Linked (Open) Data

Advanced Methods of IR
Elena Demidova
SS 2014

Materials used in the slides:
The wealth of data on the Web

Increasing numbers of individuals and organizations are contributing to data on the Web:

- Web-native companies: Amazon and Yahoo!
- Public bodies: the UK and US governments
- Research initiatives within various scientific disciplines
- …

This data covers many different domains, e.g.

- data about fuel efficiency of our cars
- a multitude of products from different vendors
- or the way our taxes are spent
- …
Neighbourhood Checker (UK)

SELECT YOUR AREA FROM THE MAP BELOW...

Colour coded by level of deprivation

http://neighbourhoodchecker.co.uk/
Key question in publishing the data

How best to provide access to data so it can be most easily reused?
How to enable the discovery of relevant data within the multitude of available data sets?
How to enable applications to integrate data from large numbers of formerly unknown data sources?
Data on the Web: HTML

- HTML is oriented towards structuring **textual documents, not data**.
- As **data is intermingled into the text**, it is hard for applications to extract snippets of structured data from HTML pages.
- HTML **links documents, not objects**.
- HTML links indicate that two documents are **related in some way**, but mostly leave the user to infer the nature of the relationship.

“As of midnight of Tuesday, 3 June, about 180,000 tickets will be up for sale on [www.fifa.com/tickets](http://www.fifa.com/tickets), in addition to those tickets that are currently available,” Fifa said in a statement.

Data on the Web: Microformats

<a class="h-card" href="http://waterpigs.co.uk">Barnaby Walters</a>

class=h-card: This element represents a person.

✓ Microformats can be used to publish structured data describing specific types of entities (people and organizations, events, reviews and ratings), through embedding of data in HTML.
  – Restricted to representing data about a small set of different types of entities and a small set of attributes.
  – It is often not possible to express relationships between entities.
  – Not suitable for sharing arbitrary data on the Web.
Data on the Web: Web APIs

- Web APIs (Amazon Product Advertising API, Flickr API, Twitter API) provide simple query access to structured data over the HTTP protocol.
- Web APIs to provide results in structured data formats such as XML and JSON
- Explosion in small, specialized applications (or mashups) that combine data from several sources, each of which is accessed through an API specific to the data provider.
Data on the Web: Web APIs

- **Specialized API** for each data set. Significant effort is required to integrate each novel data set into an application.
- Web APIs use **local scope object identifiers** – e.g., a product identifier 123456 - meaningless when taken out of the context of that specific API. There is no standard mechanism to refer to items described by one API in data returned by another.
- Data returned from Web APIs typically exists as **isolated fragments**, lacking reliable onward links signposting the way to related data.
- Web APIs make data accessible **on the Web**, they **do not place it truly in the Web**, making it linkable and therefore discoverable.
From data islands to a global data space

Linking data across the Web requires a standard mechanism for specifying the existence and meaning of connections between items described in this data.

Linked Data enables links between items in different data sources connecting these sources into a single global data space.

The use of Web standards and a common data model make it possible to implement generic applications that operate over the complete data space.
Linked Data: principles

- P1: Use URIs as names for things
- P2: Use HTTP URIs so that people can look up those names.
- P3: When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- P4: Include links to other URIs, so that they can discover more things.

http://www.w3.org/DesignIssues/LinkedData.html
P1: Naming things with (HTTP) URIs

HTTP URIs provide a simple way to create globally unique names. HTTP URIs are created in a decentralized fashion, as every owner of a domain name, or delegate of the domain name owner, may create new URI references. HTTP URIs serve as a means of accessing information describing the identified entity.
P2: Making URIs deferenceable

Any HTTP URI should be dereferenceable:

✓ HTTP clients can look up the URI using the HTTP protocol and retrieve a description of the resource that is identified by the URI.

✓ Descriptions of resources are embodied in the form of Web documents.

✓ Descriptions intended to be read by humans are often represented as HTML.

✓ Descriptions that are intended for consumption by machines are represented as RDF data.

✓ HTTP content negotiation can be used to indicate the format preferred by the client.
P2. Making URIs deferenceable

Essential: do not confuse the objects themselves with the Web documents that describe the objects.

Common practice: use different URIs to identify the real-world object and the document that describes it.

