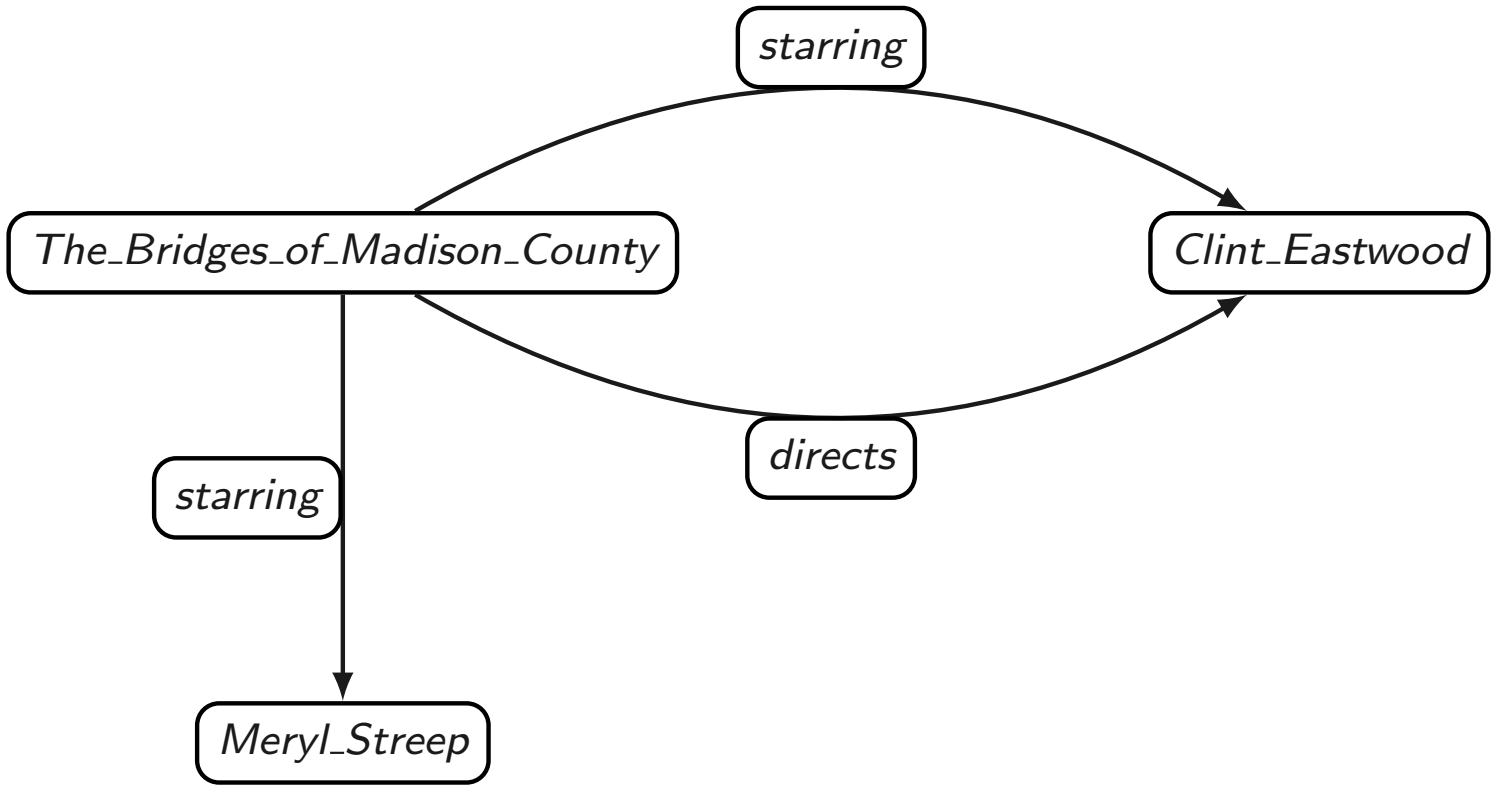
The Semantic Web



RDF in a nutshell

- RDF is the W3C proposal framework for representing information in the Web
- Abstract syntax based on directed labeled graph
- Extensible URI-based vocabulary

