Lecture 2

Knowledge Graphs

COMP 474/6741, Winter 2021

René Witte
Department of Computer Science
and Software Engineering
Concordia University

Outline

René Witte

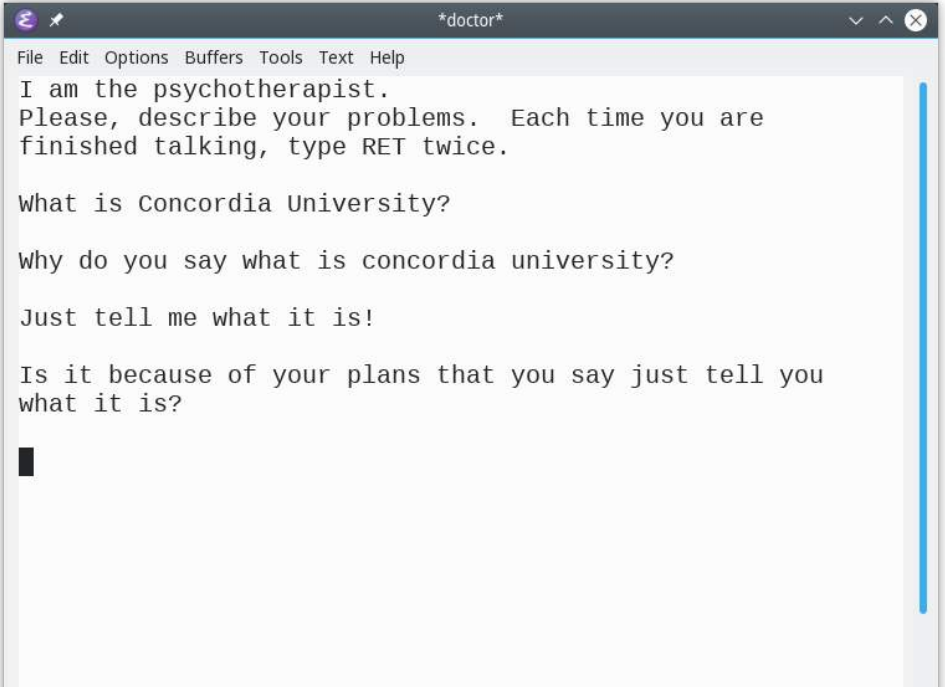
Slides Credit

- Includes slides from Jay Pujara & Sameer Singh, *Mining Knowledge Graphs from Text*, https://kgtutorial.github.io/
- Includes slides by Ivan Herman, W3C [Her]

Outline

René Witte

Eliza

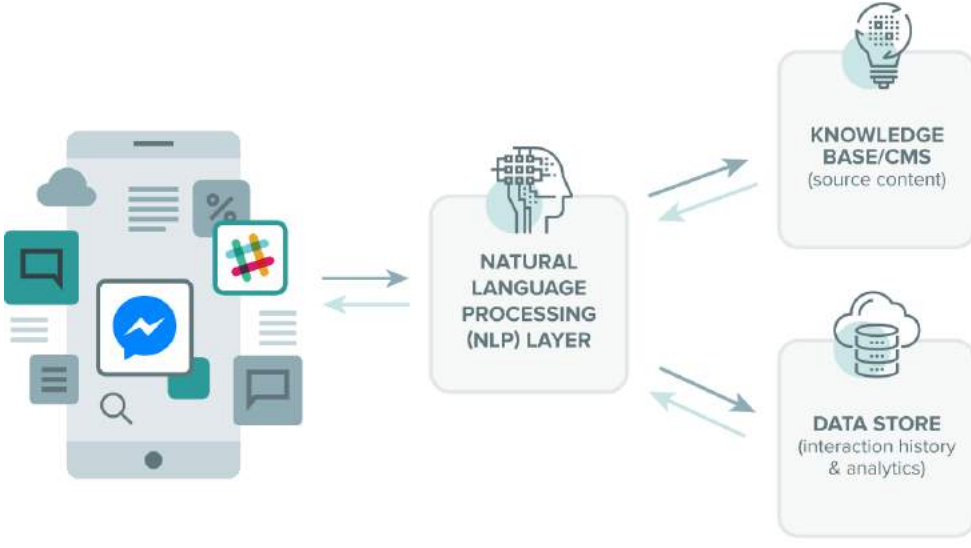


René Witte

Google Assistant

René Witte

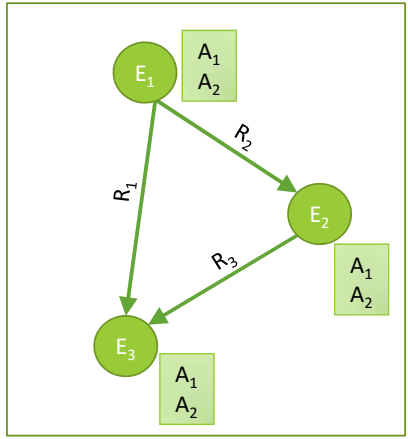
Generic Assistant Architecture



René Witte

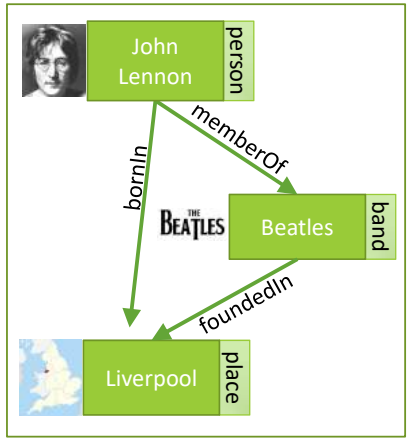
What is a knowledge graph?

- Knowledge in graph form!
- Captures entities, attributes, and relationships
- Nodes are entities
- Nodes are labeled with attributes (e.g., types)
- Typed edges between two nodes capture a relationship between entities



Example knowledge graph

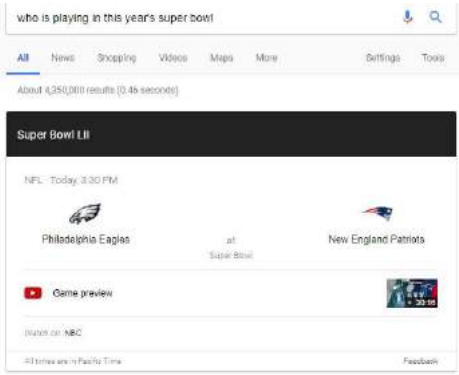
- Knowledge in graph form!
- Captures entities, attributes, and relationships
- Nodes are entities
- Nodes are labeled with attributes (e.g., types)
- Typed edges between two nodes capture a relationship between entities



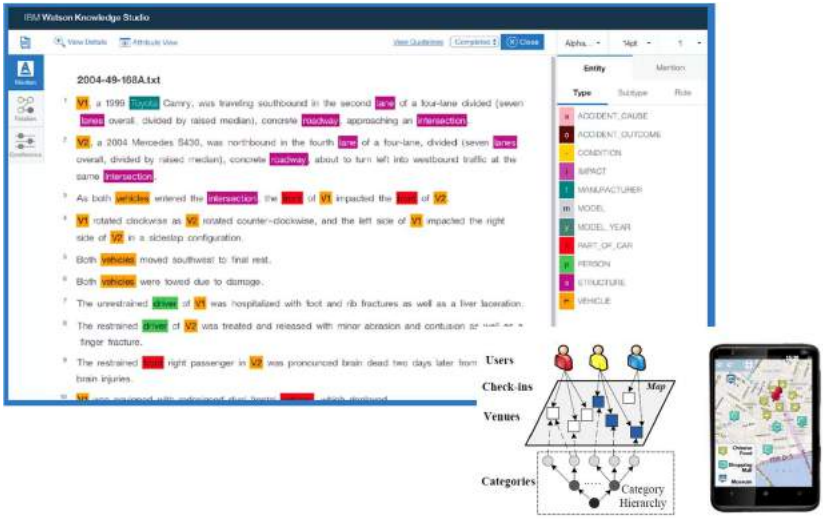
Why knowledge graphs?

