TECHNISCHE UNIVERSITÄT DRESDEN

WORKING WITH KNOWLEDGE GRAPHS

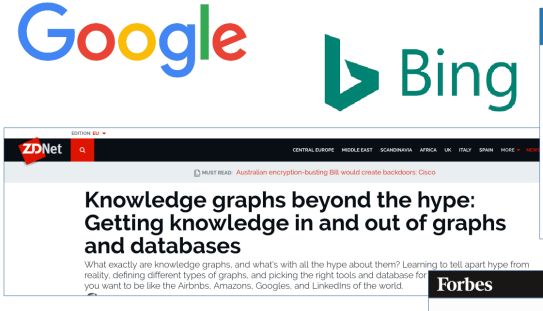
Lecture 1: Wikidata, RDF, and SPARQL

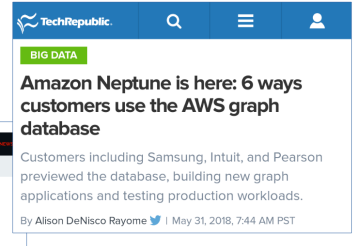
Markus Krötzsch

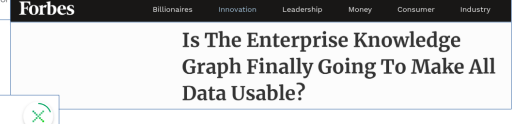
Knowledge-Based Systems, TU Dresden

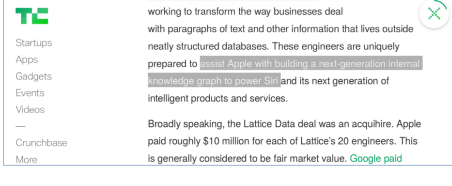
EDBT Summer School 2019

Knowledge Graphs Everywhere





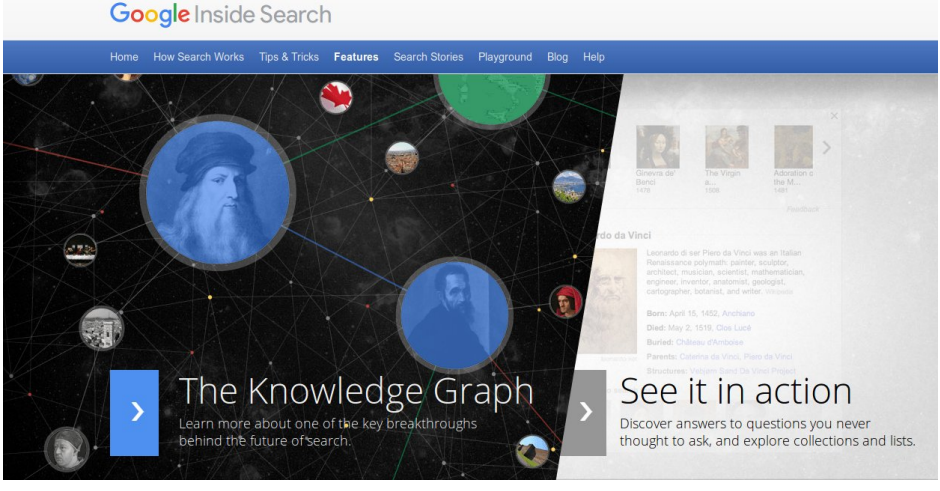




What is a Knowledge Graph?

What is a Knowledge Graph?

The original “Knowledge Graph” (Google, 2012):



Many knowledge graphs, many technologies

There are a number of widely used publicly available knowledge graphs:

. . . and a variety of technologies for working with them:

What is special about Knowledge Graphs?

A Knowledge Graph is a data set that is:

- structured (in the form of a specific data structure)
- normalised (consisting of small units, such as vertices and edges)
- connected (defined by the – possibly distant – connections between objects)

Moreover, knowledge graphs are typically:

- explicit (created purposefully with an intended meaning)
- declarative (meaningful in itself, independent of a particular implementation or algorithm)
- annotated (enriched with contextual information to record additional details and meta-data)
- non-hierarchical (more than just a tree-structure)
- large (millions rather than hundreds of elements)

(Counter-)Examples

Typical knowledge graphs:

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[(though hardly annotated)]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[though hardly annotated]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[though hardly annotated]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[though hardly annotated]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
- WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[(though hardly annotated)]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
- WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model
- Global data from schema.org: maybe not very connected

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[(though hardly annotated)]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
- WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model
- Global data from schema.org: maybe not very connected
- Document stores (Lucene, MongoDB, etc.): structured, but not normalised; connections secondary

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[(though hardly annotated)]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
- WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model
- Global data from schema.org: maybe not very connected
- Document stores (Lucene, MongoDB, etc.): structured, but not normalised; connections secondary

Primarily not knowledge graphs:

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[though hardly annotated]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
- WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model
- Global data from schema.org: maybe not very connected
- Document stores (Lucene, MongoDB, etc.): structured, but not normalised; connections secondary

Primarily not knowledge graphs:

- Wikipedia: mostly unstructured text; not normalised; connections (links) important but secondary (similar: The Web)

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia[though hardly annotated]
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
- WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model
- Global data from schema.org: maybe not very connected
- Document stores (Lucene, MongoDB, etc.): structured, but not normalised; connections secondary

Primarily not knowledge graphs:

- Wikipedia: mostly unstructured text; not normalised; connections (links) important but secondary (similar: The Web)
- Relational database of company X: structured and possibly normalised, but no focus on connections (traditional RDBMS support connectivity queries only poorly)

Wikidata

A Free Knowledge Graph

Wikidata [CACM 2014]

- Wikipedia's knowledge graph
- Free, community-built database
- Large graph
  (August 2019: >733M statements on >58M entities)
- Large, active community (>12,000 active editors in July 2019)
- Many applications

Freely available, relevant, and active knowledge graph

Many applications (1)

As of today, Wikidata content has been used in many ways.