An RDF graph


starring
The_Bridges_of_Madison_County
Clint_Eastwood
directs
starring
Meryl_Streep

An RDF graph in real life: DBpedia

```
http://dbpedia.org/resource/The_Bridges_of_Madison_County_(film)
http://dbpedia.org/property/director
http://dbpedia.org/resource/Clint_Eastwood .

http://dbpedia.org/resource/The_Bridges_of_Madison_County_(film)
http://dbpedia.org/property/starring
http://dbpedia.org/resource/Clint_Eastwood .

http://dbpedia.org/resource/The_Bridges_of_Madison_County_(film)
http://dbpedia.org/property/starring
http://dbpedia.org/resource/Meryl_Streep .
```

Prefixes simplify the notation

Prefixes can be defined in an RDF graph to simplify notation

- They are defined also using triples

Prefixes simplify the notation

Prefixes can be defined in an RDF graph to simplify notation

- They are defined also using triples

We can include in an RDF graph the following triples:

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

Prefixes simplify the notation

Prefixes can be defined in an RDF graph to simplify notation

- They are defined also using triples

We can include in an RDF graph the following triples:

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

Then `<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>` can be replaced by `rdf:type`

How are URIs created and assigned?

How are URIs created and assigned?

There is no centralized mechanism

How are URIs created and assigned?

There is no centralized mechanism

A key component to deal with this issue: `owl:sameAs`

```
http://dbpedia.org/resource/Meryl_Streep   owl:sameAs
http://cs.dbpedia.org/resource/Meryl_Streepová .

http://dbpedia.org/resource/Meryl_Streep   owl:sameAs
http://yago-knowledge.org/resource/Meryl_Streep .

http://dbpedia.org/resource/Meryl_Streep   owl:sameAs
http://data.nytimes.com/32250484050106278413 .
```

How are URIs created and assigned?

There is no centralized mechanism

A key component to deal with this issue: `owl:sameAs`

```
http://dbpedia.org/resource/Meryl_Streep   owl:sameAs
http://cs.dbpedia.org/resource/Meryl_Streepová .

http://dbpedia.org/resource/Meryl_Streep   owl:sameAs
http://yago-knowledge.org/resource/Meryl_Streep .

http://dbpedia.org/resource/Meryl_Streep   owl:sameAs
http://data.nytimes.com/32250484050106278413 .
```

Reasoning capabilities are needed to deal with `owl:sameAs`

Querying RDF: SPARQL

- SPARQL is the W3C recommendation query language for RDF (January 2008)
- Originally it was a graph-matching query language
- SPARQL 1.1 is the new version of this language, its was released in March 2013

An example of a SPARQL query

Retrieve all the movies in DBpedia

An example of a SPARQL query

Retrieve all the movies in DBpedia

```
?movie  rdf:type  <http://schema.org/Movie> .
```

An example of a SPARQL query

Retrieve all the movies in DBpedia

```
WHERE
{
   ?movie  rdf:type  <http://schema.org/Movie> .
}
```

An example of a SPARQL query

Retrieve all the movies in DBpedia

```
SELECT ?movie
WHERE
{
   ?movie  rdf:type  <http://schema.org/Movie> .
}
```

Important problems when querying RDF

Important problems when querying RDF

- Returning as much information as possible

Important problems when querying RDF

- Returning as much information as possible
- Reasoning with ontologies

Important problems when querying RDF

- Returning as much information as possible
- Reasoning with ontologies
- Dealing with incomplete information

Important problems when querying RDF

- Returning as much information as possible
- Reasoning with ontologies
- Dealing with incomplete information
- Exploiting the graph structure of RDF

Important problems when querying RDF

- Returning as much information as possible
- Reasoning with ontologies
- Dealing with incomplete information
- Exploiting the graph structure of RDF
- Working with highly distributed data