Allows separate statements about an object and a document that describes the object.

http://biglynx.co.uk/people/dave-smith (URI identifying the person Dave Smith)

http://biglynx.co.uk/people/dave-smith.rdf (URI identifying the RDF/XML document describing Dave Smith)

http://biglynx.co.uk/people/dave-smith.html (URI identifying the HTML document describing Dave Smith)
P2: Dereferencing a HTTP URI: 303 redirect

Real-world objects, like houses or people, can not be transmitted over the wire using the HTTP protocol.

It is also not possible to directly dereference URIs that identify real-world objects.

In the 303 URIs strategy, the server responds with the HTTP response code 303 See Other and the URI of a Web document which describes the real-world object.

In a second step, the client dereferences this new URI and gets a Web document describing the real-world object.

Thus the 303 URIs strategy requires two HTTP requests to retrieve a single description of a real-world object.
P2: Dereferencing a HTTP URI: 303 URIs in 4 steps

1. **Client**: HTTP GET request on a URI identifying a real-world object or abstract concept. LD Client: Accept: application/rdf+xml. HTML browsers: Accept: text/html header.

2. **Server**: answers HTTP 303 See Other response code and sends the URI of a Web document that describes the real-world object or abstract concept in the requested format.

3. **Client**: performs an HTTP GET request on the URI returned by the server.

4. **Server**: answers with a HTTP response code 200 OK and sends the client the requested document, describing the original resource in the requested format.
P2: Dereferencing a HTTP URI: 303 redirect example

1. Client:
GET /people/dave-smith HTTP/1.1
Host: biglynx.co.uk
Accept: text/html;q=0.5, application/rdf+xml

2. Server:
HTTP/1.1 303 See Other
Location: http://biglynx.co.uk/people/dave-smith.rdf
Vary: Accept

3. Client:
GET /people/dave-smith.rdf HTTP/1.1
Host: biglynx.co.uk
Accept: text/html;q=0.5, application/rdf+xml

4. Server
HTTP/1.1 200 OK
Content-Type: application/rdf+xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
><rdf:Description rdf:about="http://biglynx.co.uk/people/dave-smith"
><rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
<foaf:name>Dave Smith</foaf:name>
...
P2: Dereferencing a HTTP URI: Hash URIs

*Hash URI* strategy: URIs may contain a special part that is separated from the base part of the URI by a hash symbol (#) - the *fragment identifier*.

Example: A vocabulary of Small and Medium-sized Enterprises

http://biglynx.co.uk/vocab/sme/

*Fragment identifiers* to identify the different vocabulary terms defined in the document.

http://biglynx.co.uk/vocab/sme#SmallMediumEnterprise

http://biglynx.co.uk/vocab/sme#Team

Hash URIs avoid multiple requests

Response contains descriptions of multiple objects

Hash URIs are often used to identify terms within RDF vocabularies, as the definitions of RDF vocabularies are usually rather small.
P3: Resource Description Framework (RDF)

Resource Description Framework (RDF) – the mechanism for specifying the existence and meaning of connections between items described in this data. The RDF data model represents information as node-and-arc-labeled directed graphs.

In RDF, a description of a resource is represented as a number of *triples*. The three parts of each triple are called its *subject*, *predicate*, and *object*.

The *subject* is the URI identifying the described resource.

The *object* can either be a simple *literal value*, (e.g. a string, number, or date; or the URI of another resource that is somehow related to the subject.

The *predicate*, indicates relation between subject and object. The predicate is identified by a URI. Predicate URIs come from *vocabularies*, collections of URIs that can be used to represent information about a certain domain.
P3: Literal triples and RDF links

**Literal Triples** have an **RDF literal** such as a string, number, or date as the object.

- http://biglynx.co.uk/people/matt-briggs
- http://xmlns.com/foaf/0.1/nick
- "Matty"

**RDF Links** describe the relationship between two resources. RDF links consist of three URI references.

- http://biglynx.co.uk/people/matt-briggs
- http://xmlns.com/foaf/0.1/knows
- http://biglynx.co.uk/people/dave-smith
- http://biglynx.co.uk/vocab/sme#leads
- http://biglynx.co.uk/teams/production
P3: RDF link example

http://biglynx.co.uk/people/matt-briggs
http://xmlns.com/foaf/0.1/knows
http://biglynx.co.uk/people/linda-meyer
P3: RDF links vs. HTML links

**RDF links things, not just documents:**

matt-briggs knows linda-meyer

RDF links assert connections between the entities described in the data fragments *explicitly*.