Wikipedia & the Wikimedia community:

- Wikipedia inter-language links (see any Wikipedia page)
- Data displays in pages (auto-generated info boxes, article placeholders, result tables, . . .)
- Quality checks & edit-a-thons

External re-uses of data:

- Application-specific data-excerpts (e.g., Eurowings in-flight app)
- Data integration and quality control (e.g., Google checks own KG against Wikidata)
- Authority control & identity provider (VIAF, Open Streetmaps, DBLP, . . . link their content to Wikidata)
- Data-driven journalism (individual analyses as well as data-driven information portals)

Who is Grover Cleveland

Tap to Edit

OK. Check it out:

KNOWLEDGE

Grover Cleveland

22nd and 24th president of the United States

Stephen Grover Cleveland was an American politician and lawyer who was the 22nd and 24th President of the United States. He won the popular vote for three presidential elections – in 1884, 1888, and 1892 – and was one of two Democrats to be elected president during the era of Republican political domination dating from 1861 to 1933. He was also the first and to date only President in American history to serve two non-consecutive terms in office.

See More on Wikipedia

Date of birth	March 18, 1837
Birthplace	Caldwell
Date of death	June 24, 1908
Deathplace	Princeton

Many applications (2)

As of today, Wikidata content has been used in many ways.

In research:

- Test data for KG-related algorithms
- Training data for machine-learning approaches
- Wikidata as a subject of study (social dynamics, internationality, biases, . . .)

Uses by Wikidata community:

- Software-supported error and vandalism detection
- Feature-based integration with other datasets
- Data-driven statistics as a measure of progress

What is Wikidata?

Wikidata is often described as “the free knowledge base that anyone can edit” or the “knowledge graph of Wikipedia”

What is Wikidata?

Wikidata is often described as “the free knowledge base that anyone can edit” or the “knowledge graph of Wikipedia”

It is useful to distinguish several of these aspects:

Wikidata is . . .

- **. . . a Wikimedia project** like Wikipedia and Wikimedia Commons; represented and supported by the Wikimedia Foundation (WMF)
- **. . . a dataset** that can be downloaded and freely used and distributed
- **. . . a website** through which the data can be viewed and modified
- **. . . a community** of volunteer editors that creates and controls all content

What is Wikidata?

Wikidata is often described as “the free knowledge base that anyone can edit” or the “knowledge graph of Wikipedia”

It is useful to distinguish several of these aspects:

Wikidata is . . .

- **. . . a Wikimedia project** like Wikipedia and Wikimedia Commons; represented and supported by the Wikimedia Foundation (WMF)
- **. . . a dataset** that can be downloaded and freely used and distributed
- **. . . a website** through which the data can be viewed and modified
- **. . . a community** of volunteer editors that creates and controls all content

> “And like all uses of the word ‘community,’ you were never quite sure what or who it was.” – Terry Pratchett (Jingo, 1997)

Tim Berners-Lee (Q80)

British computer scientist

edit

TimBL | Sir Tim Berners-Lee | Timothy John Berners-Lee | TBL | Tim Berners Lee | T. Berners-Lee | T Berners-Lee | Tim Berners-Lee | T.J. Berners-Lee

Tim Berners-Lee (Q80)

British computer scientist

edit

TimBL | Sir Tim Berners-Lee | Timothy John Berners-Lee | TBL | Tim Berners Lee | T. Berners-Lee | T Berners-Lee | Tim Berners-Lee | T.J. Berners-Lee

instance of	human	edit
	1 reference	

Tim Berners-Lee (Q80)

British computer scientist

edit

TimBL | Sir Tim Berners-Lee | Timothy John Berners-Lee | TBL | Tim Berners Lee | T. Berners-Lee | T Berners-Lee | Tim Berners-Lee | T.J. Berners-Lee

instance of	human		edit
	1 reference		
employer	CERN		edit
	start time	1984	
	end time	1994	
	position held	Fellow	
	0 references		
			+ add reference

Tim Berners-Lee (Q80)

British computer scientist edit

TimBL | Sir Tim Berners-Lee | Timothy John Berners-Lee | TBL | Tim Berners Lee | T. Berners-Lee | T Berners-Lee | Tim Berners-Lee | T.J. Berners-Lee

instance of	human 1 reference	edit
employer	CERN start time: 1984 end time: 1994 position held: Fellow 0 references	edit + add reference
award received	Queen Elizabeth Prize for Engineering point in time: 2013 together with: Robert Kahn, Vint Cerf, Louis Pouzin, Marc Andreessen 1 reference	edit

Tim Berners-Lee (Q80)

British computer scientist

TimBL | Sir Tim Berners-Lee | Timothy John Berners-Lee | TBL | Tim Berners Lee | T. Berners-Lee | T Berners-Lee | Tim Berners-Lee | T.J. Berners-Lee

edit

instance of	human	edit
	1 reference	
employer	CERN	edit
	start time: 1984	
	end time: 1994	
	position held: Fellow	
	0 references	
	+ add reference	
award received	Queen Elizabeth Prize for Engineering	edit
	point in time: 2013	
	together with: Robert Kahn, Vint Cerf, Louis Pouzin, Marc Andreessen	
	1 reference	