- Humans:
  - Combat information overload
  - Explore via intuitive structure
  - Tool for supporting knowledge-driven tasks
- AIs:
  - Key ingredient for many AI tasks
  - Bridge from data to human semantics
  - Use decades of work on graph analysis

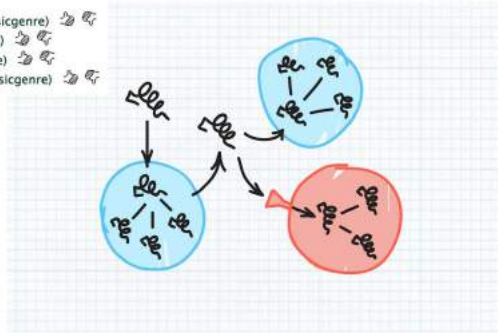
Applications 1: QA/Agents



Applications 2: Decision Support

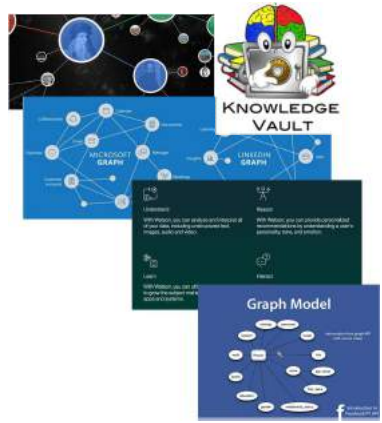


Applications 3: Fueling Discovery



Knowledge Graphs & Industry

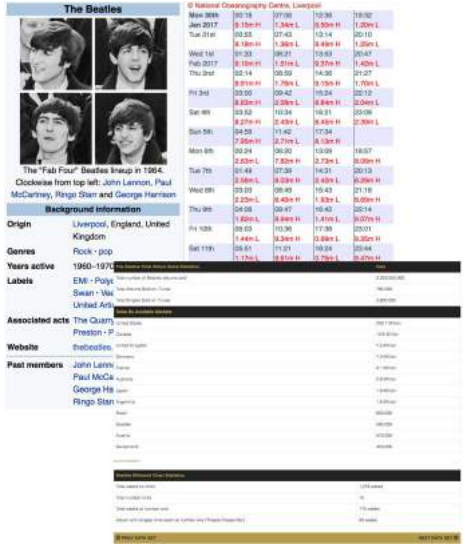
- Google Knowledge Graph
  - Google Knowledge Vault
- Amazon Product Graph
- Facebook Graph API
- IBM Watson
- Microsoft Satori
  - Project Hanover/Literome
- LinkedIn Knowledge Graph
- Yandex Object Answer
- Diffbot, GraphIQ, Maana, ParseHub, Reactor Labs, SpazioDati



René Witte

Where do knowledge graphs come from?

- Structured Text
  - Wikipedia Infoboxes, tables, databases, social nets



Where do knowledge graphs come from?

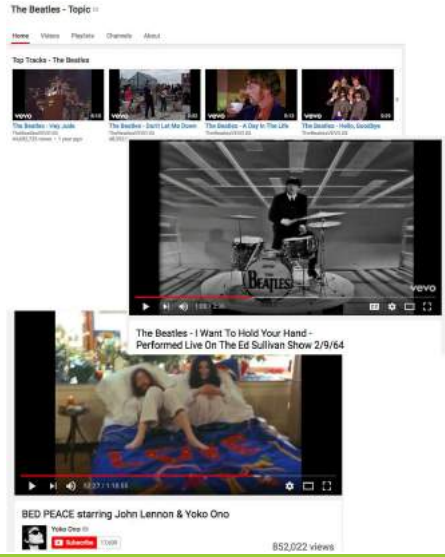
- Structured Text
  - Wikipedia Infoboxes, tables, databases, social nets
- Unstructured Text
  - WWW, news, social media, reference articles

Where do knowledge graphs come from?

- Structured Text
  - Wikipedia Infoboxes, tables, databases, social nets
- Unstructured Text
  - WWW, news, social media, reference articles
- Images

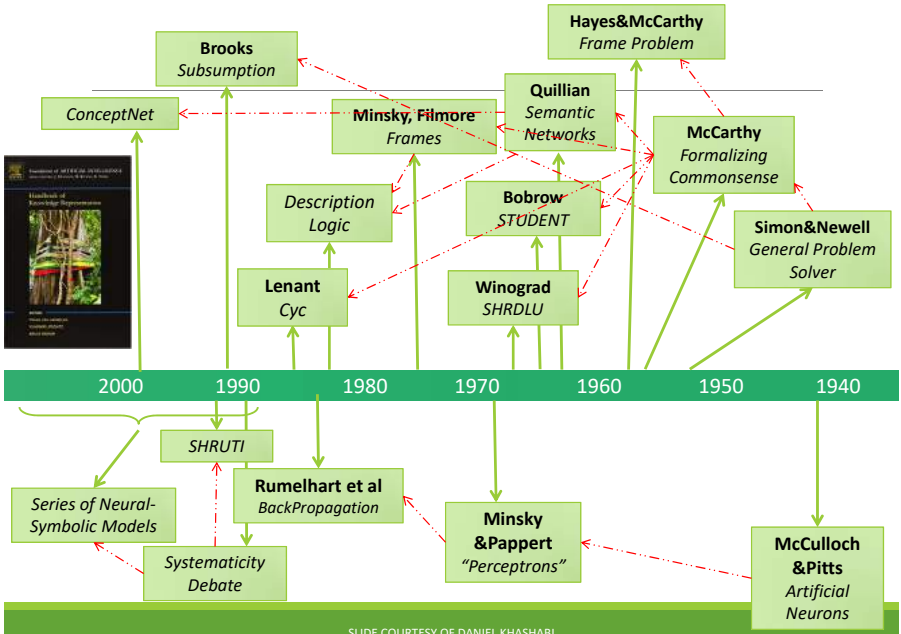
Where do knowledge graphs come from?

- Structured Text
  - Wikipedia Infoboxes, tables, databases, social nets
- Unstructured Text
  - WWW, news, social media, reference articles
- Images
- Video
  - YouTube, video feeds



Outline

History of Knowledge Representation (KR)



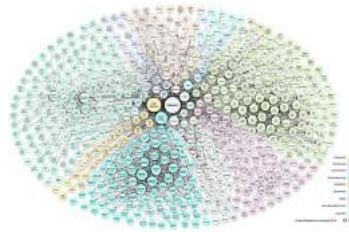
René Witte

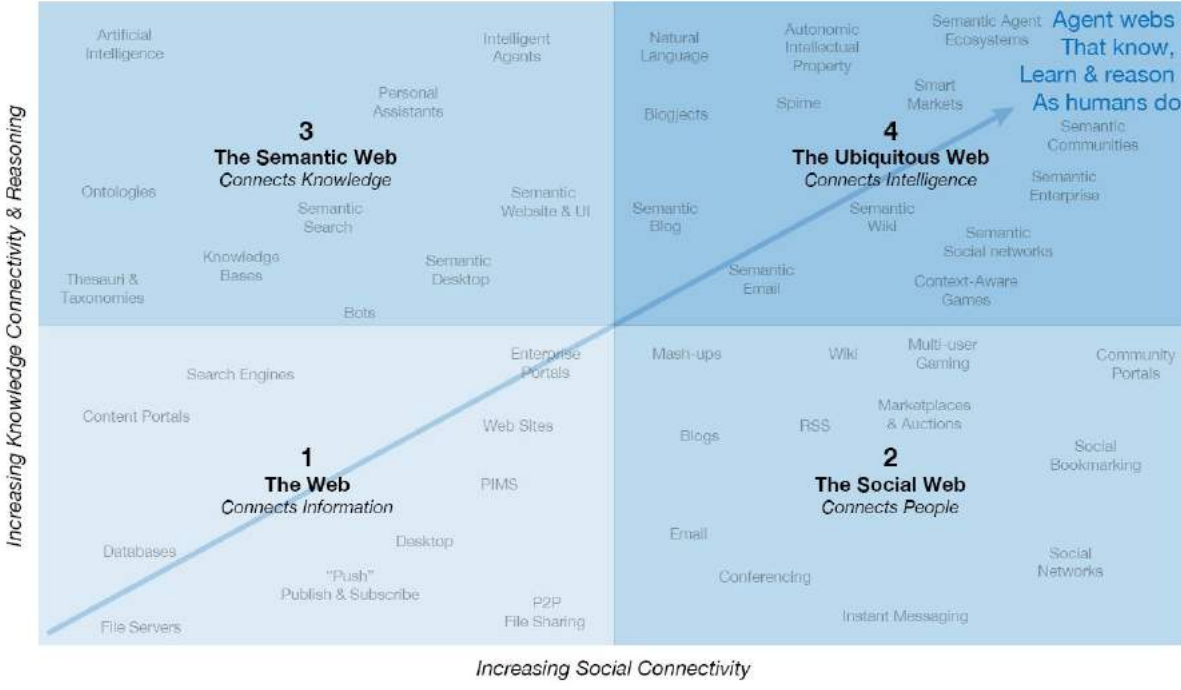
Knowledge Representation

- Decades of research into knowledge representation
- Most knowledge graph implementations use RDF triples
  - <rdf:subject, rdf:predicate, rdf:object> : r(s,p,o)
  - Temporal scoping, reification, and skolemization...
- ABox (assertions) versus TBox (terminology)
- Common ontological primitives
  - rdfs:domain, rdfs:range, rdf:type, rdfs:subClassOf, rdfs:subPropertyOf, ...
  - owl:inverseOf, owl:TransitiveProperty, owl:FunctionalProperty, ...