The Center for Semantic Web Reseacrh

An initiative of four universities: PUC Chile, Universidad de Chile, Universidad de Talca y Universidad Técnica Federico Santa María

- Funded by the Iniciativa Científica Milenio

The Center for Semantic Web Reseacrh

An initiative of four universities: PUC Chile, Universidad de Chile, Universidad de Talca y Universidad Técnica Federico Santa María

- Funded by the Iniciativa Científica Milenio

Members:

- 13 full-time professors
- A large number of Ph.D, Master's and undergrad students

The Center for Semantic Web Reseacrh

We do research on all the areas mentioned before.

- With an emphasis on their scientific aspects

The Center for Semantic Web Reseacrh

We do research on all the areas mentioned before.

- With an emphasis on their scientific aspects

But we are also interested in developing prototypes incorporating our results

The Center for Semantic Web Reseacrh

We do research on all the areas mentioned before.

- With an emphasis on their scientific aspects

But we are also interested in developing prototypes incorporating our results

- Let me show you two of our applied projects ...

Detecting and measuring the impact of earthquakes using social sensors

Why do we use Social Media?

- Communicative nature
- Twitter is largely used in mobile devices
- Users use Twitter like a news source

What is a social sensor?

- Each Twitter user is a social sensor
- Social sensor detects an event and publish a post (without a known pattern)
- Very noisy type of sensor

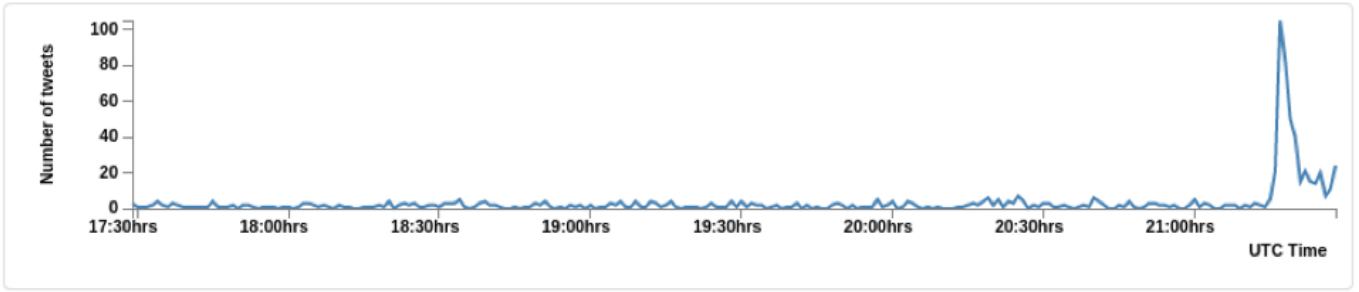
Chile as a study case

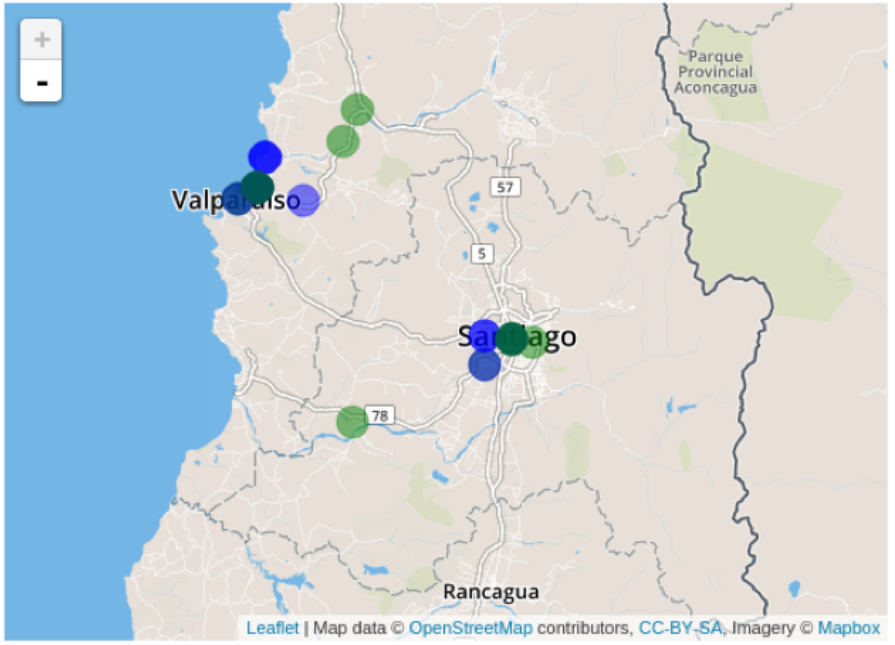
- One of the most seismic countries in the world
- One of the top-ten highest Twitter users per capita
- Seismological Center of University of Chile is very interested