Wikipedia (126 entries) edit

af	Tim Berners-Lee
als	Tim Berners-Lee
am	ቲም ባርነርስ ሊ
an	Tim Berners-Lee
ar	تيم بيرنرز لي
arz	تيم بيرنرز لى
ast	Tim Berners-Lee
as	টিম বার্নাৰ্ছ লী
azb	تیم برنرز لی
az	Tim Berners-Li
bar	Tim Berners-Lee
bat_smg	Tim Berners-Lee
ba	Тим Бернерс-Ли
be_x_old	Тым Бэрнэрз-Лі
be	Цім Бернерс-Лі
bg	Тим Бърнърс-Лий
bn	টিম বার্নার্স-লি
br	Tim Berners-Lee
bs	Tim Berners-Lee
ca	Tim Berners-Lee
ce	Бернерс-Ли, Тим
ckb	تیم بێرنەرز لی

Wikidata's IDs

Why does Wikidata use abstract (numeric) QIDs and PIDs rather than something more readable?

Wikidata's IDs

Why does Wikidata use abstract (numeric) QIDs and PIDs rather than something more readable?

International

- Identifiers work for any language and cultural backgrounds

Stable

- Labels can change without IDs changing
- Multiple entities can have the same label
- IDs of deleted entities are never used again

Convenient

- Numeric IDs are quite short
- Uniform format is practical

How to find the ID of an item?

Main methods:

(1) Use the auto-completing search bar on wikidata.org

(2) Go to the item's Wikipedia page and select "Wikdata item" from the sidebar

Several other projects have started to use Wikidata IDs for tagging and inter-linking.

Wikidata statements

Wikidata's basic information units

- Built from Wikidata items ("CERN", "Vint Cerf"), Wikidata properties ("award received", "end time"), and data values ("2013")
- Based on directed edges ("Tim Berners-Lee −employer→ CERN")
- Annotated with property-value pairs ("end time: 1994")
  - same property can have multiple annotation values ("together with: Robert Kahn, Vint Cerf, . . . ")
  - same properties/values used in directed edges and annotations
- Properties have pre-defined datatypes
- Items and properties can be subjects/values in statements
- Multi-graph

edit label

Elizabeth Taylor (Q34851)

Elizabeth Rosemond Taylor | Liz Taylor | Dame Elizabeth Rosemond Taylor

British-American actress

instance of: Elizabeth Taylor is a(n) human

Human relationships

Own statements | From related entities

spouse

8 statements

- Larry Fortensky (construction worker and seventh husband of Elizabeth Taylor)
  - end time : 1996-10-31
  - start time : 1991-10-06
- John Warner (Republican politician and Secretary of the Navy from the United States)
  - end time : 1982-11-07
  - start time : 1976-12-04
- Richard Burton (Welsh actor)
  - start time : 1975-10-10
  - end time : 1976-07-29
- Richard Burton (Welsh actor)
  - start time : 1964-03-15
  - end time : 1974-06-26
- Eddie Fisher (American entertainer and singer)
  - end time : 1964-03-06
  - start time : 1959-05-12
- Mike Todd (American theatre and film producer)
  - end time : 1958-03-22
  - start time : 1957-02-02
- Michael Wilding (English television and film actor)
  - end time : 1957-01-30
  - start time : 1952-02-21
- Conrad Hilton, Jr. (American hotelier)
  - end time : 1951-01-29
  - start time : 1950-05-06

Links

- **Wikidata page**
- **Official website**
- **Wikipedia article**
- **Reasonator**

Identifiers

SFDb person ID	75200
Elonet person ID	224907
PORT person ID	7869
AllMovie artist ID	p70015

Representing Knowledge Graphs

Tales of two graphs

Tales of two graphs

Once upon a time, there was … the World Wide Web

- A huge network of many connections,
- lacking meaning and data

“Let’s define the meaning of hyperlinks and add data values!”

$\rightsquigarrow$ Resource Description Framework (RDF)

Tales of two graphs

Once upon a time, there was … the World Wide Web

- A huge network of many connections,
- lacking meaning and data

"Let's define the meaning of hyperlinks and add data values!"

⇝ Resource Description Framework (RDF)

Once upon a time, there were … document databases

- Huge collections of objects described by attributes,
- lacking connections and relationships

"Let's add a type of 'edge' object that can connect documents!"

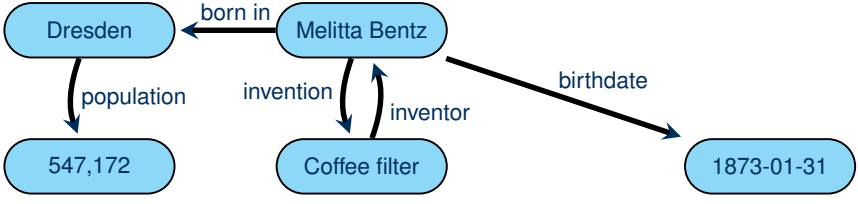
⇝ Property Graph

RDF

Conceived as a data exchange format for describing data about (Web) resources.