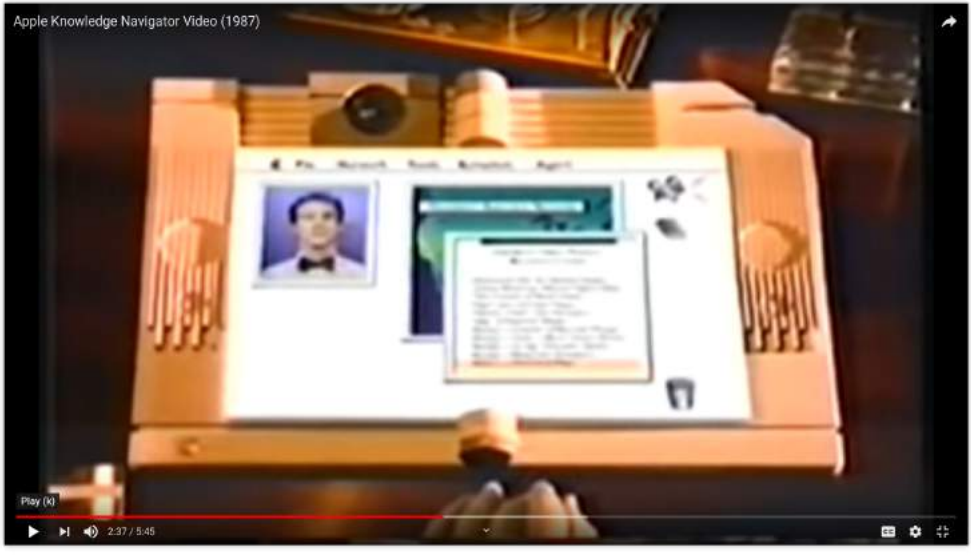
Semantic Web

- Standards for defining and exchanging knowledge
  - RDF, RDFa, JSON-LD, schema.org
  - RDFS, OWL, SKOS, FOAF
- Annotated data provide critical resource for automation
- Major weakness: annotate everything?




Increasing Knowledge Connectivity & Reasoning
Increasing Social Connectivity
Artificial Intelligence
Intelligent Agents
Personal Assistants
3
The Semantic Web
Connects Knowledge
Ontologies
Semantic Search
Semantic Website & UI
Knowledge Bases
Semantic Desktop
Thesauri & Taxonomies
Bots
Natural Language
Autonomic Intellectual Property
Semantic Agent Ecosystems
Agent webs That know, Learn & reason As humans do
Smart Markets
Spime
Blogjects
Semantic Communities
4
The Ubiquitous Web
Connects Intelligence
Semantic Enterprise
Semantic Blog
Semantic Wiki
Semantic Social networks
Semantic Email
Context-Aware Games
Enterprise Portals
Search Engines
Content Portals
Web Sites
1
The Web
Connects Information
PIMS
Desktop
Databases
"Push" Publish & Subscribe
P2P File Sharing
File Servers
Mash-ups
Wiki
Multi-user Gaming
Community Portals
Marketplaces & Auctions
Blogs
RSS
Social Bookmarking
2
The Social Web
Connects People
Email
Social Networks
Conferencing
Instant Messaging

Apple's "Knowledge Navigator" Vision (1987)



https://www.youtube.com/watch?v=umJsITGzXd0

René Witte

Today

From 1950–2020...

- Concepts have been around for a long time (Semantic Networks, Frames, Description Logic, ...)

1980s/90s

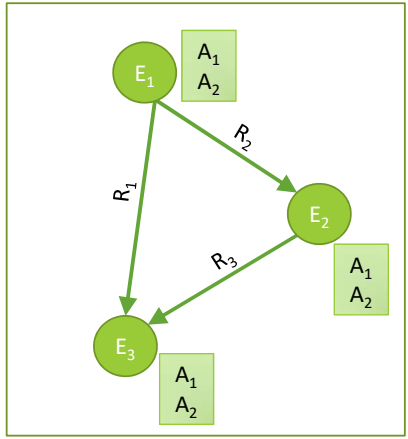
- AI/IS systems suffer from the *Knowledge Acquisition Bottleneck*
- One of the reasons for the AI Winter at that time

Technology

- Open standards, based on W3C recommendations, e.g., RDF
- Proprietary products, e.g., Neo4J or Oracle Spatial and Graph
- We now have substantial knowledge bases available, both proprietary (e.g., Facebook Graph Search, Google Knowledge Graph) and open access (e.g., Wikidata, DBpedia, YAGO)

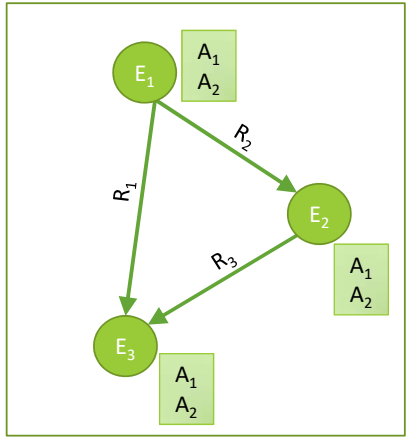
What is a knowledge graph?

- Knowledge in graph form!
- Captures entities, attributes, and relationships
- Nodes are entities
- Nodes are labeled with attributes (e.g., types)
- Typed edges between two nodes capture a relationship between entities



Basic problems

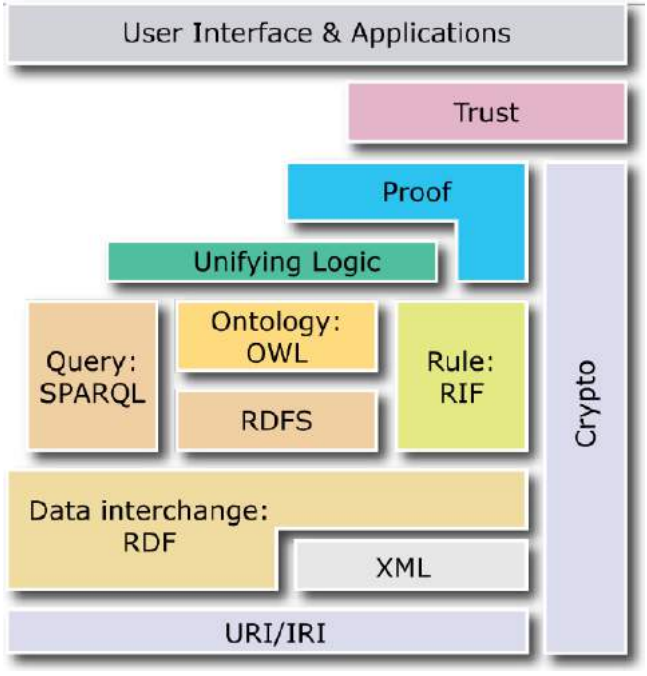
- **Who** are the entities (nodes) in the graph?
- **What** are their attributes and types (labels)?
- **How** are they related (edges)?