What do we want to do?

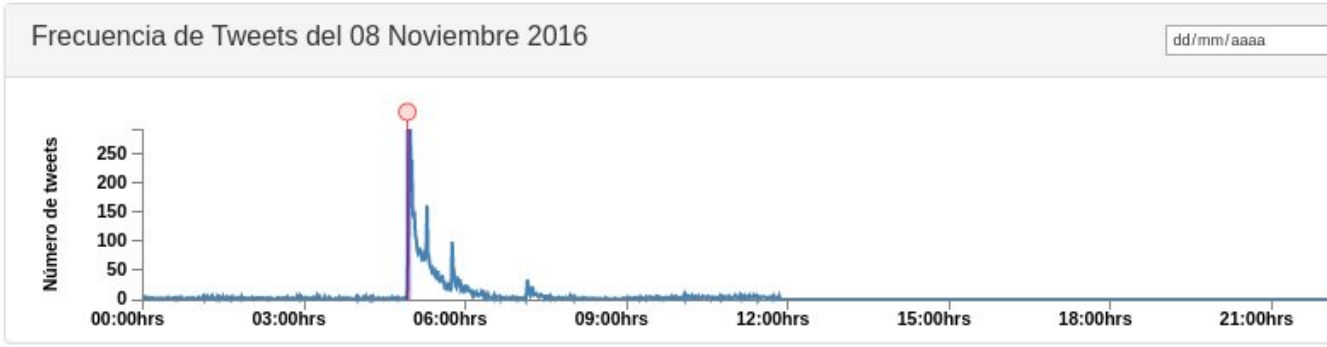
- Offer to Seismological Center of University of Chile a tool to explore earthquake related data
- Complement earthquakes information using tweets data such as location names, opinions about the event and sentiments