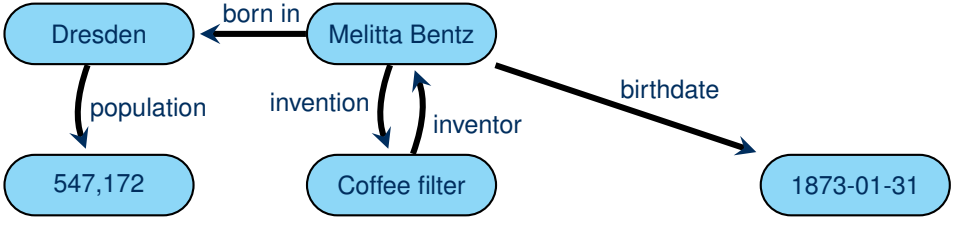
- Defined by a W3C standard (patent-free & freely accessible)
- Current version: RDF 1.1, 2014
- Roughly: directed, edge-labelled graphs

Example of such a graph:



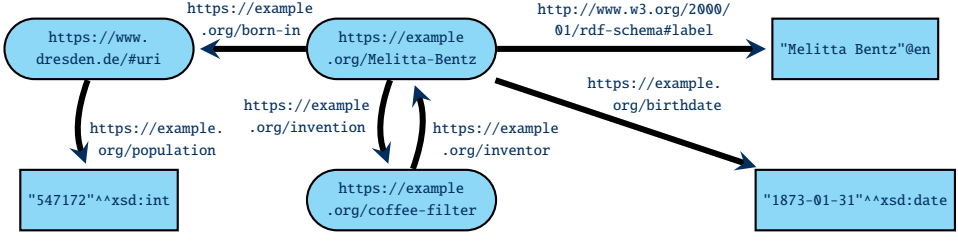
Encoding RDF Graphs

RDF uses IRIs as identifiers, and data literals for encoding values:



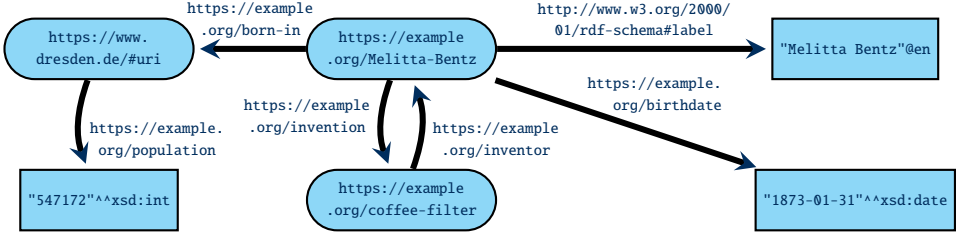
Encoding RDF Graphs

RDF uses IRIs as identifiers, and data literals for encoding values:



Encoding RDF Graphs

RDF uses IRIs as identifiers, and data literals for encoding values:



This graph can be encoded in by listing edges (“triples”), one per line:

```
<https://example.org/Melitta-Bentz>  <http://www.w3.org/2000/01/rdf-schema#label>  "Melitta Bentz"@en .
<https://example.org/Melitta-Bentz>  <https://example.org/birthdate>  "1873-01-31"^^<http://www.w3.org/2001/XMLSchema#date> .
<https://example.org/Melitta-Bentz>  <https://example.org/invention>  <https://example.org/coffee-filter> .
<https://example.org/Melitta-Bentz>  <https://example.org/born-in>  <https://www.dresden.de/#uri> .
<https://www.dresden.de/#uri>  <https://example.org/population>  "547172"^^<http://www.w3.org/2001/XMLSchema#int> .
<https://example.org/coffee-filter>  <https://example.org/inventor>  <https://example.org/Melitta-Bentz> .
```

Encoding RDF Graphs More Nicely

The popular Turtle syntax supports several simplifications to enhance readability:

```
PREFIX ex: <https://example.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

ex:Melitta-Bentz  rdfs:label  "Melitta Bentz"@en ;
                  ex:birthdate  "1873-01-31"^^xsd:date ;
                  ex:invention  ex:coffee-filter ;
                  ex:born-in  <https://www.dresden.de/#uri> .
<https://www.dresden.de/#uri>  ex:population  547172 .
ex:coffee-filter  ex:inventor  ex:Melitta-Bentz .
```

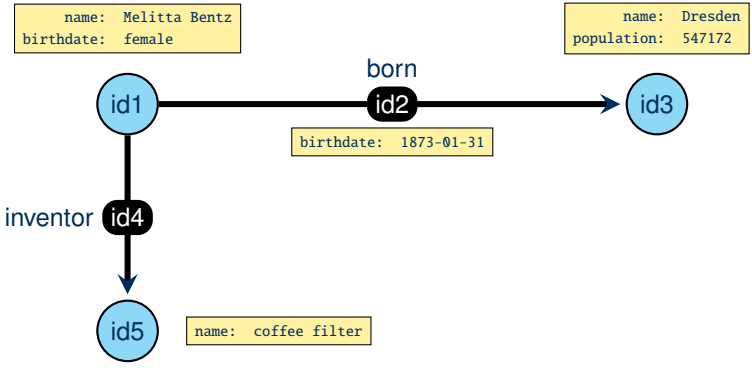
(Shown here: abbreviation IRIs using PREFIX, continued triples with semi-colon, simplified number syntax – several further features exist)

Property Graph

Conceived as a data management paradigm for adding a graph view to object DBMS.

- Data model as represented by Apache Tinkerpop Java library
- Various interpretations (Java API, relational DB extension, RDF extension)
- Directed, labelled multi-graphs + attribute-value maps for nodes and edges

Example of such a graph:



Property Graph: Specifics and Unspecifics

Edge and vertex labels

- arbitrary strings (unlike IRIs in RDF: no keys!)

Keys used in key-value maps

- arbitrary strings (subsidiary; no relation to graph vertices)

Values used in key-value maps

- Implementation-specific; for example:
  - OpenCypher: INTEGER, FLOAT, STRING, BOOLEAN (+list type)
  - SAP HANA Graph: SQL datatypes (+own extensions)
  - Amazone Neptune: RDF-like datatypes (+own extensions)
- No relation to graph: cannot refer to graph vertices

Property Graph: Specifics and Unspecifics

Edge and vertex labels

- arbitrary strings (unlike IRIs in RDF: no keys!)

Keys used in key-value maps

- arbitrary strings (subsidiary; no relation to graph vertices)

Values used in key-value maps

- Implementation-specific; for example:
  - OpenCypher: INTEGER, FLOAT, STRING, BOOLEAN (+list type)
  - SAP HANA Graph: SQL datatypes (+own extensions)
  - Amazone Neptune: RDF-like datatypes (+own extensions)
- No relation to graph: cannot refer to graph vertices

Limitations that hurt when working with knowledge graphs:

- "Inner" key-value maps cannot refer to vertices in "outer" graph
- Relationship types and keys cannot be described in graph
- Lack of standard; lack of exchange syntax (knowledge sharing difficult)

Wikidata, RDF, and SPARQL

Wikidata in RDF

Wikidata is internally stored in the document-centric form using a JSON format

Data is converted to RDF for several purposes:

- Offering complete data dumps for external uses
- Providing entity-specific linked data exports via a Web API
- Importing data into Wikidata's SPARQL query service

Wikidata in RDF

Wikidata is internally stored in the document-centric form using a JSON format

Data is converted to RDF for several purposes:

- Offering complete data dumps for external uses
- Providing entity-specific linked data exports via a Web API
- Importing data into Wikidata's SPARQL query service

Wikidata's graph view has many commonalities with RDF:

- Based on directed, labelled, multi-graph
- Properties have own identity in graph
- Order and in-page context of statements does not matter

Wikidata in RDF

Wikidata is internally stored in the document-centric form using a JSON format

Data is converted to RDF for several purposes:

- Offering complete data dumps for external uses
- Providing entity-specific linked data exports via a Web API
- Importing data into Wikidata's SPARQL query service

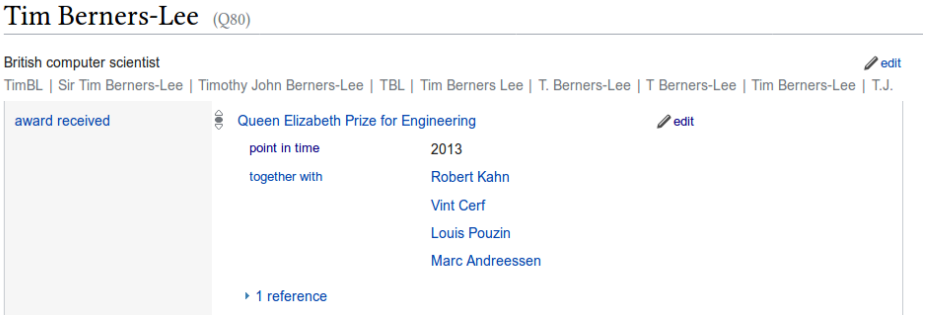
Wikidata's graph view has many commonalities with RDF:

- Based on directed, labelled, multi-graph
- Properties have own identity in graph
- Order and in-page context of statements does not matter

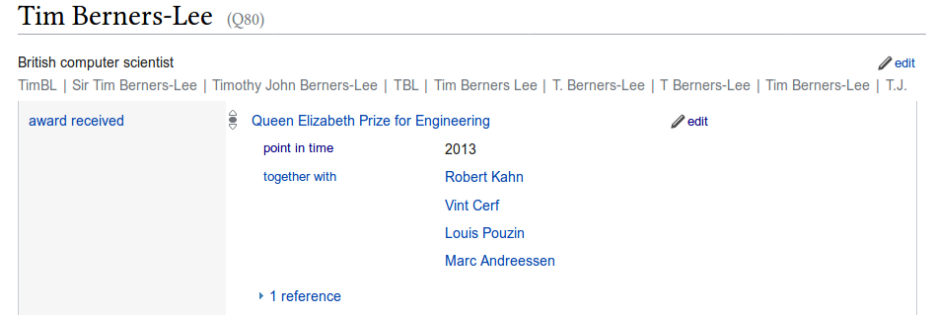
However, there are also some important differences:

- Wikidata statements can have annotations and references
- Wikidata property types do not correspond to XML Schema types
- Wikidata IDs are not immediately IRIs

Encoding statements in RDF (1)



Encoding statements in RDF (1)



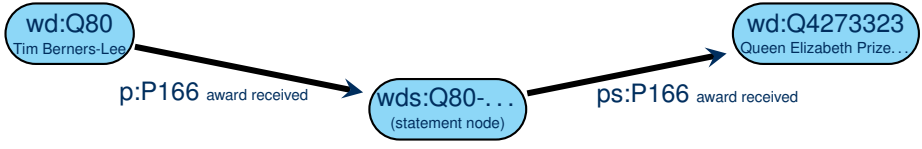
Where to store the annotations?

Note: For prefix declarations, see

`https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format`

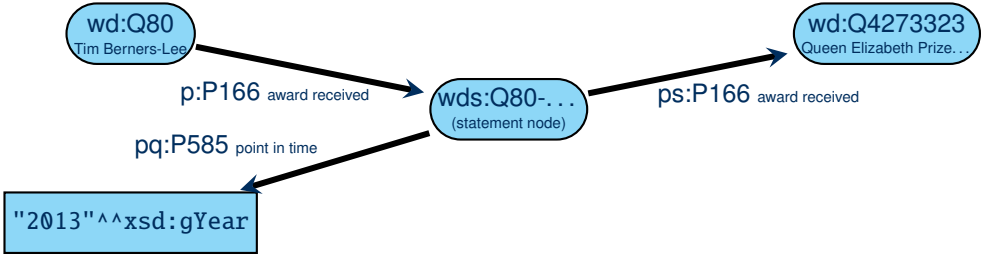
Encoding statements in RDF (2)

We can encode statements in the style of reification:



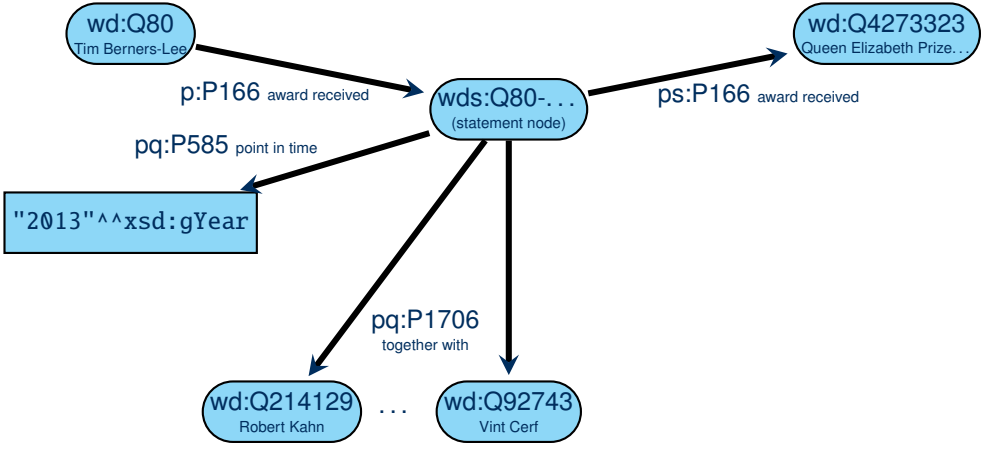
Encoding statements in RDF (2)

We can encode statements in the style of reification:



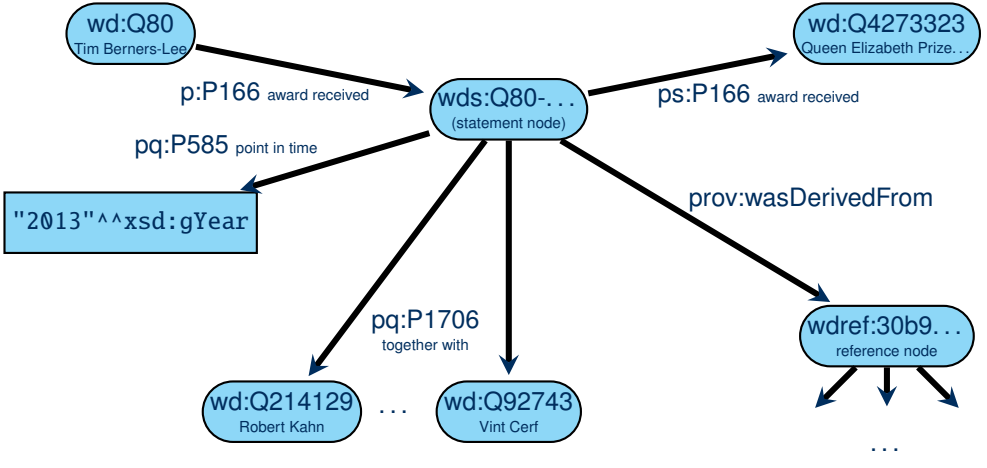
Encoding statements in RDF (2)

We can encode statements in the style of reification:



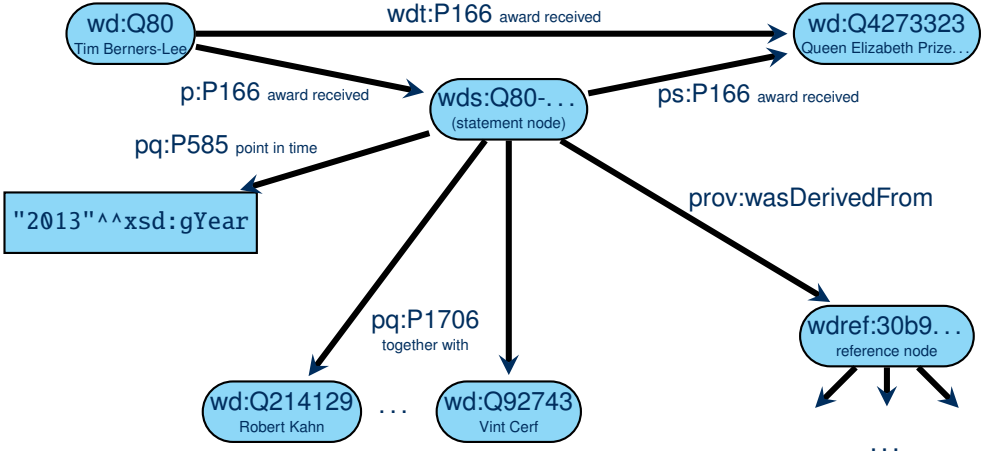
Encoding statements in RDF (2)

We can encode statements in the style of reification:



Encoding statements in RDF (2)

We can encode statements in the style of reification:



Finishing the RDF encoding

Statements in Wikidata:

- Constitute the largest part of the RDF data
- RDF-encoding introduces over 50K RDF properties

Encoding other parts of Wikidata:

- Labels, descriptions, aliases are encoded as RDF literals (with language), linked from subject with rdfs:label, schema:description, and skos:altLabel, respectively
- Sitelinks are encoded using property schema:about (from article page URL to Wikidata entity IRI)

Finishing the RDF encoding

Statements in Wikidata:

- Constitute the largest part of the RDF data
- RDF-encoding introduces over 50K RDF properties

Encoding other parts of Wikidata:

- Labels, descriptions, aliases are encoded as RDF literals (with language), linked from subject with rdfs:label, schema:description, and skos:altLabel, respectively
- Sitelinks are encoded using property schema:about (from article page URL to Wikidata entity IRI)

Available RDF data:

- Full dumps are generated weekly (currently >7.9B triples, 55GiB Turtle.gz)
  For download see `https://dumps.wikimedia.org/wikidatawiki/entities/`
- Generate smaller partial dumps (by Benno Fünfstück, experimental): https://tools.wmflabs.org/wdumps/
- Linked data exports are provided through content negotiation
  Alternatively, directly use data URLs like `http://www.wikidata.org/wiki/Special:EntityData/Q80.nt`

Querying with SPARQL

What are the ten largest cities with a female mayor?

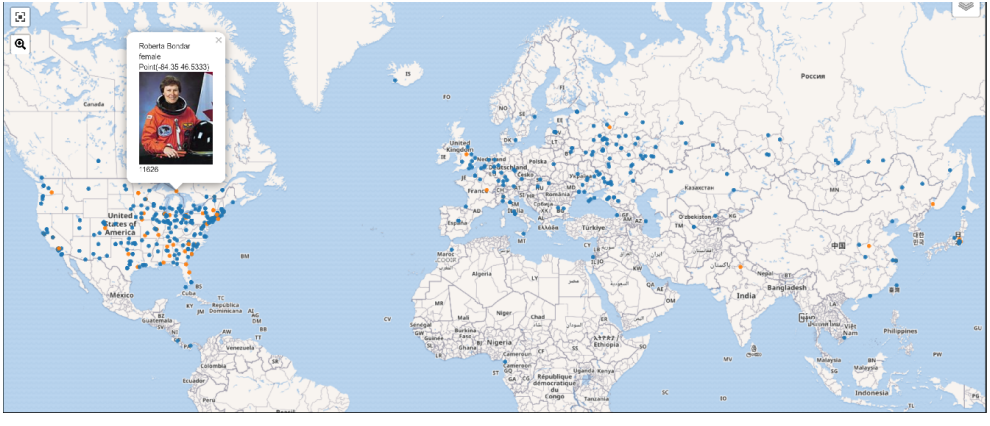
What are the ten largest cities with a female mayor?

10 results in 2309 ms Code Download Link

cityLabel	mayorLabel	population
Tokyo	Yuriko Koike	13929286
Mexico City	Claudia Sheinbaum	8918653
Hong Kong	Carrie Lam	7409800
Baghdad	Zekra Alwach	6960000
Surabaya	Tri Rismaharini	4975000
Yokohama	Fumiko Hayashi	3748482
Rome	Virginia Raggi	2873494
Taichung	Lu Shiow-yen	2803894
Chicago	Lori Lightfoot	2722389
Guayaquil	Cynthia Viteri	2698077

Where are people born who travel to space?

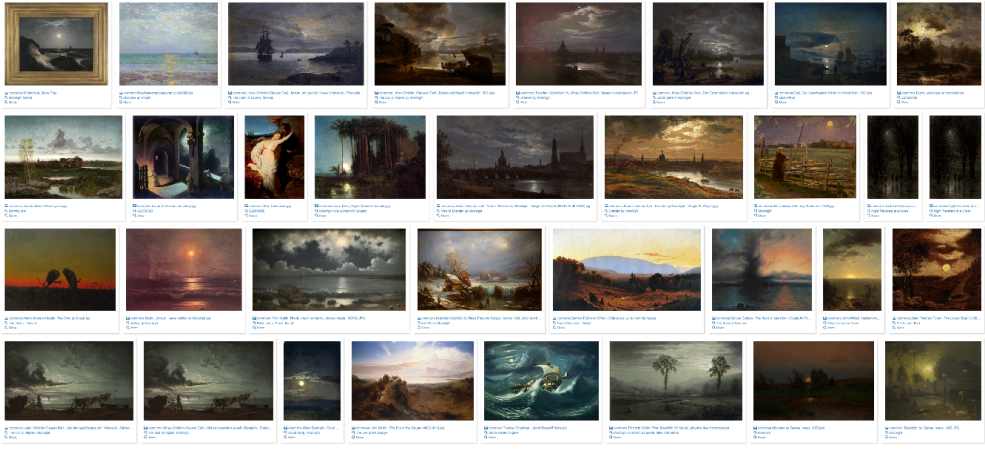
(colour-coded by gender)



Which days of the week do disasters occur?

Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	25	33	22	18	26	28	23
2	24	26	23	23	22	32	12
3	24	27	21	31	23	28	38
4	24	25	33	25	26	26	24
5	37	23	32	18	19	17	29
6	25	28	32	20	24	33	22
7	18	22	25	16	22	18	17
8	32	28	19	25	22	23	19
9	20	25	29	29	27	21	23
10	20	20	19	14	25	25	29
11	30	34	28	23	22	20	20
12	41	33	27	30	20	20	23
13	35	26	29	21	25	24	25
14	24	23	27	23	22	28	17

Which 19th century paintings show the moon?



Which films co-star more than one future head of government?

Star in the Dust	1956 film by Charles F. Haas	2	Clint Eastwood, mayor; George Wallace, Governor of Alabama
The Two Who Stole the Moon	1962 Polish film by Jan Batory	2	Jarosław Kaczyński, Prime Minister of Poland; Lech Kaczyński, Mayor of Warsaw
Ragasiya Police 115	1968 film by B. R. Panthulu	2	M. G. Ramachandran, Chief Minister of Tamil Nadu; Jayalalithaa, Chief Minister of Tamil Nadu
Québec : Duplessis et après…	documentary	2	Bernard Landry, Premier of Quebec; René Lévesque, Premier of Quebec
Q3541438	1994 film by Claude Lanzmann	2	Ariel Sharon, Prime Minister of Israel; Ehud Barak, Prime Minister of Israel
Batman & Robin	1997 American superhero film based on the DC Comics character Batman	2	Arnold Schwarzenegger, Mr. Freeze / Governor of California; Jesse Ventura, Governor of Minnesota

SPARQL Basics

The SPARQL query language (W3C, 2011) is used to query RDF graphs.

- Graph patterns are RDF graphs with variables (in Turtle syntax)
- Can be combined with various operators (UNION, MINUS, OPTIONAL)
- Filter expressions express additional conditions

Example: On Wikidata, find women born after 1921 that have an article on English Wikipedia but no image, ordered from youngest to oldest:

```
SELECT ?person ?born WHERE {
 ?person eg:instanceOf eg:human ;
   eg:gender eg:female ;
   eg:birthdate ?born .
 FILTER (YEAR(?born) >= 1921) # date after 1921
 ?wikipage schema:about ?person ; # article about ?person
   schema:isPartOf <https://en.wikipedia.org/> . # on Wikipedia En
 MINUS { ?person eg:image ?image }
} LIMIT 100  ORDER BY DESC(?born)
```

SPARQL Basics

The SPARQL query language (W3C, 2011) is used to query RDF graphs.

- Graph patterns are RDF graphs with variables (in Turtle syntax)
- Can be combined with various operators (UNION, MINUS, OPTIONAL)
- Filter expressions express additional conditions

Example: On Wikidata, find women born after 1921 that have an article on English Wikipedia but no image, ordered from youngest to oldest:

```
SELECT ?person ?born WHERE {
  ?person wdt:P31 wd:Q5 ; # instance of human
    wdt:P21 wd:Q6581072 ; # gender: female
    wdt:P569 ?born . # date of birth: ?born
  FILTER (YEAR(?born) >= 1921) # date after 1921
  ?wikipage schema:about ?person ; # article about ?person
    schema:isPartOf <https://en.wikipedia.org/> . # on Wikipedia En
  MINUS { ?person wdt:P18 ?image } # minus person with image
} LIMIT 100  ORDER BY DESC(?born)
```

Path expressions

SPARQL can express reachability queries along paths described by regular expressions.

Example: On Wikidata, find all kinds of rock music with a French label:

```
SELECT ?genre ?genreLabel WHERE {
 ?genre wdt:P279+ wd:Q11399 ; # (indirect) subclass of: rock music
   rdfs:label ?genreLabel . FILTER ( LANG(?genreLabel)="fr" )
}
```

- Further path operators: * (zero or more), | (alternative), ^ (converse edge direction), / (concatenation), . . .
- Can be combined and nested
- However: SPARQL cannot count or return paths

Other notable features

Grouping and aggregates

- Including count, max, min, string concatenation, ...

Variable assignments

- Example: `BIND (SUBSTR(?label, 0, 1) as ?initial)`

Subqueries

- Use whole SPARQL **`SELECT`** blocks inside patterns

SPARQL on Wikidata

Wikidata SPARQL Query Service (WDQS):

- Official query service since mid 2015
- User interface at `https://query.wikidata.org/`
  - Query editing support (auto-completion, suggestions. examples)
  - Support for many different result visualisations
  - With library of example queries that helps to learn SPARQL
- All the data (7.9B triples), live (latency<60s)
- Very liberal configuration:
  - 60sec timeout
  - No limit on result size
  - No limit on parallel queries, but CPU-time budget per client
- Extra SERVICEs in SPARQL (geo, Wikipedia API, labels, ...)

For details, see [ISWC 2018].

Summary and conclusions

Wikidata is currently the largest and fastest-growing free knowledge graph

There are two predominant ways of encoding knowledge graphs:

- **RDF:** W3C labelled graph standard; highly normalised; many datatypes; various DBMS (BlazeGraph, Virtuoso, Amazon Neptune, Stardog, ...)
- **Property Graph:** graph-structured object DB; distinct graph and object layer; diverse implementations (Neo4j, SAP Hana Graph, Amazon Neptune, ...)

Complex data (as in Wikidata) is encoded in RDF by reification

SPARQL is a powerful RDF query language used for live queries on Wikidata

Wikidata is an exciting resource with many uses in research

References and further reading

CACM 2014 Denny Vrandečić, Markus Krötzsch: Wikidata: a free collaborative knowledgebase. Commun. ACM, 57(10):78-85, 2014

ISWC 2018 Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, Adrian Bielefeldt: Getting the Most out of Wikidata: Semantic Technology Usage in Wikipedia's Knowledge Graph. Proc. 17th International Semantic Web Conference (ISWC'18), 2018.

KG2018 Course notes "Knowledge Graphs", TU Dresden, 2018, `https://iccl.inf.tu-dresden.de/web/Knowledge_Graphs_(WS2018/19)/en` (more detailed slides on RDF, SPARQL, Wikidata, Property Graph, and Cypher)

WDQS Wikidata Query Service `https://query.wikidata.org/`; see also SPARQL documentation links on that page