Outline

René Witte

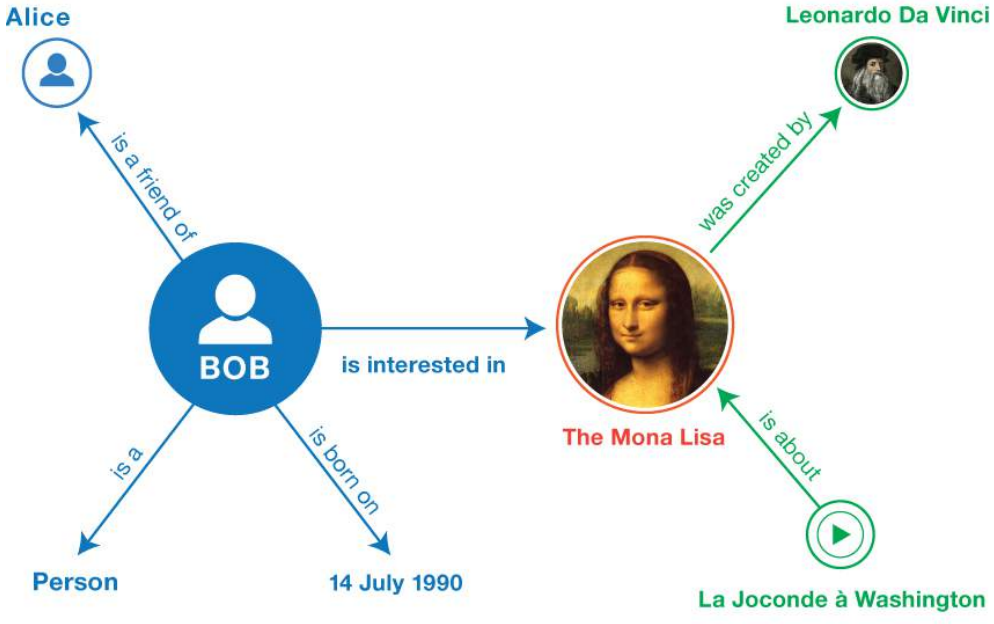
The W3C “Layer Cake”



René Witte
Concordia University

Knowledge as Graphs

René Witte



→ **Worksheet #1: Tasks 1 & 2**

https://www.w3.org/TR/rdf11-primer/

Triples

René Witte

Representation of Knowledge Graphs

In a system, we represent graphs in form of triples:

```
<subject> <predicate> <object>
```

(The *predicate* is sometimes called *property*.)

Examples

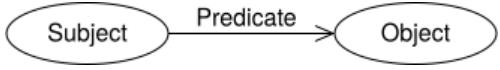
```
<Bob> <is a> <person>.
<Bob> <is a friend of> <Alice>.
<Bob> <is born on> <the 14th of July 1990>.
<Bob> <is interested in> <the Mona Lisa>.
<the Mona Lisa> <was created by> <Leonardo da Vinci>.
```

→ **Worksheet #1: Tasks 3 & 4**

Graphs vs. Triples

René Witte

<subject> <predicate> <object>



RDF Triples

The Resource Description Framework (RDF)

W3C (World Wide Web Consortium) standard (“recommendation”)

- first public draft 1997
- RDF 1.0 in 1999; revised in 2004
- RDF 1.1 in 2014 (current version)

Family of standards: RDF, RDFS, RDFa, Turtle, N3, SPARQL, . . .

RDF Triples

René Witte

Format of triples

In RDF,

- Subject and predicate must be URIs (IRIs)
- Object can be IRI or literal

Examples

```
<http://www.wikidata.org/entity/Q12418>
    <http://purl.org/dc/terms/title>
    "Mona Lisa" .

<http://www.wikidata.org/entity/Q12418>
    <http://purl.org/dc/terms/creator>
    <http://dbpedia.org/resource/Leonardo_da_Vinci> .
```

→ **Worksheet #1: Tasks 5 & 6**

RDF Literals

"Mona Lisa"
In this triple

```
<http://www.wikidata.org/entity/Q12418>
    <http://purl.org/dc/terms/title> "Mona Lisa" .
```

"Mona Lisa" is a string literal

Things to know about literals

- Literals have a datatype, e.g., `string` or `int`
- Strings can have a language tag, e.g.,

  ```
  "Leonardo da Vinci"@en
  "Léonard de Vinci"@fr
  ```

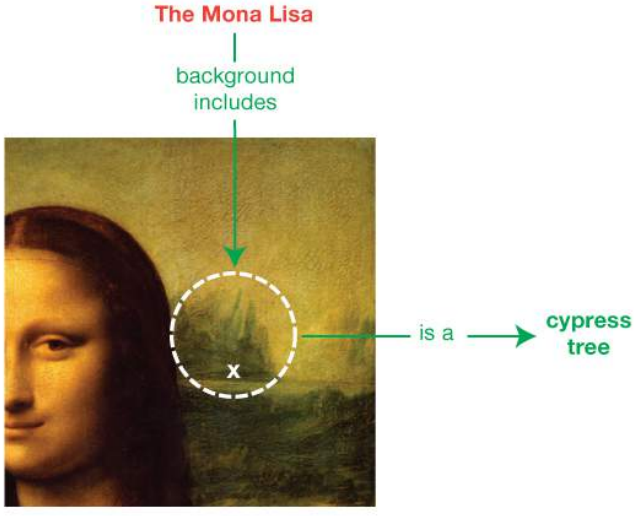
- Strings are often used to provide human-readable labels
  "Hey, how did you like the movie Q168154?"
- For strings only, datatype can be omitted:
  `"Mona Lisa"` is equivalent to `"Mona Lisa"^^xsd:string`
- Again, literals can only appear in the object position of a triple `<s> <p> <o>`

All the details about datatypes: https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/#section-Datatypes

René Witte
Concordia University

Blank Nodes

René Witte




```
<http://dbpedia.org/resource/Mona_Lisa> <lio:shows> _:x .
_:x a <http://dbpedia.org/resource/Cypress> .
```

DBpedia

http://dbpedia.org/resource/Leonardo_da_Vinci

DBpedia URIs

Make sure you use the correct URI!

- `http://dbpedia.org/resource/...` is the canonical URI
- The DBpedia server returns either
  - `http://dbpedia.org/page/...` (HTML data, for a human)
  - `http://dbpedia.org/data/...` (RDF data, for an AI)

→ Worksheet #1: Task 7

René Witte

Namespaces

René Witte

Shortening URIs

Instead of always writing full URIs (IRIs), we can split them into a prefix and suffix, e.g.: `<http://dbpedia.org/resource/Leonardo_da_Vinci>`

- We define a prefix dbpedia:
  `PREFIX dbpedia: <http://dbpedia.org/resource/>`
- and now we can simple write:
  `dbpedia:Leonardo_da_Vinci`
- Note: angle brackets <> only for full IRIs

→ reduces dataset sizes, easier to read

Conventions

Commonly used URLs use the same namespace prefix

- E.g., FOAF (friend-of-a-friend):
  `PREFIX foaf: <http://xmlns.com/foaf/0.1/>`
- Lookup a prefix at https://prefix.cc/

→ Worksheet #1: Tasks 8 & 9

Serialization

René Witte

Formats

There is no single format `.rdf` (like `.xml`), commonly used are:

RDF/XML for data exchange (somewhat deprecated)

RDFa for embedding RDF into web pages

N-Triples (N3) for streaming RDF data and bulk dataset up-/download

Turtle for human-readable files

JSON-LD for web applications

plus some variations/extensions.

N-Triples

So far, we've mostly used the N-Triples format:

```
<http://www.wikidata.org/entity/Q12418> ↩
    <http://purl.org/dc/terms/title> "Mona Lisa" .
```

each line in a file is one triple, full IRIs only (no namespace prefixes) and ended by a period `'.'`

Turtle

René Witte

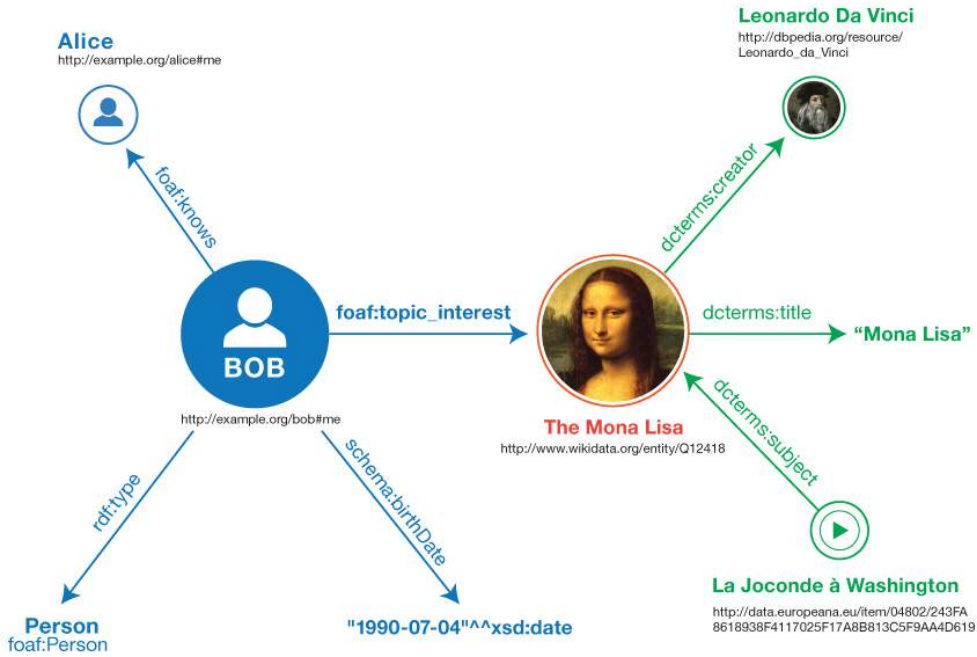

```
BASE <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
PREFIX schema: <http://schema.org/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX wd: <http://www.wikidata.org/entity/>

<bob#me>
   a foaf:Person ;
   foaf:knows <alice#me> ;
   schema:birthDate "1990-07-04"^^xsd:date ;
   foaf:topic_interest wd:Q12418 .

wd:Q12418
   dcterms:title "Mona␣Lisa" ;
   dcterms:creator <http://dbpedia.org/resource/Leonardo_da_Vinci> .

<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619>
   dcterms:subject wd:Q12418 .
```

Graph corresponding to the Turtle example



https://www.w3.org/TR/rdf11-primer/

René Witte

RDF in programming practice

- For example, using Python+RDFLib:
  - a “Graph” object is created
  - the RDF file is parsed and results stored in the Graph
  - the Graph offers methods to retrieve (or add):
    - triples
    - (property,object) pairs for a specific subject
    - (subject,property) pairs for specific object
    - etc.
  - the rest is conventional programming…
- Similar tools exist in Java, PHP, etc.

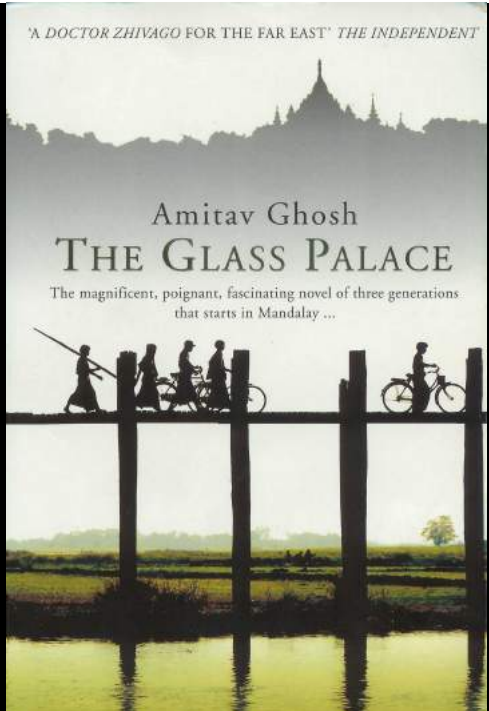
René Witte
Concordia University

Python example using RDFLib

```
# create a graph from a file
graph = rdflib.Graph()
graph.parse("filename.rdf", format="rdfxml")
# take subject with a known URI
subject = rdflib.URIRef("URI_of_Subject")
# process all properties and objects for this subject
for (s,p,o) in graph.triples((subject,None,None)) :
  do_something(p,o)
```

Outline

Let's start with a Book...



René Witte
Concordia University

A simplified bookstore data (dataset “A”)

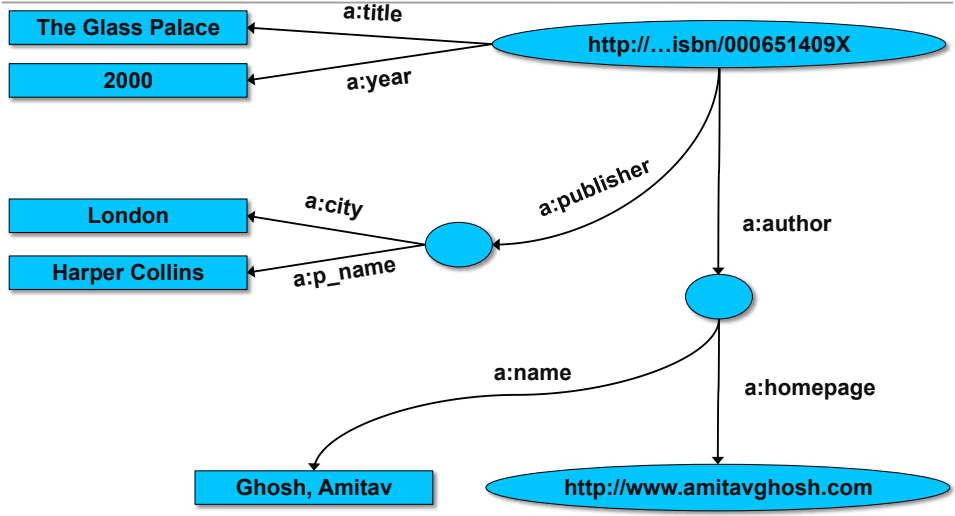
ISBN	Author	Title	Publisher	Year
0006511409X	id_xyz	The Glass Palace	id_qpr	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

ID	Publisher’s name	City
id_qpr	Harper Collins	London

René Witte

1st: export your data as a set of *relations*



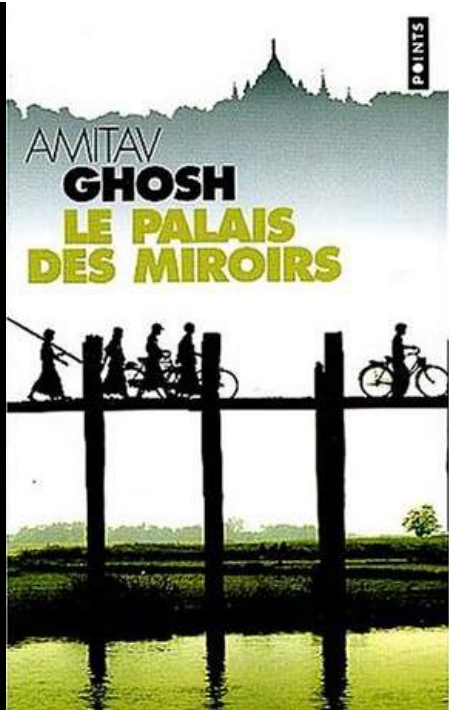
René Witte

Some notes on the exporting the data

- Relations form a graph
  - the nodes refer to the “real” data or contain some literal
  - how the graph is represented in machine is immaterial for now

René Witte
Concordia University

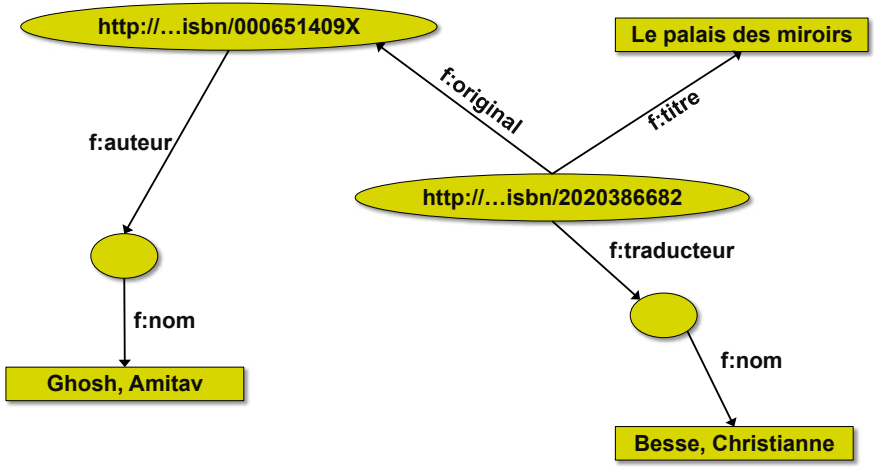
Now the same book in French. . .



Another bookstore data (dataset “F”)

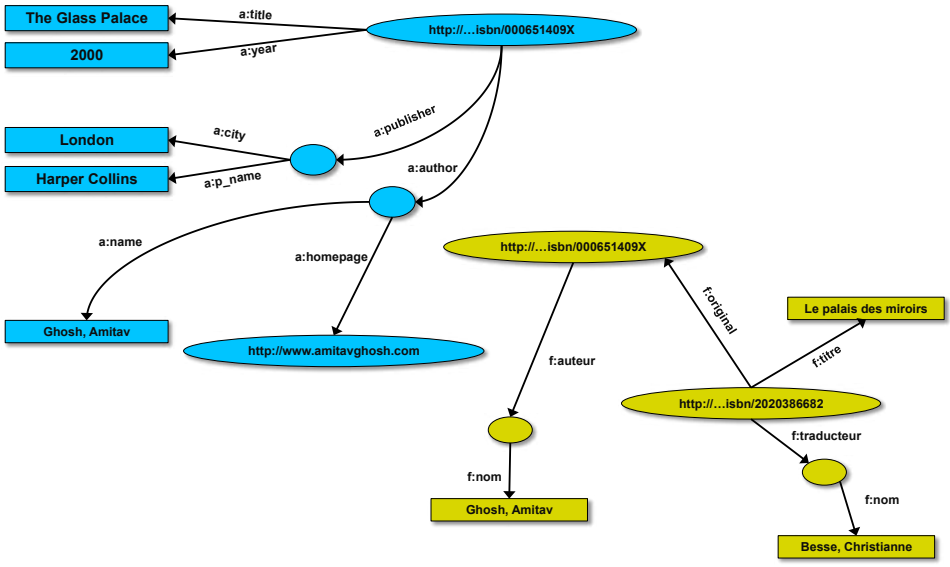
	A	B	C	D
1	**ID**	**Titre**	**Traducteur**	**Original**
2	ISBN 2020286682	Le Palais des Miroirs	$A12$	ISBN 0-00-6511409-X
3				
4				
5				
6	**ID**	**Auteur**		
7	ISBN 0-00-6511409-X	$A11$		
8				
9				
10	**Nom**			
11	Ghosh, Amitav			
12	Besse, Christianne			

2nd: export your second set of data



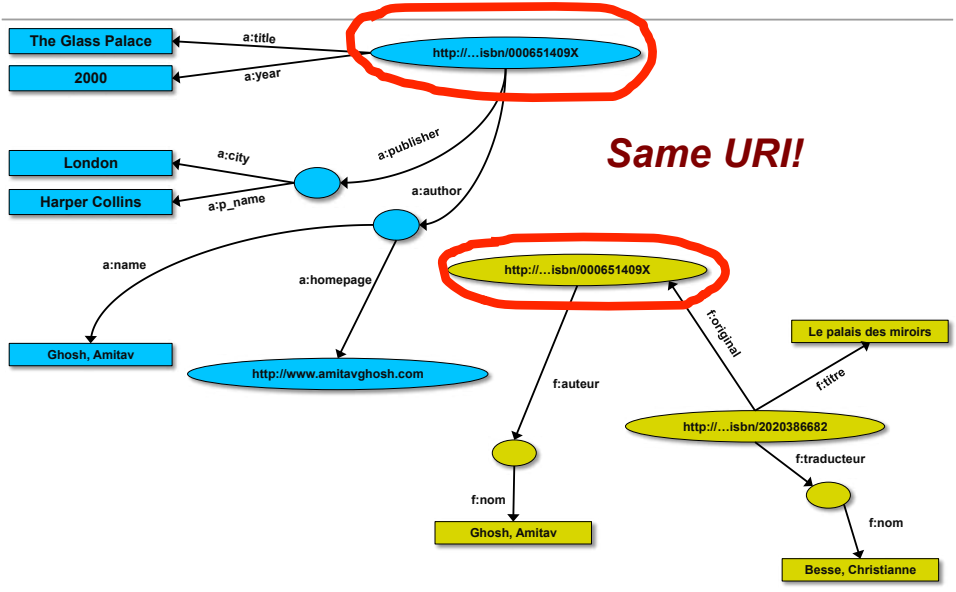
René Witte
Concordia University

3rd: start merging your data



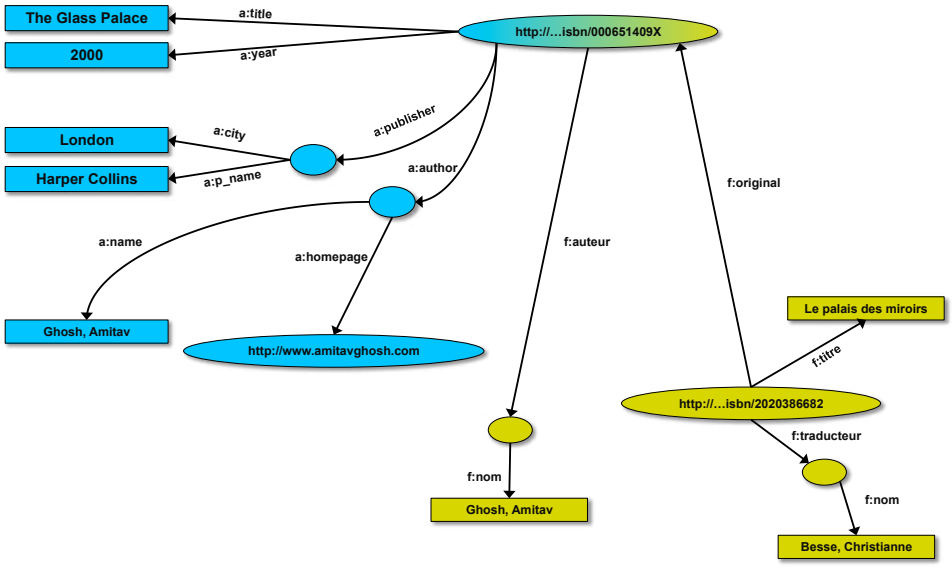
René Witte

3rd: start merging your data (cont)



René Witte

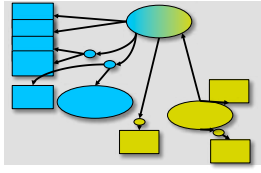
3rd: start merging your data



René Witte

Start making queries…

- User of data “F” can now ask queries like:
  - “give me the title of the original”
    - well, … « donnes-moi le titre de l’original »
- This information is not in the dataset “F”…
- …but can be retrieved by merging with dataset “A”!

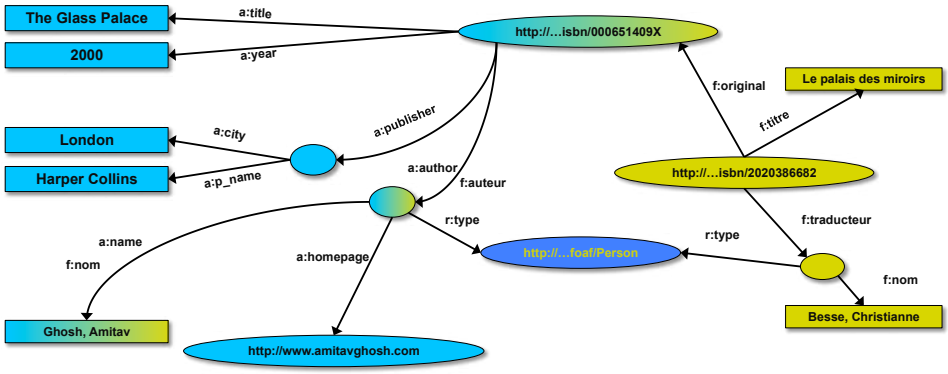


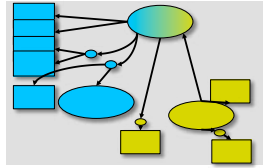
However, more can be achieved…

- We “feel” that a:author and f:auteur should be the same
- But an automatic merge doest not know that!
- Let us add some extra information to the merged data:
  - a:author same as f:auteur
  - both identify a “Person”
  - a term that a community may have already defined:
    - a “Person” is uniquely identified by his/her name and, say, homepage
    - it can be used as a “category” for certain type of resources

René Witte
Concordia University

3rd revisited: use the extra knowledge

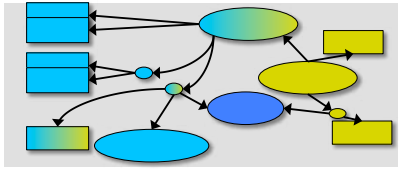




René Witte

Start making richer queries!

- User of dataset “F” can now query:
  - “donnes-moi la page d’accueil de l’auteur de l’original”
    - well… “give me the home page of the original’s ‘auteur’”
- The information is not in datasets “F” or “A”…
- …but was made available by:
  - merging datasets “A” and datasets “F”
  - adding three simple extra statements as an extra “glue”



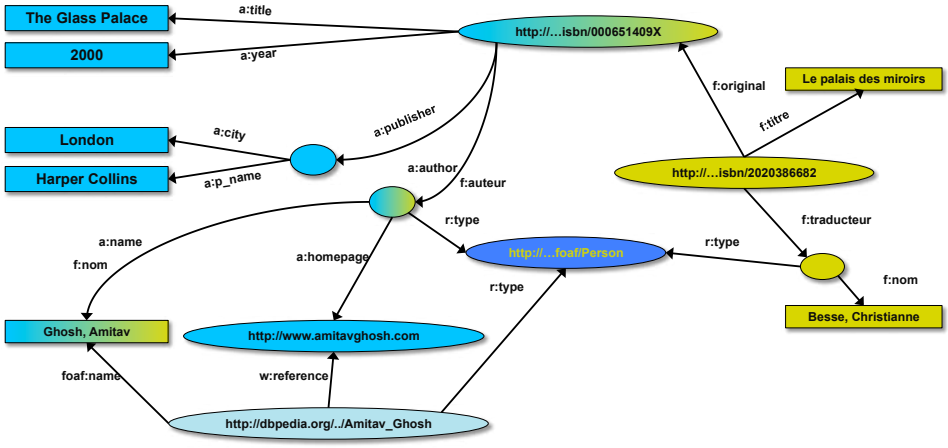
René Witte
Concordia University

Combine with different datasets

- Using, e.g., the “Person”, the dataset can be combined with other sources
- For example, data in Wikipedia can be extracted using dedicated tools
  - e.g., the “dbpedia” project can extract the “infobox” information from Wikipedia already…