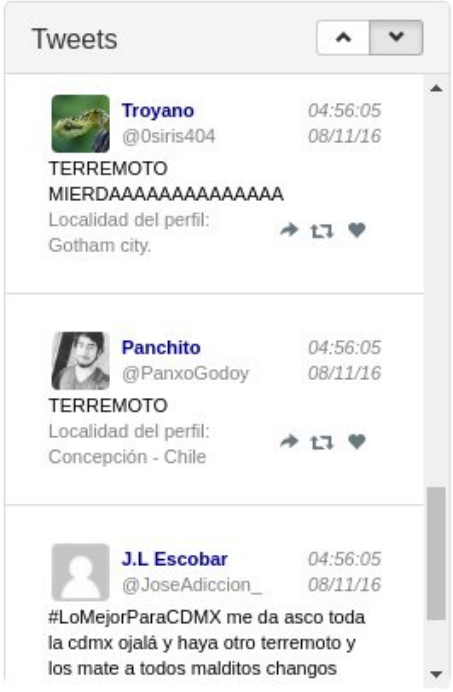
Visualization tool

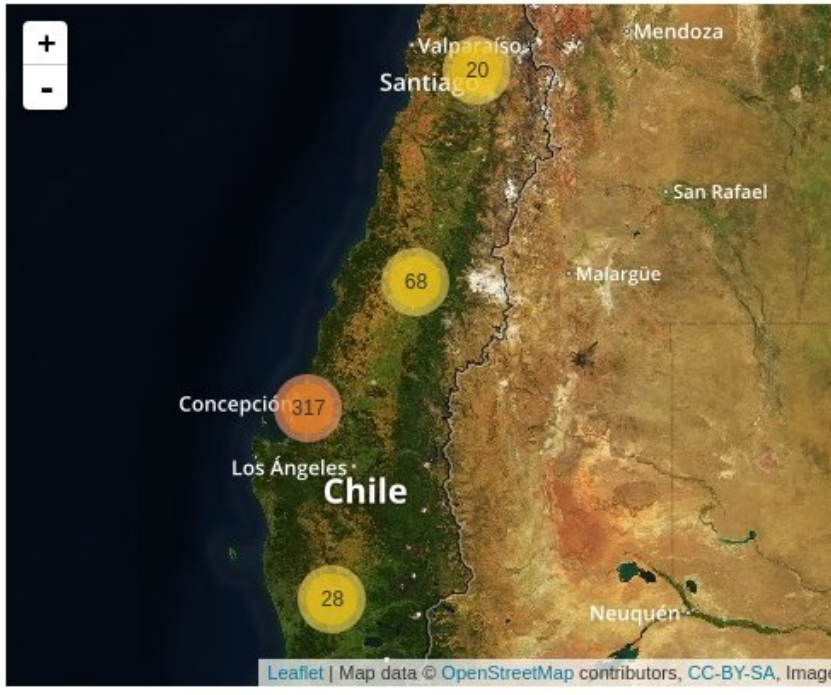




Visualization tool







Results

- Currently the system detects earthquakes over 4 in Richter scale
- It throws alerts in less than 2.5 minutes in the worst case
- The platform allows to observe Twitter information only a few seconds after an earthquake strikes
- Geographical visualization has high granularity level although a lot of data are not obtained by GPS

Open data and the Chilean constitutional process

Open data and the Chilean constitutional process

A basis for a new constitution.

- Effort to create a new Constitution via several small assemblies countrywide

But the data gathered from the process was not really open

We created a Web repository to:

- analyze and visualize the (small) data available
- raise awareness of the importance of opening the data of this process

Web site: `http://constitucionabierta.cl`

http://constitucionabierta.cl

Our efforts helped to make the government to open the data of the process

Today there is public access to (anonymized) data of

- Political opinions of 8.000 groups (of 10-30 people)
- Individual opinions of 80.0000 participants
- 2 million "constitutional concepts" prioritized

We have processed this data and cooperated with several public and social organizations

Thank you!

Working with highly distributed data

Web data is highly distributed

Data can be stored in different repositories

- Different pieces of data have to be collected to answer a query

Web data is highly distributed

Data can be stored in different repositories

- Different pieces of data have to be collected to answer a query

An important notion to deal with this issue: SPARQL endpoint

- A Web service that accepts a SPARQL query as input, and returns (part of) the result to the query

Web data is highly distributed

Data can be stored in different repositories

- Different pieces of data have to be collected to answer a query

An important notion to deal with this issue: SPARQL endpoint

- A Web service that accepts a SPARQL query as input, and returns (part of) the result to the query

SPARQL has an operator `SERVICE` to query an endpoint

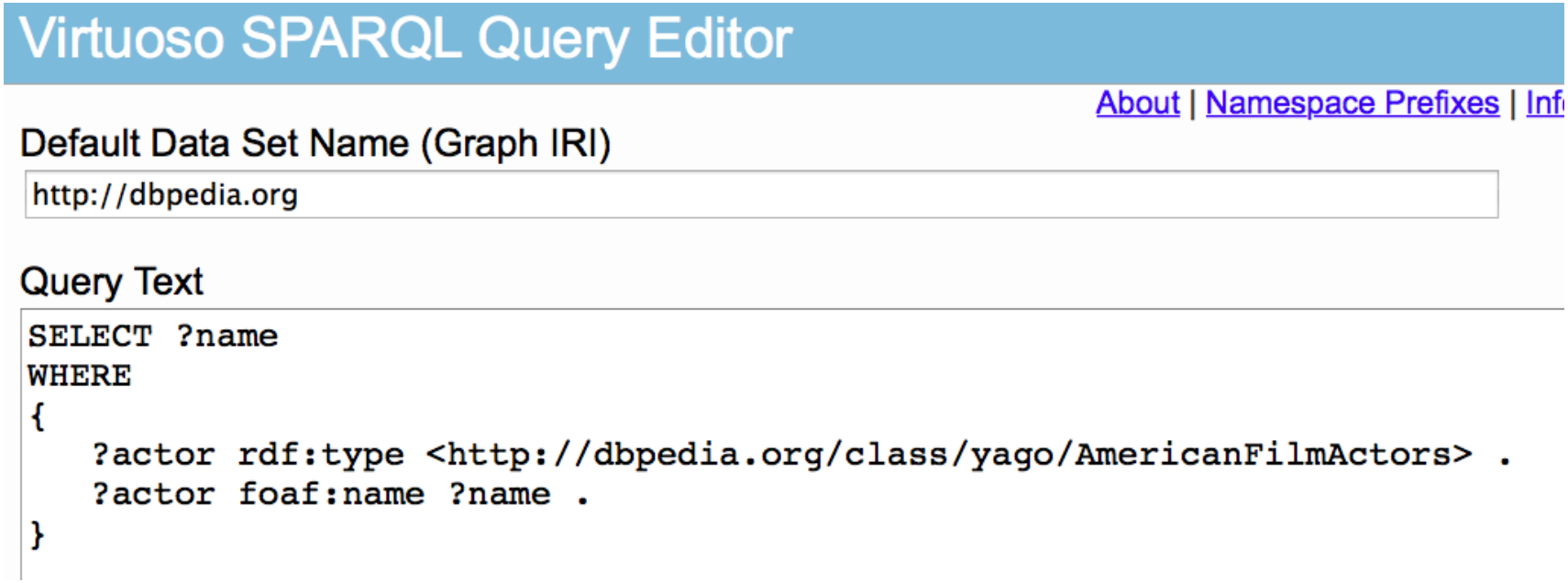
The SPARQL endpoint of DBpedia

Querying DBpedia

We want to retrieve the list of American actors in DBpedia

Querying DBpedia

We want to retrieve the list of American actors in DBpedia



The answer to the query

name
"Courtenay Taylor"@en
"Taylor, Courtenay"@en
"Nakia Burrise"@en
"Burrise, Nakia"@en
"Alan Hale, Sr."@en
"Hale, Alan, Sr."@en
"Alec Baldwin"@en
"Baldwin, Alec"@en

...

The SPARQL endpoint of DBLP

We want to retrieve the list of authors in DBLP

The SPARQL endpoint of DBLP

We want to retrieve the list of authors in DBLP

SPARQL:

```
PREFIX d2r: <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-server/config.rdf#>
PREFIX swrc: <http://swrc.ontoware.org/ontology#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX map: <file:///home/diederich/d2r-server-0.3.2/dblp-mapping.n3#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
SELECT ?name
WHERE
{
  ?paper  dc:creator  ?author .
  ?author  foaf:name  ?name .
}
```

Results: Browse Go! Reset

The answer to the query

SPARQL results:

name
"Sanjeev Saxena"
"Hans-Ulrich Simon"
"Nathan Goodman"
"Oded Shmueli"
"Norbert Blum"
"Arnold Schönhage"
"Juha Honkala"
"Chua-Huang Huang"
"Christian Lengauer"

. . .

We would like to combine the previous results . . .

```
SELECT ?name
WHERE
{
  ?actor rdf:type <http://dbpedia.org/class/yago/AmericanActors> .
  ?actor foaf:name ?name .
  SERVICE <http://dblp.l3s.de/d2r/sparql>
  {
     SELECT ?name
     WHERE
     {
        ?paper dc:creator ?author .
        ?author foaf:name ?name .
     }
  }
}
```

Open issues when dealing with distribution

Some important problems:

- The notion of SPARQL endpoint needs to be formalized
  - What queries are accepted?
  - How is the time distributed between them?
  - Should a pricing model be used?
  - What is the protocol to return the answer to a query?

- A more general notion of endpoint should be formalized and studied

Open issues when dealing with distribution (cont'd)

- Usability needs to be hugely improved
  - schema/structure extraction and visualization play a fundamental role here
- Approaches for discovering relevant data should be studied
- Operators to distribute the execution of queries should be studied in more depth

Other important problems when querying RDF

Returning as much information as possible

RDF follows an open world assumption

Users may be unaware of the structure of the data

Returning as much information as possible

RDF follows an open world assumption

Users may be unaware of the structure of the data

Thus, the possibility of obtaining additional information if possible is important in this scenario

- In fact, this feature was present from the very beginning in SPARQL

An optional operator

Retrieve each movie in DBpedia and its gross if this information is available

```
SELECT ?movie ?gross
WHERE
{
   ?movie  rdf:type  <http://schema.org/Movie> .
   OPTIONAL
   {
      ?movie  <http://dbpedia.org/property/gross>  ?gross .
   }
}
```

Part of the answer to the query

?movie	?gross
http://dbpedia.org/resource/Frozen_(2013_film)	"1.274E9"
http://dbpedia.org/resource/Amazon_Souls	

What is new?

The `OPTIONAL` operator essentially corresponds to a left-outer join in relational algebra

But ...

What is new?

The `OPTIONAL` operator essentially corresponds to a left-outer join in relational algebra

But ...

- The fragments of SPARQL that are natural to study are different than for the case of relational algebra
  - The complexity of evaluating these fragments was not known [Pérez, A. & Gutierrez 2009; Schmidt, Meier & Lausen 2010]
- New notions of safeness are needed to avoid a counterintuitive behavior [Pérez, A. & Gutierrez 2009]
- New optimization techniques are needed [Pérez, A. & Gutierrez 2009; Letelier, Pérez, Pichler & Skritek 2013; Pichler & Skritek 2014]

Reasoning with ontologies

Reasoning capabilities are needed

Reasoning with ontologies

Reasoning capabilities are needed

- We already mentioned owl:sameAs

Reasoning with ontologies

Reasoning capabilities are needed

- We already mentioned owl:sameAs
- An RDF graph can use RDF Schema (RDFS) to establish hierarchies of classes and properties

Reasoning with ontologies

Reasoning capabilities are needed

- We already mentioned owl:sameAs
- An RDF graph can use RDF Schema (RDFS) to establish hierarchies of classes and properties
- The Web Ontology Language (OWL) can be used to define more complex relations between classes and properties

The use of RDFS vocabulary

The following triples are included in DBpedia:

The use of RDFS vocabulary

The following triples are included in DBpedia:

```
http://dbpedia.org/class/yago/Professor110480730
rdfs:subClassOf
http://dbpedia.org/class/yago/Academician109759069 .

http://dbpedia.org/class/yago/Academician109759069
rdfs:subClassOf
http://dbpedia.org/class/yago/Educator110045713 .

http://dbpedia.org/ontology/championInDoubleFemale
rdfs:subPropertyOf
http://dbpedia.org/ontology/championInDouble .

http://dbpedia.org/ontology/championInDouble
rdfs:subPropertyOf
http://dbpedia.org/ontology/champion .
```

The use of RDFS vocabulary

Some numbers in DBpedia:

- triples with `rdfs:subClassOf` as predicate are at least 450K
- triples with `rdfs:subPropertyOf` as predicate are at least 1K

The use of RDFS vocabulary

Some numbers in DBpedia:

- triples with `rdfs:subClassOf` as predicate are at least 450K
- triples with `rdfs:subPropertyOf` as predicate are at least 1K

We need reasoning capabilities to deal with:

- `rdfs:subClassOf`, `rdfs:subPropertyOf`

The use of RDFS vocabulary

Some numbers in DBpedia:

- triples with `rdfs:subClassOf` as predicate are at least 450K
- triples with `rdfs:subPropertyOf` as predicate are at least 1K

We need reasoning capabilities to deal with:

- `rdfs:subClassOf`, `rdfs:subPropertyOf`
- and other elements of RDFS such as `rdfs:domain` and `rdfs:range`

Answering a query with RDFS vocabulary

Retrieve all the educators in DBpedia

```
SELECT ?educator
WHERE
{
   ?educator  rdf:type
   <http://dbpedia.org/class/yago/Educator110045713> .
}
```

Answering a query with RDFS vocabulary

The answer to the previous query should be the same as for the following query:

```
SELECT ?educator
WHERE
{
   { ?educator rdf:type
     <http://dbpedia.org/class/yago/Academician109759069> . }
   UNION
   { ?educator rdf:type
     <http://dbpedia.org/class/yago/Educator110045713> . }
}
```

Open issues about reasoning with ontologies

Two important problems:

- Development of efficient query answering algorithms over large RDF graphs with RDFS vocabulary
- Identification of fragments of OWL that have good expressive power and can be efficiently evaluated

Exploiting the graph structure of RDF

The structure of an RDF graph stores information

It is important to have operators that can deal with this structure

- In particular, navigating an RDF graph is an important functionality

Exploiting the graph structure of RDF

The structure of an RDF graph stores information

It is important to have operators that can deal with this structure

- In particular, navigating an RDF graph is an important functionality

Properties paths in SPARQL allow to express reachability queries

Navigating RDF graphs

Get starring actors in the same movie:

```
SELECT ?actor1 ?actor2
WHERE
{
   ?movie  <http://dbpedia.org/property/starring>  ?actor1 .
   ?movie  <http://dbpedia.org/property/starring>  ?actor2 .
}
```

Navigating RDF graphs

Previous query can be rewritten by using navigation patterns:

Navigating RDF graphs

Previous query can be rewritten by using navigation patterns:

```
SELECT ?actor1 ?actor2
WHERE
{
   ?actor1 ^<http://dbpedia.org/property/starring> ?movie .
   ?movie <http://dbpedia.org/property/starring> ?actor2 .
}
```

Navigating RDF graphs

Previous query can be rewritten by using navigation patterns:

```
SELECT ?actor1 ?actor2
WHERE
{
   ?actor1
   ^<http://dbpedia.org/property/starring>/
    <http://dbpedia.org/property/starring>
   ?actor2 .
}
```

Navigating RDF graphs

Previous query can be rewritten by using navigation patterns:

```
SELECT ?actor1 ?actor2
WHERE
{
   ?actor1
   ^<http://dbpedia.org/property/starring>/
    <http://dbpedia.org/property/starring>
   ?actor2 .
}
```

The expression in red is called a property path

Navigating RDF graphs

Get starring actors that are connected:

```
SELECT ?actor1 ?actor2
WHERE
{
   ?actor1
   (^<http://dbpedia.org/property/starring>/
     <http://dbpedia.org/property/starring>)+
   ?actor2 .
}
```

Navigating RDF graphs

Get starring actors that are connected:

```
SELECT ?actor1 ?actor2
WHERE
{
   ?actor1
   (^<http://dbpedia.org/property/starring>/
     <http://dbpedia.org/property/starring>)+
   ?actor2 .
}
```

Can this query be answered?

- Can it be answered starting from a specific node?

Open issues in exploiting the graph structure of RDF

Some important problems:

- Development of efficient evaluation algorithms for reachability queries over large RDF graphs
- Standardization of a query language where paths are first-class citizens