TECHNISCHE UNIVERSITÄT DRESDEN

KNOWLEDGE GRAPHS

Lecture 1: Introduction / The Resource Description Framework RDF

Markus Krötzsch
Knowledge-Based Systems

TU Dresden, 15th Oct 2019

Introduction and Organisation

Course Tutors

Markus Krötzsch
Lectures

Larry González
Exercises

Organisation

Lectures
Tuesday, DS 3 (11:10–12:40), APB E005

Exercise Sessions (starting today)
Tuesday, DS 5 (14:50–16:20), APB E005

Web Page
`https://iccl.inf.tu-dresden.de/web/Knowledge_Graphs_(WS2019/20)`

Lecture Notes
Slides of current and past lectures will be online.

Modules
INF-B-510, INF-B-520, INF-BAS2, INF-VERT2, INF-BAS6, INF-VERT6, INF-E-3, INF-PM-FOR, MCL-KR, MCL-TCSL

Goals and Prerequisites

Goals

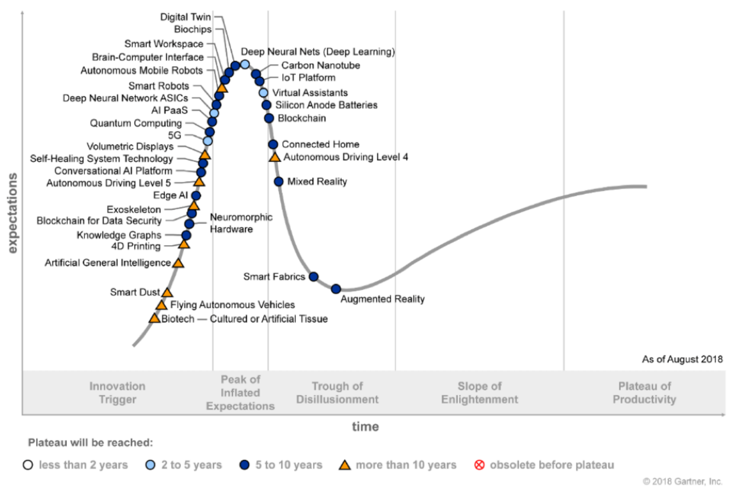
- Introduce basic notions of **graph-based knowledge representation(s)**
- Study important **graph data management approaches** (RDF, Property Graph) and **query languages**
- Learn about relevant **methods, tools, and datasets**
- Discuss aspects of **modelling and quality assurance**
- Get to know some **methods for analysing networks and graphs**

(Non-)Prerequisites

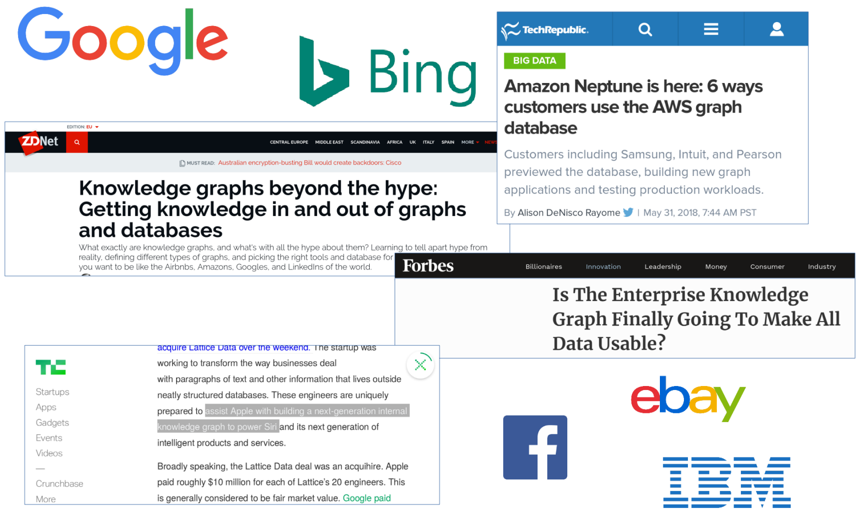
- No particular prior courses needed
- Basic programming skills are assumed; practical experience beyond basic courses will be helpful
- Interesting optional synergies: databases, machine learning, social networks, graph theory

Motivation

The Hype

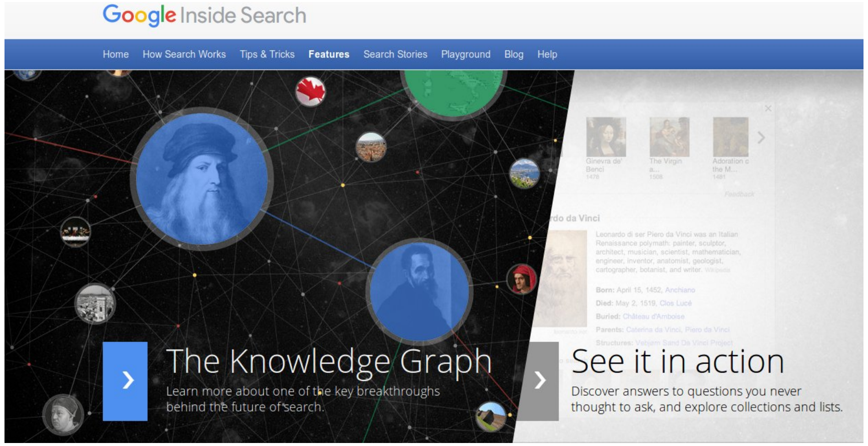


Knowledge Graphs Everywhere



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:

So what is a Knowledge Graph?

A first attempt at a definition:

A Knowledge Graph is a knowledge base that is a graph.

So what is a knowledge base?

- "A knowledge base is a technology used to store complex structured and unstructured information used by a computer system. [. . .] [It] represents facts about the world" – Wikipedia (14 Oct 2019, id 920346796)
- "A collection of knowledge expressed using some formal knowledge representation language." – Free Online Dictionary of Computing, 15 Oct 2018
- 1. a store of information or data that is available to draw on.
  2. the underlying set of facts, assumptions, and rules which a computer system has available to solve a problem.

  – Lexico (Oxford University Press/Dictionary.com), 14 Oct 2019

So what is a Knowledge Graph?

A first attempt at a definition:

A Knowledge Graph is a knowledge base that is a graph.

So what is a graph?

- "a collection of points and lines connecting some (possibly empty) subset of them" – Wolfram MathWorld, 14 Oct 2019
- "a collection of vertices and edges that join pairs of vertices" – Merriam-Webster, 14 Oct 2019
- "a structure amounting to a set of objects in which some pairs of the objects are in some sense 'related'." – Wikipedia (14 Oct 2019, id 920437291)

(we'll have more to say about mathematical graphs later)

So what is a Knowledge Graph?

A first attempt at a definition:

A Knowledge Graph is a knowledge base that is a graph.

In summary:

- a collection of facts, rules, or other forms of knowledge
- that express some kind of relationships or connections

⇝ a paradigm rather than a specific class of things

What is special about Knowledge Graphs?

A second attempt at a definition:

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:

- 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 sub-ordinary

Primarily not knowledge graphs:

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

Lecture Outline

- **Resource Description Framework (RDF)**
  Underlying graph model; URIs; syntax
- **SPARQL**
  Query features; syntax and semantics; expressive power and complexlity
- **Property graph**
  Underlying graph model; syntax and semantics of Cypher
- **Wikidata**
  Data model; applications; aspects of modelling; query answering
- **Ontologies and rules**
  Datalog; negation; existential rules; ontological models; OWL
- **RDF constraint languages**
  SHACL & ShEX; syntax and semantics; complexity and implementation
- **Network analysis**
  Centrality measures, PageRank, community detection

Graphs in Computer Science and Mathematics

What is a graph?

Definition 1.1: A simple undirected graph G consists of a set V of vertices and a set E of edges, where each edge is a set of two vertices. Two vertices $v_1, v_2 \in V$ are adjacent (in G) if there is an edge $\{v_1, v_2\} \in E$.

Vertices are sometimes also called nodes; undirected edges are sometimes also called arcs.

Unless otherwise noted, we assume all graphs to be finite.

Discrete mathematics considers a variety of other kinds of "graphs":

- Directed or undirected
- Simple graph or multi-graph
- Possibly labelled edges or vertices
- Possibly with self-loops
- Possibly with higher arity edges (hypergraphs)

Directed and other graphs

Definition 1.2: A simple directed graph (a.k.a. simple digraph) G consists of a set V of vertices and a set $E \subseteq V \times V$ of (directed) edges from a source vertex to a target vertex.

Other terms are similar to undirected graphs; directed edges are also known as arrows and are often denoted as such, e.g., $v_1 \xrightarrow{e_1} v_2$.

Definition 1.3: The following generalisations apply to directed and to undirected graphs.

- A graph with self-loops is a graph extended with the option of having edges that relate a vertex to itself.
- A multi-graph is a graph that may have multiple edges with the same vertices (in the same direction).
- An edge-labelled graph is a graph that has an additional labelling function $\lambda : E \to L$ that maps each edge in E to an element a set of labels L (similarly for vertex-labelled graphs).

Other basic notions

Definition 1.4: An edge are said to be incidental to the vertices it connects. The degree of a vertex is the number of edges that are incidental to it. In a digraph, the in-degree of a vertex is the number of edges pointing towards it; analogously for out-degree.

Definition 1.5: A directed path in a digraph is a sequence of consecutive edges $v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} v_n$. An undirected path is a sequence of edges that may point either way (or that are simply undirected).

A simple path (directed or undirected) is a path without repeated vertices other than possibly the first and last node.

Definition 1.6: Two vertices are connected if there is an undirected path from one to the other. A graph is connected if any pair of two distinct vertices is connected. A digraph is strongly connected if there is a directed path from any vertex to any other vertex (hence: one directed path in either direction).

Representing graphs (1)

There are several obvious ways of representing graphs in computer science.

Definition 1.7: The adjacency matrix of a graph $G = \langle V, E\rangle$ is the boolean $|V| \times |V|$ matrix that contains, at any coordinate $\langle v_1, v_2\rangle$, the value 1 if there is an edge connecting v_1 and v_2.

Notes:

- Adjacency matrices for undirected graphs are symmetric.
- Loops (if allowed) show up as 1 in the diagonal.
- The matrix could be adapted to multi-graphs by storing the numbers of edges.
- The matrix could be adapted to labelled simple graphs by storing the labels.

Representing graphs (2)

There are several obvious ways of representing graphs in computer science.

Definition 1.8: The adjacency list of a graph $G = \langle V, E\rangle$ is the list of all of its edges.

Notes:

- We can write edges as pairs (order is irrelevant for undirected graphs).
- Loops (if allowed) show up as edges with repeated vertices.
- The list could be adapted to multi-graphs by adding the number of edges to each line, or by allowing repeated lines.
- The matrix could be adapted to labelled graphs by adding labels to each line (for multi-graph: repeat lines rather than also storing number).
- The list does not encode V: vertices without edges are missing (might be listed separately if relevant to application)

Which graph representation to pick?

Each representation has its pros and cons:

- Matrix: space efficient for dense graphs (1 bit per edge); can be processed with matrix operations (highly parallel); space inefficient for sparse graphs; not natural for labelled multi-graphs
- List: space efficient for sparse graphs; easy to use for labelled multi-graphs; harder to process (esp. if edge order can be random); not space efficient for dense graphs

Note: knowledge graphs are typically sparse and labelled, but parallel processing still makes matrices attractive in some applications.

There are also other options.

Example 1.9: We could also encode the adjacency matrix by giving, for each row, a list of all vertices whose column is set to 1. This is equivalent to ordering edges by first vertex and combining them into a single line.

Encoding Graphs in RDF

Encoding Graphs

We have seen that graphs can be encoded in several ways:

- Adjacency matrix (and variants)
- Adjacency list (and variants)
- Other derived representations

This is enough to store and manipulate graphs in software,
but it is not enough to exchange graphs across applications.

Open questions:

- What kind(s) of graph do we want to exchange?
- How are vertices given (numbers? strings? specific ids? . . .)?
- Are edge labels supported and what are they?
- Can the graph include values of data types (integer? float? string? times? . . .)?
- How exactly are these things encoded in bytes in a file?

The Resource Description Framework

RDF is a W3C[1] standard for exchanging graphs

- First proposed in 1999
- Updated in 2004 (RDF 1.0) and in 2014 (RDF 1.1)
- Originally built for Web data exchange
- Meanwhile used in many graph database applications
- Supported by many other W3C standards (RDFa, SPARQL, OWL, SHACL, . . .)

In this course: focus on graph representation features of RDF 1.1

W3C creates open standards: patent-free & freely accessible

- Gentle RDF 1.1 introduction: `https://www.w3.org/TR/rdf11-primer/`
- Specification of graph model: `https://www.w3.org/TR/rdf11-concepts/`
- Specific file formats are defined in other documents, linked from those

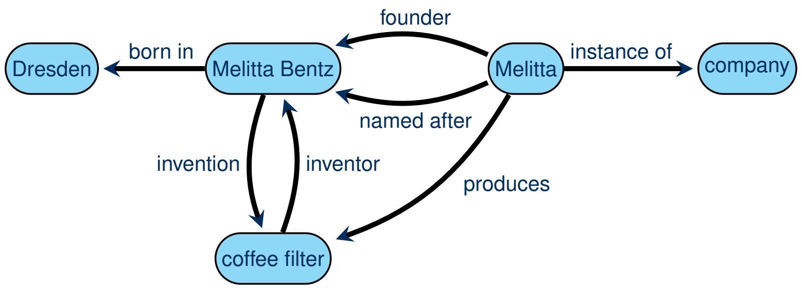
[1] World Wide Web Consortium

Graphs in RDF

RDF allows us to specify graphs that are:

- directed (edges have a source and a target)
- edge-labelled (edges have one label)
- a restricted form of multi-graphs (multiple edges can exist between same vertices, but only if they have different labels)

Example of such a graph:



Identifiers in RDF

How should we refer to vertices? What kind of labels are allowed?

Definition 1.10: A Uniform Resource Identifier (URI) is a sequence (string) of a subset of ASCII characters as defined in RFC 3986 (link). Every (absolute) URI consists of a string that defines a scheme, followed by a colon (:) and another sequence of characters specifying an authority, path, query, and fragment, where all parts other than the path are optional.

A International Resource Identifier (IRI) is a generalised form of URI that allows for an expanded range of Unicode glyphs in part of its syntax.

Example 1.11: URLs are a well-known form of URI, for example:

$$\underbrace{\texttt{https}}_{\text{scheme}}:\underbrace{\texttt{//example.org}}_{\text{authority}}\underbrace{\texttt{/some/page}}_{\text{path}}\underbrace{\texttt{?get=something\&lang=en}}_{\text{query}}\underbrace{\texttt{\#results}}_{\text{fragment}}$$

Convention: We ignore the differences between URIs and IRIs in this course.
(The lecturer will often say “URI” when he should say “IRI.” Please ignore.)

URIs vs. URLs

The widely used term Uniform Resource Locator (URL) is an informal way to refer to URIs that specify the location of a digital document. Not all URIs support this.

Example 1.12: Many URI schemes have been defined. Examples include

- `http`, `https`, `ftp` for transferring data using various protocols
- `mailto` for emails (=path)
- `irc` for specifying IRC channels
- `file` for referring to locations on some file system
- `urn` for naming resources without defining a protocol or resolution mechanism; used, e.g., for ISBNs
- ...

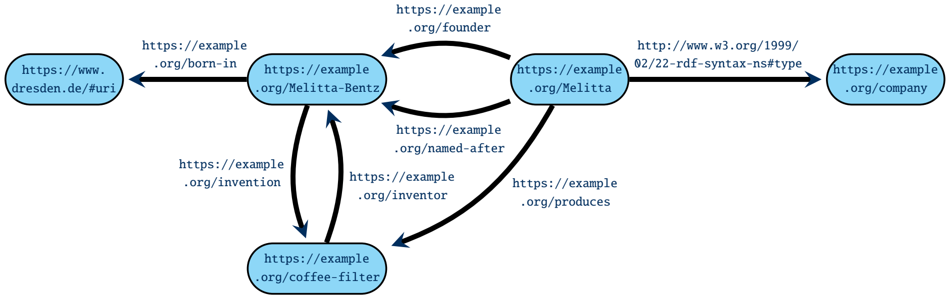
Registered schemes usually provide some additional syntax requirements, access protocols, and resolution mechanisms, and they often relate to a registration procedure based on some authority.

IRIs in RDF

RDF uses IRIs in two ways:

- IRIs define resources that appear as vertices in the graph
- IRIs are used as edge labels

Example of such a graph:



Note: It is not always obvious what an IRI is supposed to refer to, and many IRIs may refer to the same thing – we cannot assume that all RDF data in the world is integrated.

Summary

Knowledge Graphs are a data management concept of practical importance

Mathematics studies various types of graphs, which can be represented by several common data structures

RDF is a W3C standard for describing directed, edge-labelled graphs in an interoperable way

It identifies vertices and edge-types using IRIs

What's next?

- More RDF features: data types and blank nodes
- Formats for exchanging RDF data
- Modelling in RDF