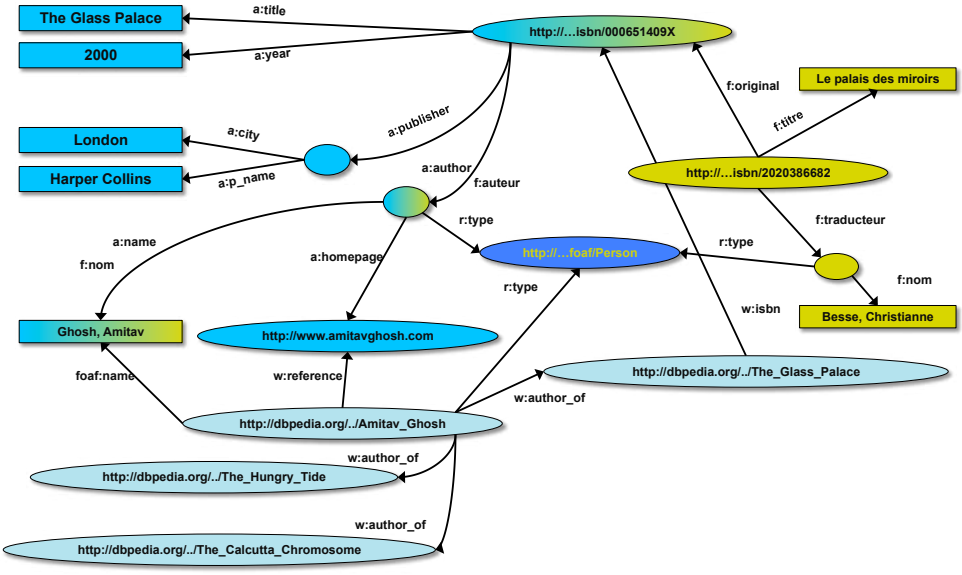
René Witte
Concordia University

Merge with Wikipedia data



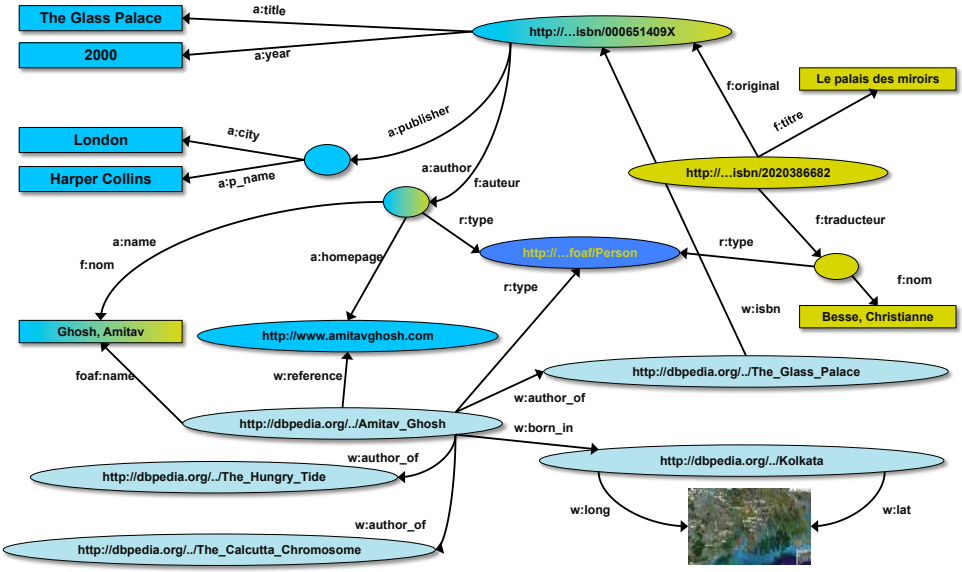
René Witte

Merge with Wikipedia data



René Witte

Merge with Wikipedia data



René Witte

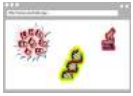
Is that surprising?

- It may look like it but, in fact, it should not be…
- What happened via automatic means is done every day by Web users!
- The difference: a bit of extra rigour so that machines could do this, too

→ Worksheet #1: Task 10

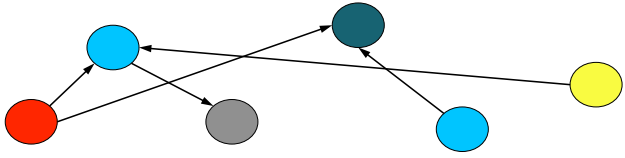
Outline

What did we do?



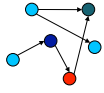
Applications

Manipulate
Query
…



Data represented in abstract format

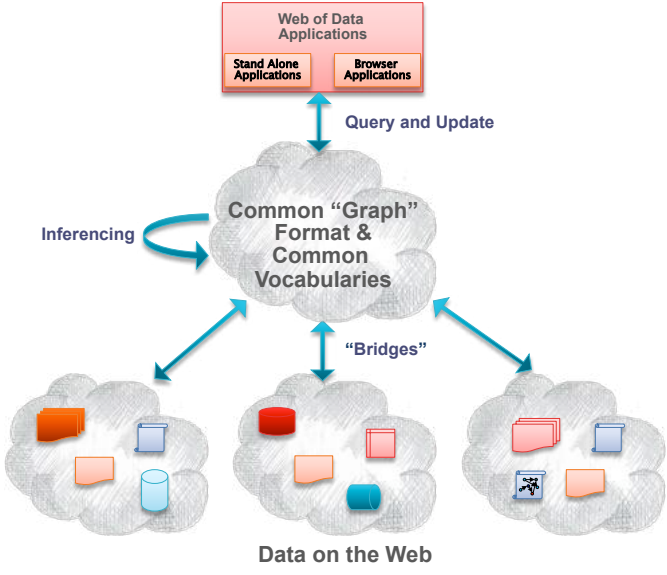
Map,
Expose,
…



Data in various formats

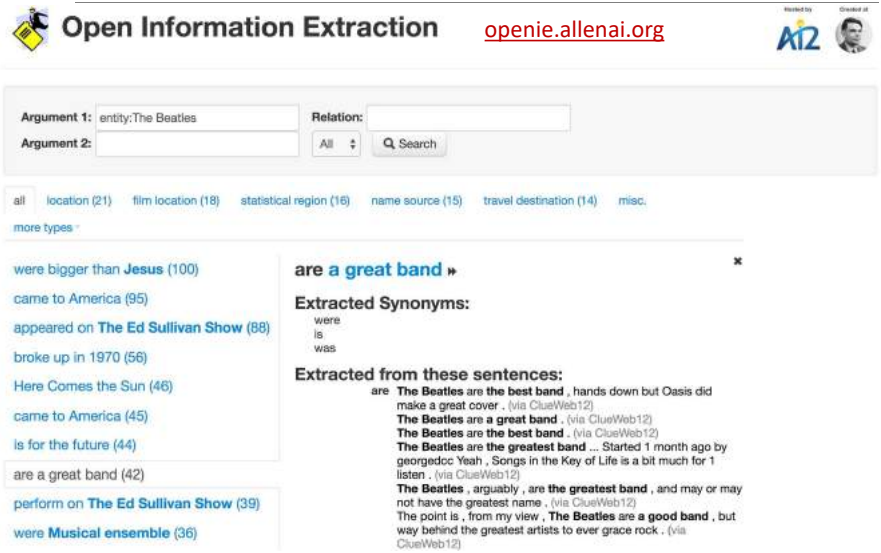
René Witte

What did we do? (alternate view)

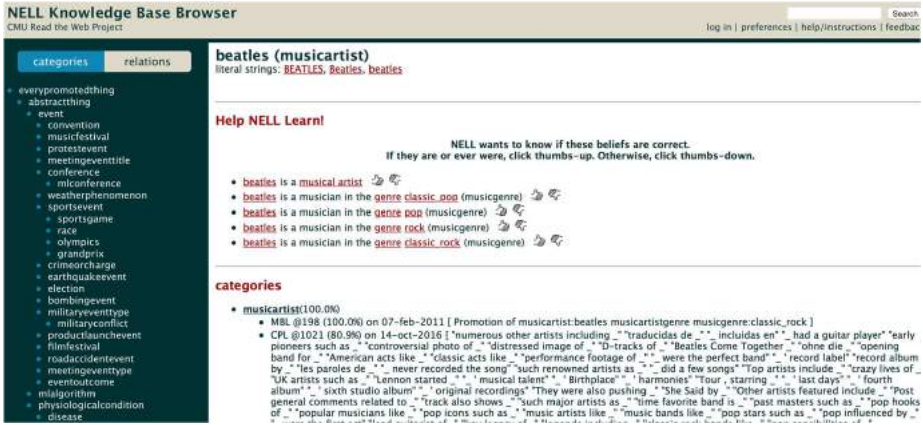


René Witte

Success story: OpenIE (ReVerb)

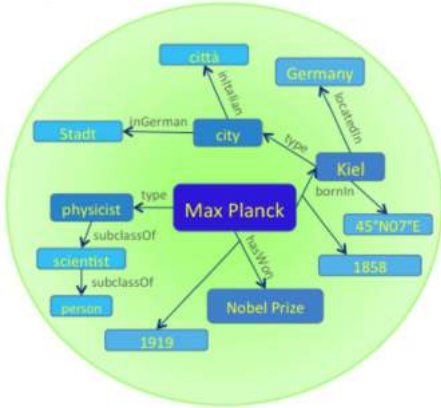


Success story: NELL



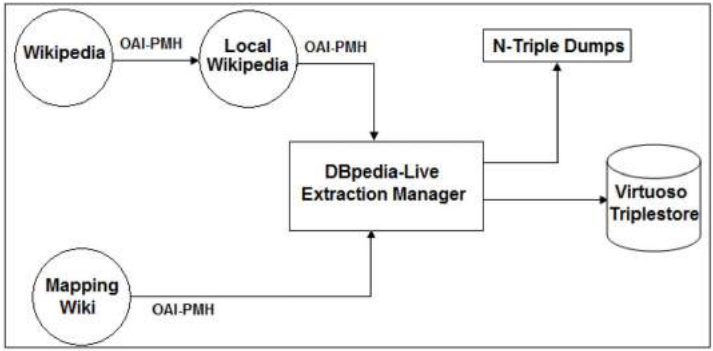
Success story: YAGO

- **Input:** Wikipedia infoboxes, WordNet and GeoNames
- **Output:** KG with 350K entity types, 10M entities, 120M facts
- Temporal and spatial information

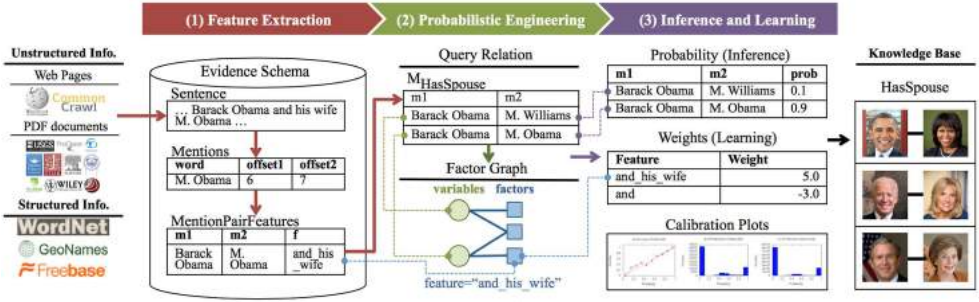


Success story

- DBPedia is automatically extracted structured data from Wikipedia
  - 17M canonical entities
  - 88M type statements
  - 72M infobox statements

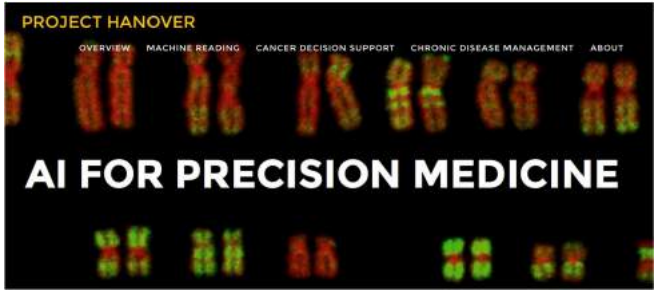


DeepDive



- Best Precision/recall/F1 in KBP-slot filling task 2014 evaluations (31 teams participated)

Interesting application of Knowledge Graphs



Microsoft Research

Chronic disease management:
develop AI technology for predictive and preventive personalized medicine to reduce the national healthcare expenditure on chronic diseases
(90% of total cost)

Aristo Science QA challenge

- Science questions dataset

  ~5K 4-way multiple choice questions

  Frogs lay eggs that develop into tadpoles and then into adult frogs. This sequence of changes is an example of how living things _____

  (A) go through a life cycle

  (B) form a food web

  (C) act as a source of food

  (D) affect other parts of the ecosystem

Science knowledge

frog's life cycle, metamorphosis

Common sense knowledge

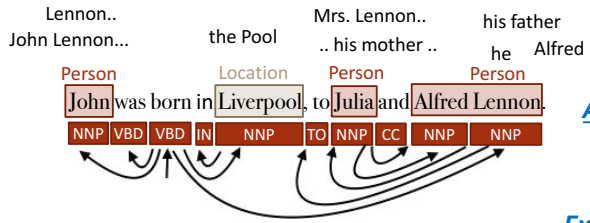
frog is an animal, animals have life cycle

Knowledge Extraction

John was born in Liverpool, to Julia and Alfred Lennon.

Text

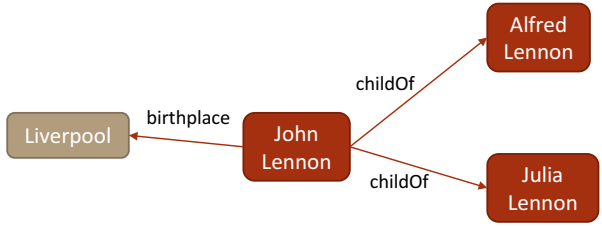
NLP



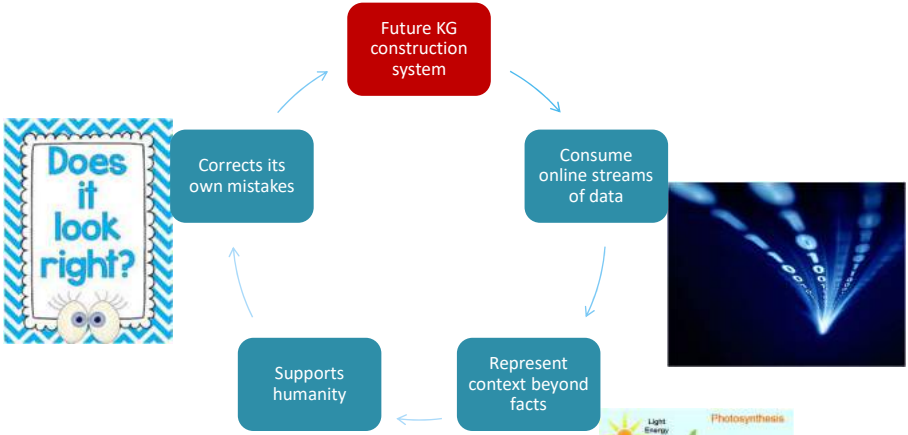
Annotated text

Information Extraction

Extraction graph



Future……



Outline

Reading Material

Required

- [Yu14, Chapters 1, 2] (Introduction, RDF)

Supplemental

- [Wor14] (RDF Primer)
- [RN10, Chapter 12] (Knowledge Representation)
- Graph databases: The best kept secret for effective AI, https://www.youtube.com/watch?v=2ZzGMzitNgo

References

[Her] Ivan Herman.
Tutorial on Semantic Web Technologies.
http://www.w3.org/People/Ivan/CorePresentations/RDFTutorial/.

[RN10] Stuart Russell and Peter Norvig.
Artificial Intelligence: A Modern Approach.
Prentice Hall, 3rd edition, 2010.
https://encore.concordia.ca/iii/encore/record/C__Rb2591108?lang=eng.

[Wor14] World Wide Web Consortium (W3C).
RDF 1.1 Primer.
http://www.w3.org/TR/rdf11-primer/, 24 June 2014.

[Yu14] Liyang Yu.
A Developer's Guide to the Semantic Web.
Springer-Verlag Berlin Heidelberg, 2nd edition, 2014.
Available online at
https://concordiauniversity.on.worldcat.org/oclc/897466408.