**RDF links are typed:**

matt-briggs knows linda-meyer

HTML links indicate that two documents are *related in some way*. RDF enables the data publisher to state *explicitly* the nature of the connection.
P4: External RDF links

**Relationship Links** point at related things in other data sources, for instance, other people, places or genes.

**Identity Links** point at URI aliases used by other data sources to identify the same real-world object or abstract concept.

**Vocabulary Links** point from data to the definitions of the vocabulary terms that are used to represent the data, as well as from these definitions to the definitions of related terms in other vocabularies.
P4: External RDF links

**Relationship Links** point at related things in other data sources, for instance, other people, places or genes. E.g., relationship links enable people to point to background information about the place they live, or to bibliographic data about the publications they have written.

```
<http://biglynx.co.uk/people/dave-smith>
  rdf:type foaf:Person ;
  foaf:name "Dave Smith" ;
  foaf:based_near <http://sws.geonames.org/3333125/> ;
  foaf:based_near <http://dbpedia.org/resource/Birmingham> ;
  foaf:topic_interest <http://dbpedia.org/resource/Wildlife_photography> ;
```
P4: External RDF links

Identity Links point at URI aliases used by other data sources to identify the same real-world object or abstract concept. Identity links enable clients to retrieve further descriptions about an entity from other data sources. Identity links enable different views of the world to be expressed on the Web of Data.

<http://www.dave-smith.eg.uk#me>
<http://www.w3.org/2002/07/owl#sameAs>
<http://biglynx.co.uk/people/dave-smith>.

Warning: The OWL semantics treat RDF statements as facts rather then as claims by different information providers.
P4: External RDF links

**Vocabulary Links** point from data to the definitions of the vocabulary terms that are used to represent the data, as well as from these definitions to the definitions of related terms in other vocabularies. Vocabulary links make data self-descriptive and enable Linked Data applications to understand and integrate data across vocabularies.

```
<http://biglynx.co.uk/vocab/sme#SmallMediumEnterprise>

rdf:type rdfs:Class ;
  rdfs:label "Small or Medium-sized Enterprise" ;
  rdfs:subClassOf <http://umbel.org/umbel/sc/Business> ;
  rdfs:subClassOf <http://sw.opencyc.org/concept/Mx4rvVjQNPwpEbGdrCN5y29ycA> ;
  rdfs:subClassOf <http://rdf.freebase.com/ns/m/0qb7t>.
```
Rationale for adopting Linked Data

**A unifying data model.** Linked Data relies on RDF as a single, unifying data model that provides the globally unique identification of entities and allows different schemata to be used in parallel.

**A standardized data access mechanism.** Linked Data commits itself to a specific pattern of using the HTTP protocol. The data sources can be accessed using generic data browsers and enables the complete data space to be crawled by search engines.

**Hyperlink-based data discovery.** Linked Data allows hyperlinks to be set between entities in different data sources. These data links connect all Linked Data into a single global data space and enable Linked Data applications to discover new data sources at run-time.

**Self-descriptive data.** Linked Data relies on shared vocabularies, makes the definitions of these vocabularies retrievable, and by allows interlinking of terms from different vocabularies.
Data sets published on the Web as Linked Data
Data sets published on the Web as Linked Data

http://blog.law.cornell.edu/voxpop/files/2013/01/lod-cloud_2010.png
Data sets published on the Web as Linked Data

The distribution of triples by domain

The distribution of links by domain

http://lod-cloud.net/state/
Work in groups (15 minutes)

Assignment: Design a keyword search system for Linked Data

Requirements:
Automatically discover Linked Data sources
Allow keyword matching in any part of data (labels, URIs, text)
++ Allow keyword search across connected entities
++ Ensure data freshness

• List the system components and data structures
• Describe the offline processing
• Describe the query processing
• Compare to the database keyword search
Sig.ma Linked Data search engine

http://sig.ma/
Federated query / search in LOD: challenges

- **Source discovery**
  - How to find sources containing potentially relevant data
  - Existing catalogues (e.g. DataHub) are incomplete
  - Statistics describing data sources are sparse

- **Source selection**
  - How to select sources for execution of potential query
  - Data quality issues

- **Query interpretation**
  - Schema definitions are often not available
  - External RDF links are rare

- **Result merging**
  - Results can be contradictory
References and further reading

References:

Further reading: