KNOWLEDGE GRAPHS 201

Krzysztof Kutt, PhD
Knowledge in AI Systems
WFAIS UJ

THE GRAPHS

Graphs are everywhere

THE OPEN GRAPH PROTOCOL

- https://ogp.me/
- Created by Facebook in 2010
- When you want to add likes to a social network you need clean information (page category, title, canonical URL, image)
  Parsing is difficult – it is better to create simple schemes
- One simple scheme is better than many! (see below)
- The whole model is based on RDF Schema.
- The canonical machine representation is in RDFa. JSON-LD and Microdata are also supportted.

```html
<html xmlns:og="http://opengraphprotocol.org/schema/" xmlns:dc="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/">
<head>
  <title>The Rock (1996)</title>
  <meta property="dc:title" content="The Rock" />
  <meta property="og:type" content="movie" />
  <link rel="canonical" href="http://www.imdb.com/title/tt0117500/" />
  <meta property="foaf:logo" content="http://ia.media-imdb.com/images/rock.jpg" />
  ...
</head>
...
</html>
```

- Now, used also by the Google, and many other (for graphs and links preview)
- Required: `og:type, og:title, og:image, og:url` (unique ID for the graph!)

Try it yourself!

- Source of data: https://www.imdb.com/title/tt0082971/
- Check the data available in the source: https://www.opengraph.xyz/

Source: Facebook (2010), The Open Graph Protocol Design Decisions.

SCHEMA.ORG

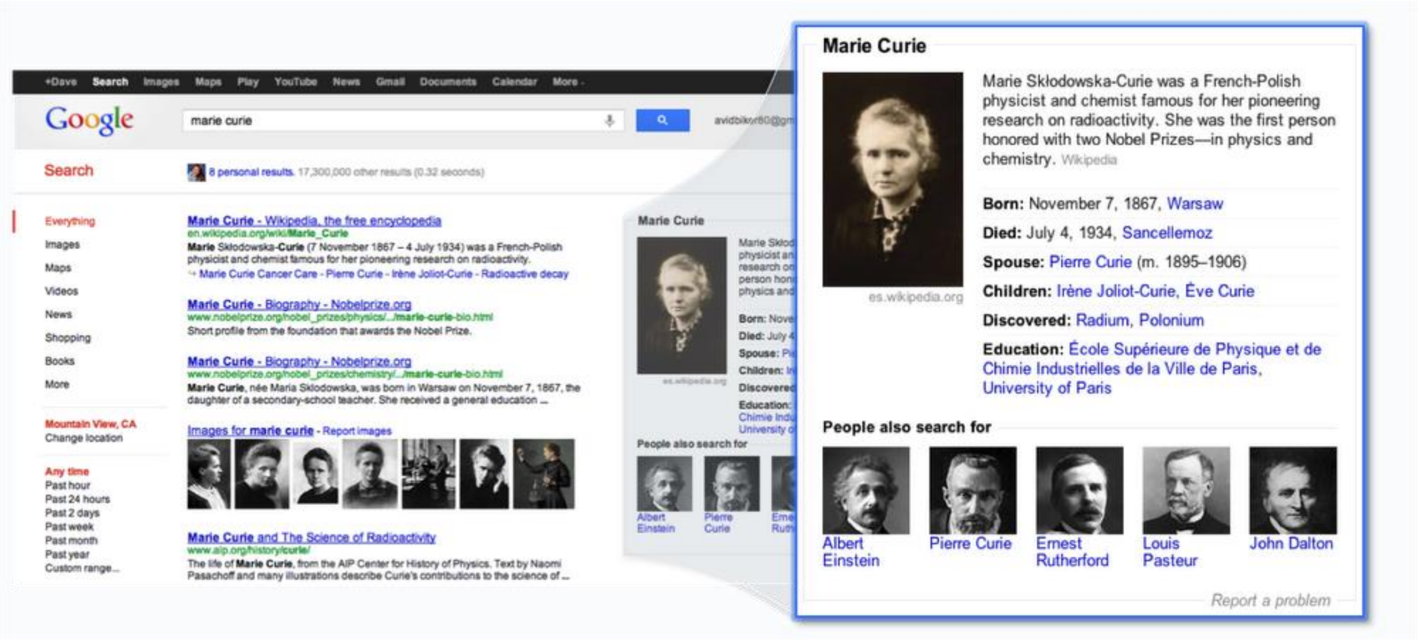
- Founded by Google, Microsoft, Yahoo and Yandex (in 2011)
- Inspired by FOAF, OpenCyc and others
- Shared vocabulary for structured data on the Internet
- Thing is the most generic type
- The whole model is based on RDF Schema.
- The canonical machine representation is in RDFa. JSON-LD and Microdata are also supportted.

Try it yourself!

- Movie schema: https://schema.org/Movie
- Source of data (one of many movie databases): https://www.imdb.com/title/tt0082971/
- Check the data available in the source: https://validator.schema.org/?hl=en-GB
- Search results: https://www.google.com/search?q=Raiders+of+the+Lost+Ark

WEB SEARCH

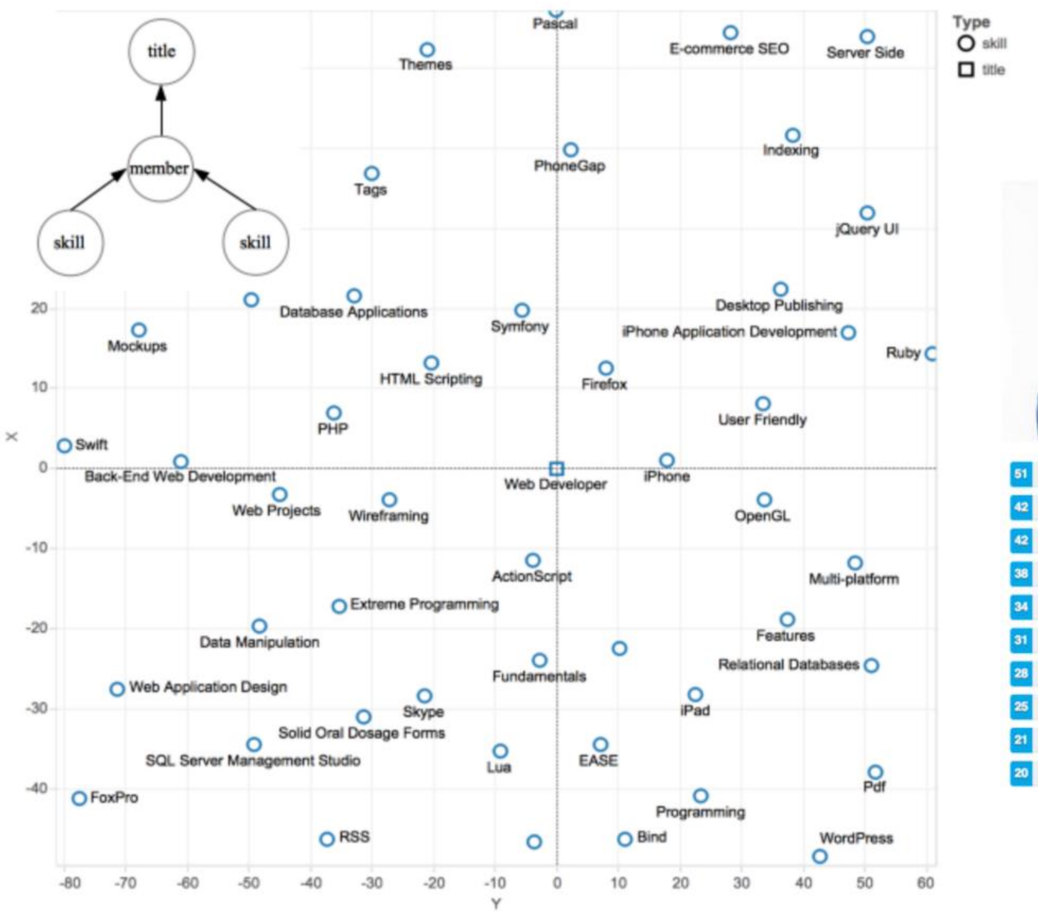
- "Things not strings" paradigm, analogous to semantic search
- Approach promoted by The Google Knowledge Graph
  - It uses https://schema.org/
  - API compliant with JSON-LD ([API introduction]())
  - It is used to generate the rankings of the most notable entities that match certain criteria
  - It is used to fill info in knowledge panel
- Now, used also by other major search engines, e.g., Microsoft Bing ([Bing Entity Search API]())

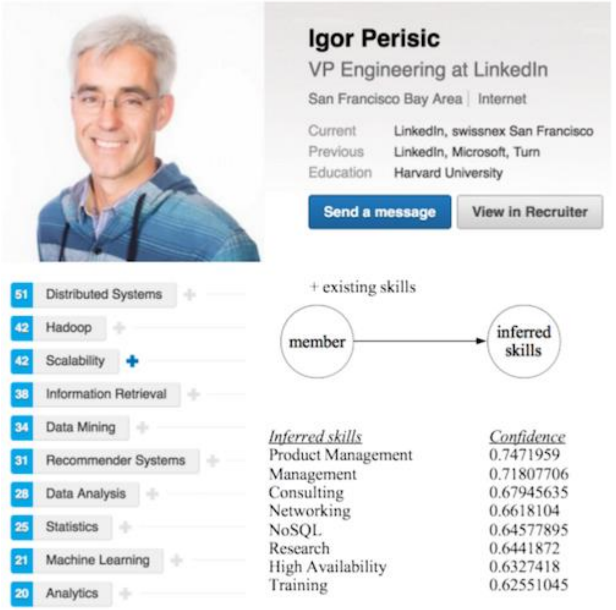


Source: Google (2012), [Introducing the Knowledge Graph: things, not strings]().

SOCIAL NETWORKS

- <u>Facebook:</u>
  - graph describing users, celebrities, places, movies
  - to connect people, understand their interests and provide recommendations
- <u>LinkedIn:</u>
  - users, jobs, skills, etc.
  - for targetted advertising, advanced search and recommendations for jobs-people matches

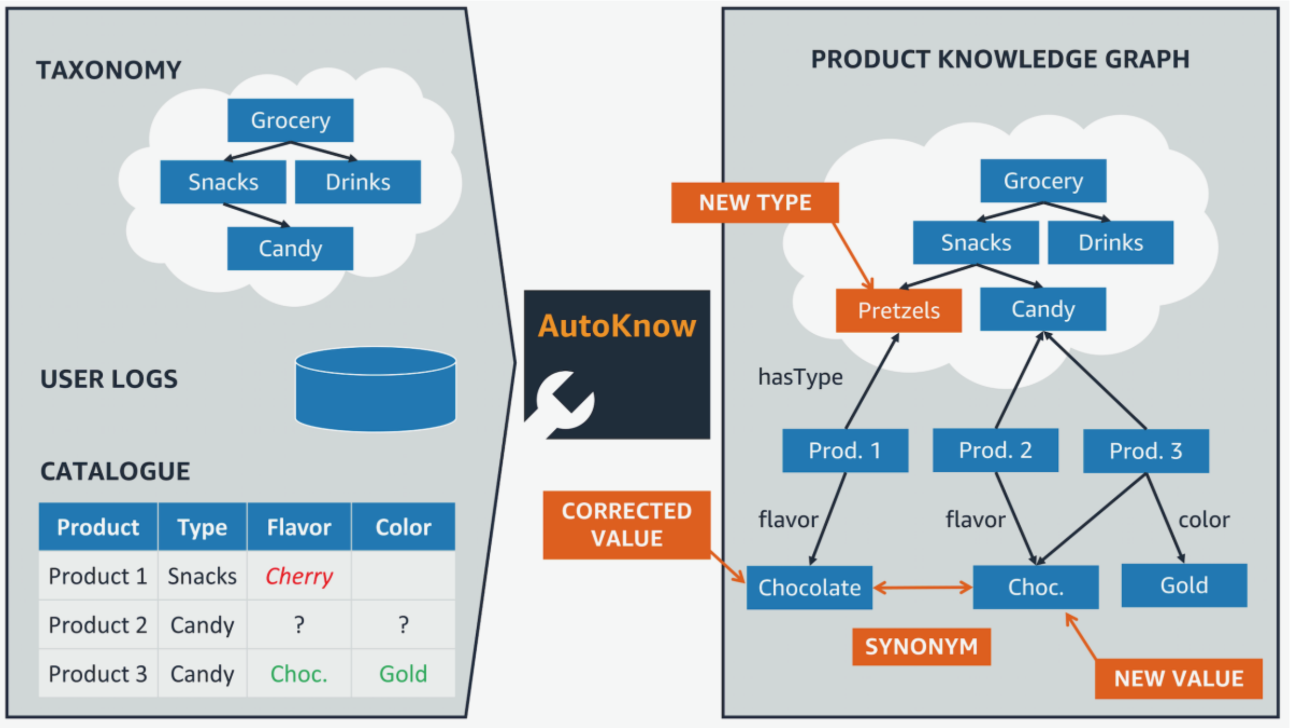




Source: LinkedIn (2016), <u>Building The LinkedIn Knowledge Graph</u>.

COMMERCE

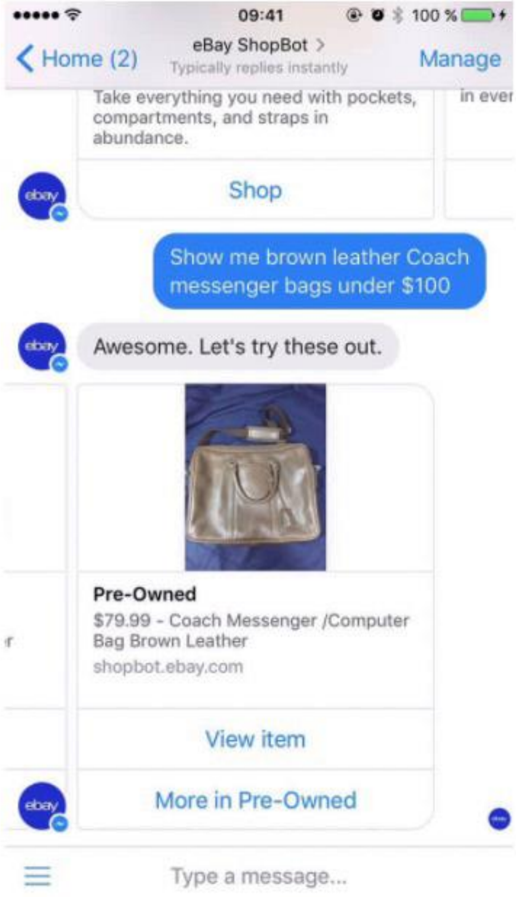
- Enterprise knowledge graphs are used by companies concerned with selling or renting goods and services
- Amazon!
  - Goals: to enable more advanced semantic search and to improve product recommendations
  - AutoKnow: a suite of techniques for automatically augmenting product knowledge graphs with both structured data and data extracted from free-form text sources (see the image)

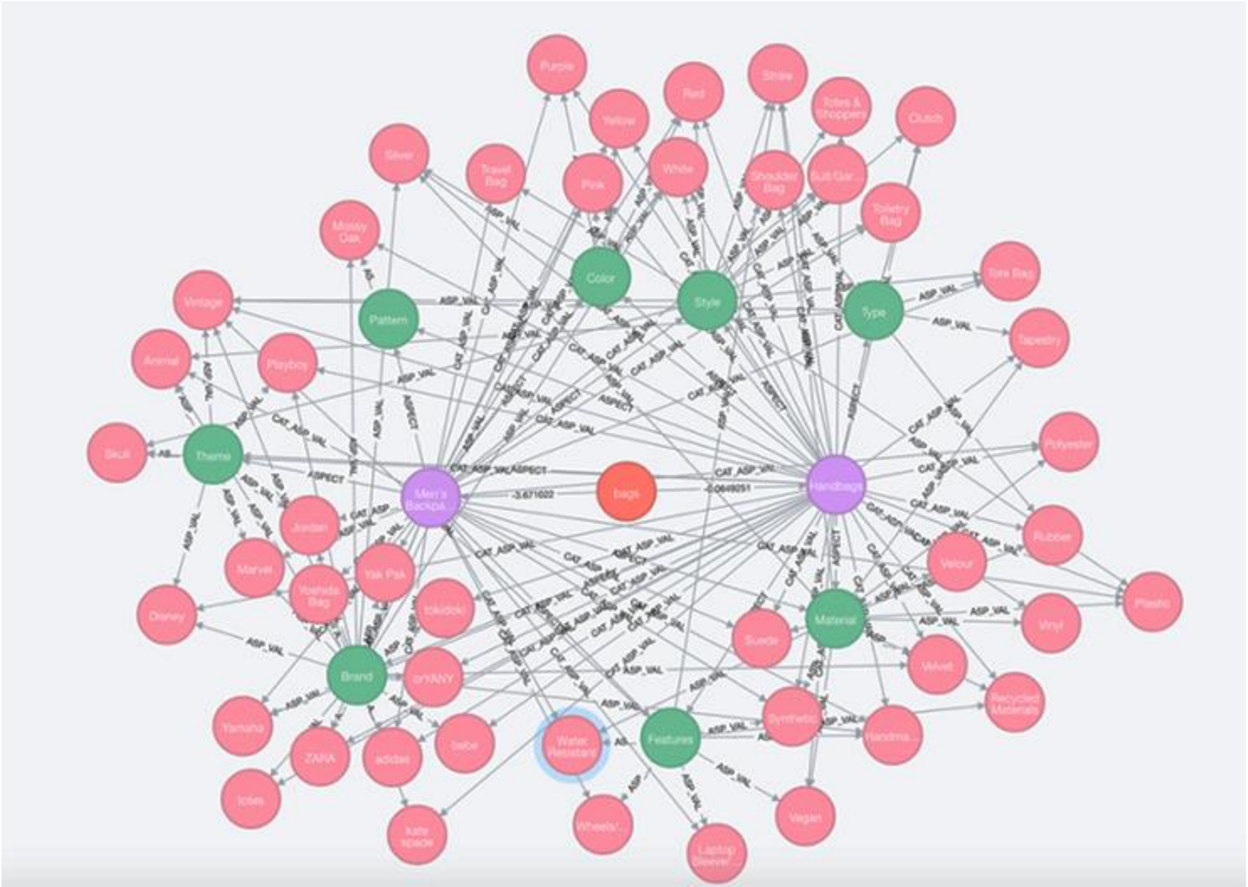


Source: Amazon (2020), Building product graphs automatically.

COMMERCE

- <u>eBay!</u>
  - Graph with product descriptions and shopping behaviour patterns
  - Goal: to power ShopBot – conversational agent

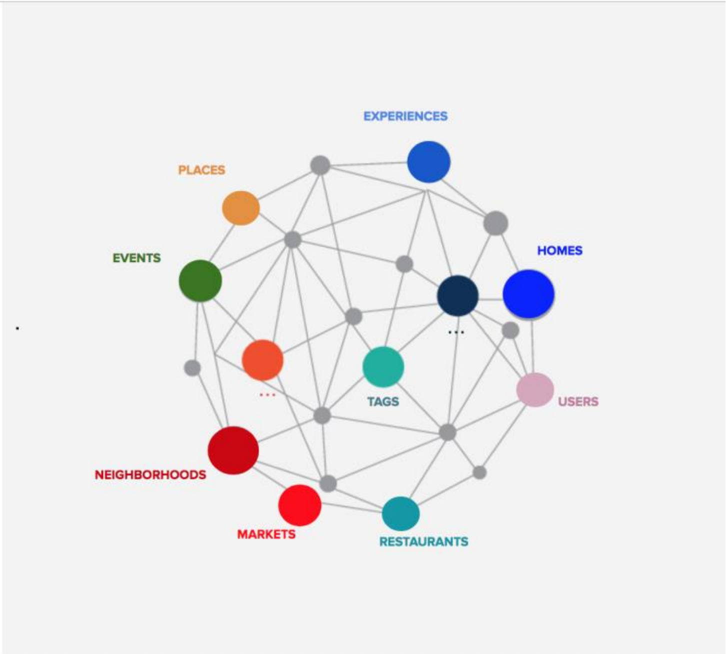


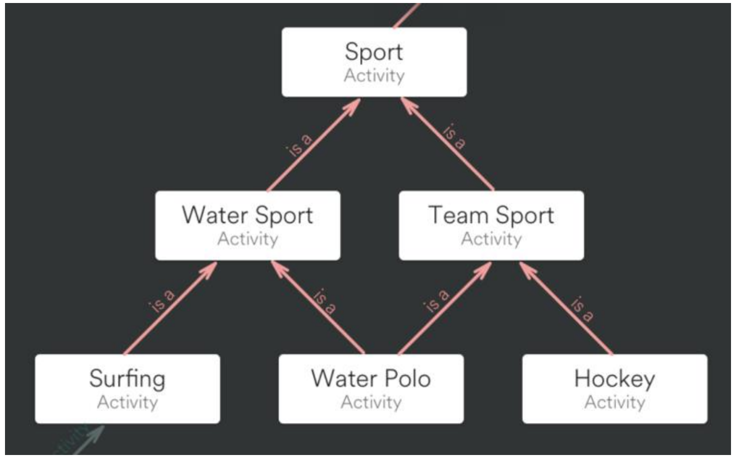


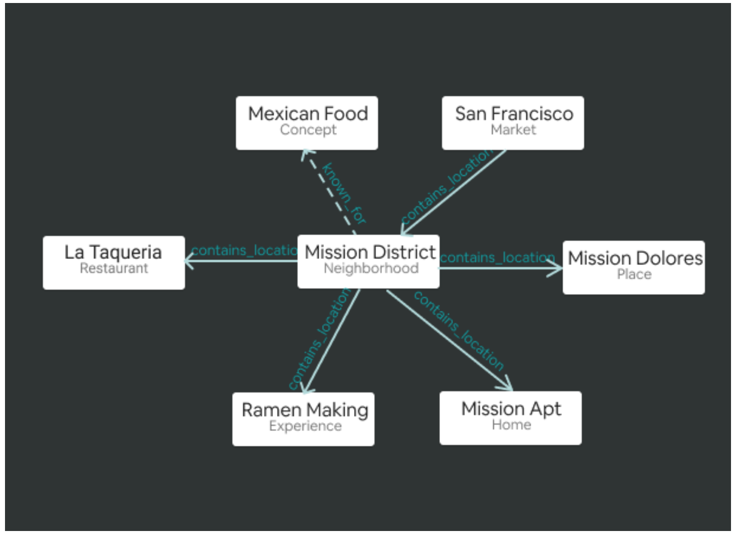
Source: eBay (2017), [Cracking the Code on Conversational Commerce](#).

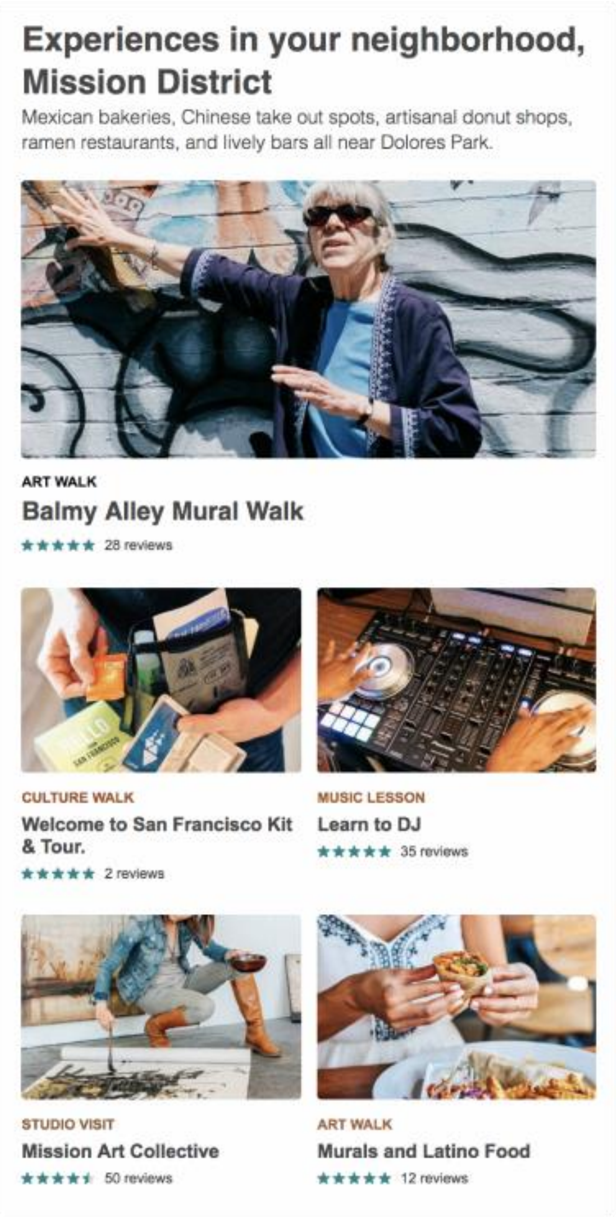
COMMERCE

- [Airbnb!]()
  - Places, events, experiences, etc.
  - Used to recommend atractions available in the neighbourhood of a particular home for rent





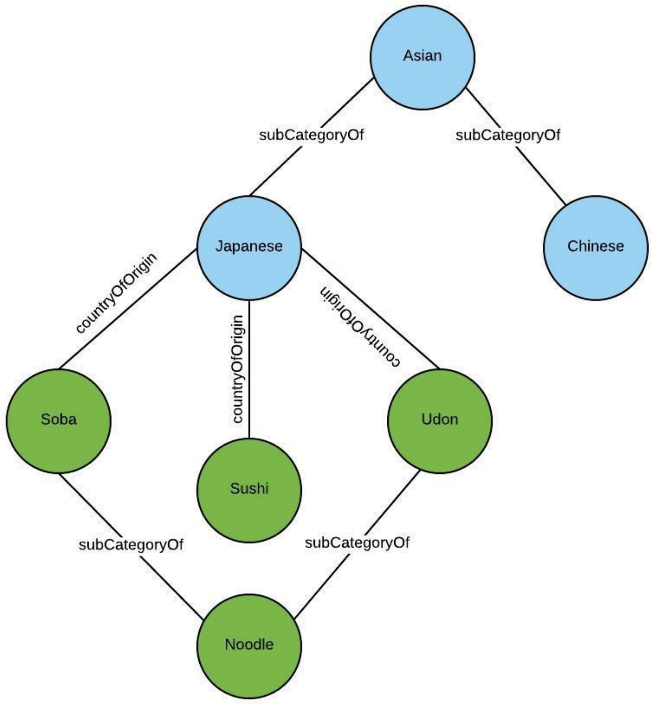


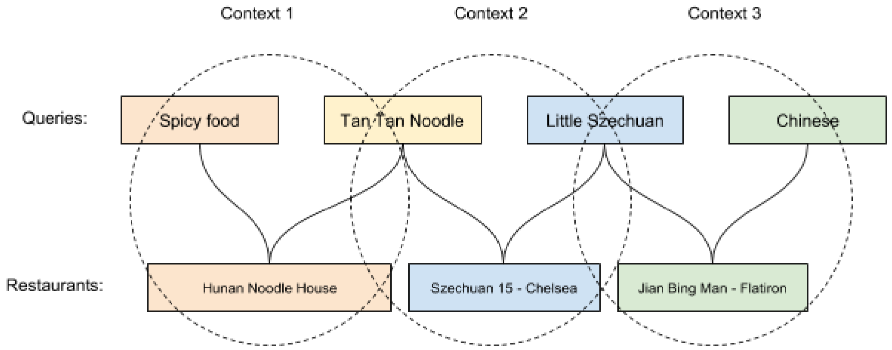


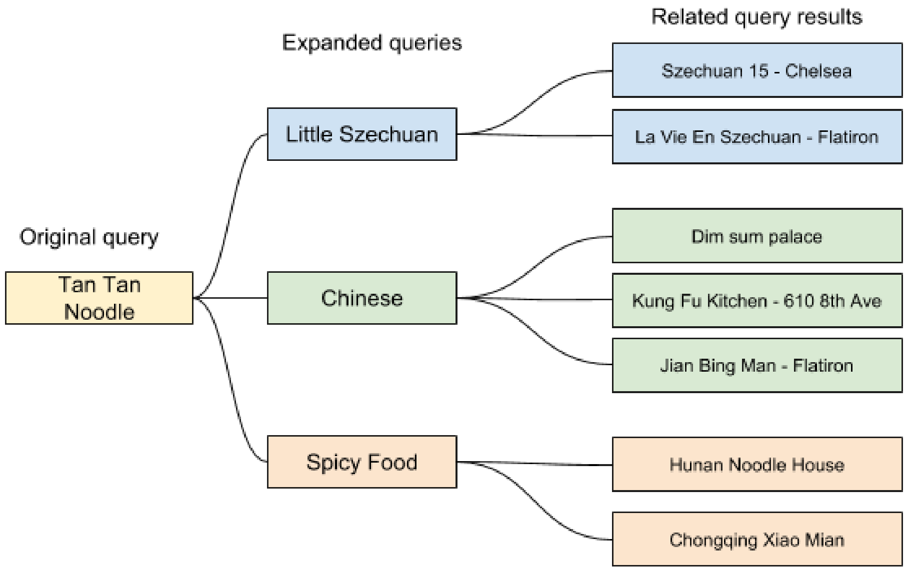
Source: Spencer Chang (2018), [Scaling Knowledge Access and Retrieval at Airbnb]().

COMMERCE

- <u>Uber!</u>
  - Graph focused on food and restaurants
  - Goal: offer semantic search and recommendations for people who do not know exactly what they want to eat
  - Query Expansion (see the figure): “Tan Tan Noodle” expands to three queries that further retrieve a set of relevant restaurant







Source: Uber (2018), [Food Discovery with Uber Eats: Building a Query Understanding Engine]().

QUERY LANGUAGES

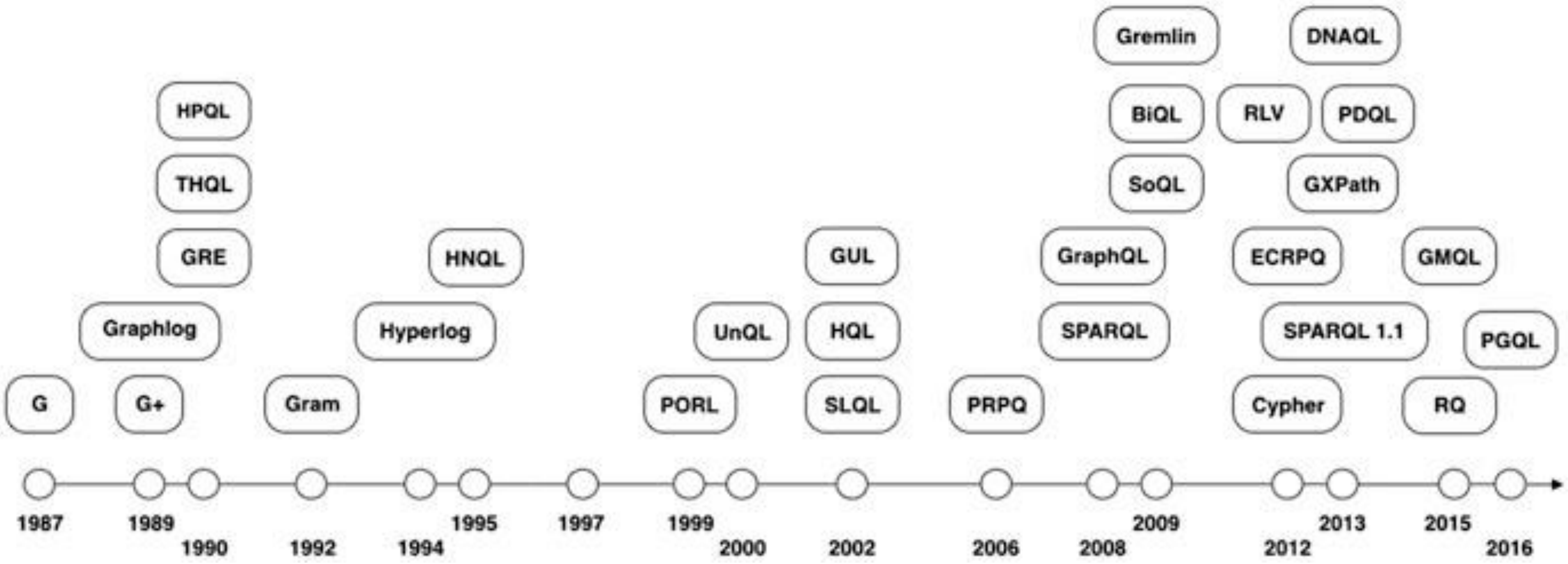
Knowledge is there
to be extracted

Give me control of a database query language,
and I care not who makes its engine

-- George Anadiotis

Source: G. Anadiotis (2018), Back to the future: Does graph database success hang on query language?.

EVOLUTION OF GRAPH QUERY LANGUAGES



Source: R. Angles and C. Gutierrez (2018), An Introduction to Graph Data Management in: Graph Data Management, pp. 1-32, Springer.

GRAPH QUERY LANGUAGES

Directed Edge-Labelled Graphs

- SPARQL (for RDF)

Property Graphs

- Gremlin (for Apache TinkerPop)
- Cypher (by Neo4j) → openCypher (since 2015)
- GQL (Graph Query Language) -- ISO standard published in 2024!

GREMLIN

- For Apache TinkerPop (Graph Computing Framework)
- Groovy/Java-based; native support also for other languages: C#, JS, Python, …
- Graph traversal language
- Sequence of steps on the data stream:
  (a) map-step (objects → stream transformation)
  (b) filter-step (remove objects from the stream)
  (c) sideEffect-step (compute statistics)

```
// What are the names of Gremlin's friends' friends?
g.V().has("name","gremlin"). //get the vertex with name "gremlin"
  out("knows").     //traverse to the people that Gremlin knows
  out("knows").     //traverse to the people those people know
  values("name")    //get those people's names
```

In SPARQL?

```
// What are the names of the projects created by two friends?
g.V().match(
  as("a").out("knows").as("b"),   //there exists some "a" who knows "b"
  as("a").out("created").as("c"), //there exists some "a" who created "c"
  as("b").out("created").as("c"), //there exists some "b" who created "c"
  as("c").in("created").count().is(2) //the "c" was created by 2 people
).select("c").by("name")  //get the name of all matching "c" projects
```

In SPARQL?

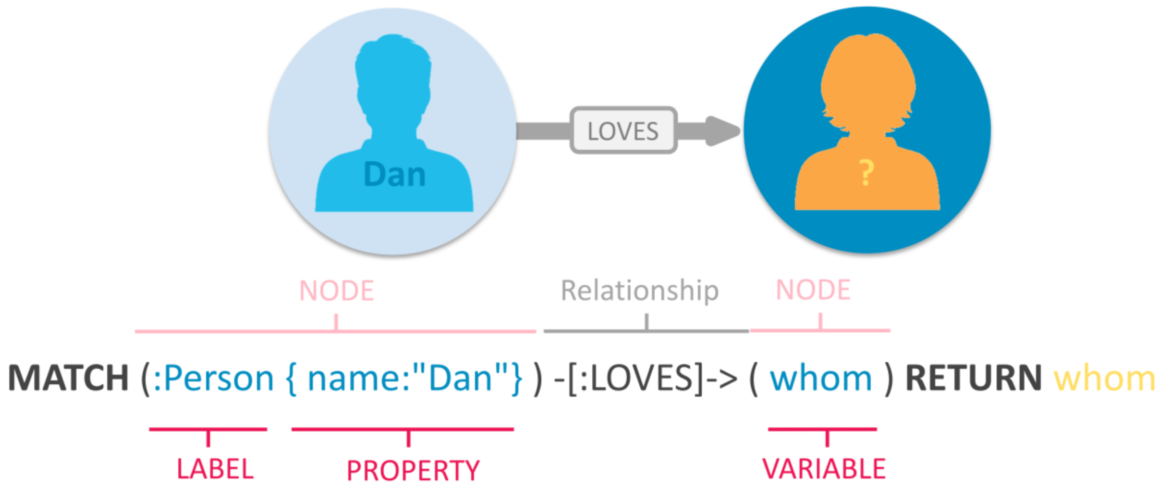
CYPHER

- Developed for Neo4j, but now driven by the community (openCypher)
- Remote execution by Cypher REST API
- Docs: https://neo4j.com/developer/cypher/
- Syntax based on ASCII art

```
//node
(variable:Label {propertyKey: 'propertyValue'})
//relationship
-[variable:RELATIONSHIP_TYPE]->
//Cypher pattern
(node1:LabelA)-[rel1:RELATIONSHIP_TYPE]->
(node2:LabelB)
```

- Keywords:

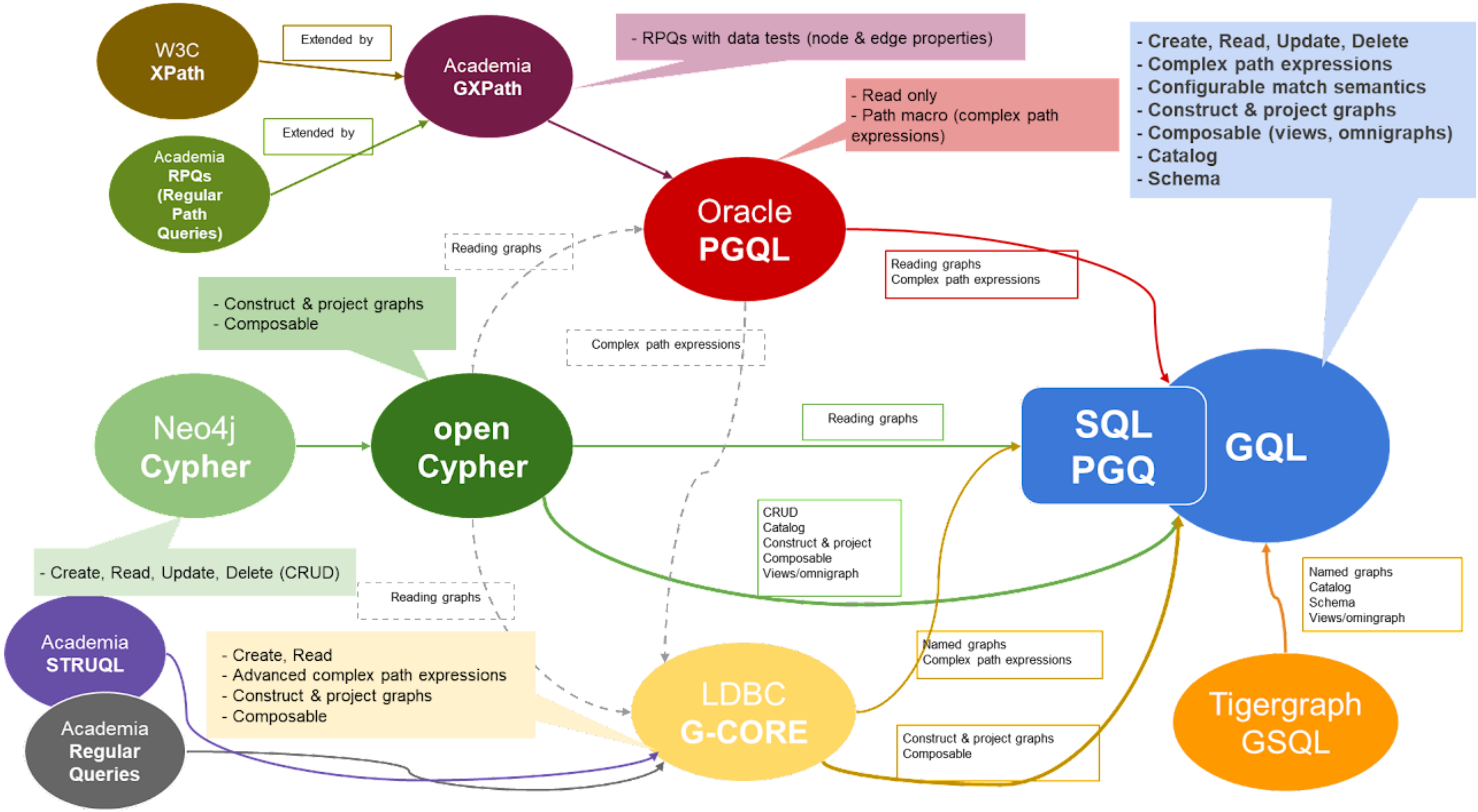
```
MATCH search pattern            (WHERE  in SPARQL)
WHERE additional constraints (FILTER in SPARQL)
RETURN something               (SELECT in SPARQL)
```



In SPARQL?

GQL (GRAPH QUERY LANGUAGE)

- One language to rule them all (i.e., an ISO standard similar to SQL for relational databases)
- Work started in 2019. Standard published in 04.2024
- Cypher as a starting point!
- For more details, see: https://www.gqlstandards.org/



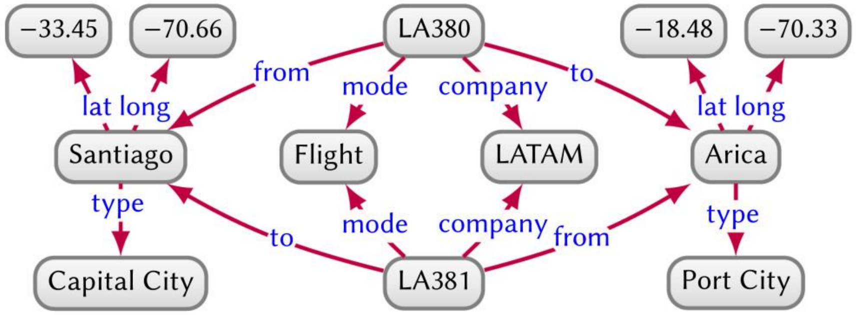
Source: ELWG (2019), Existing Languages.

PROPERTY GRAPHS 101

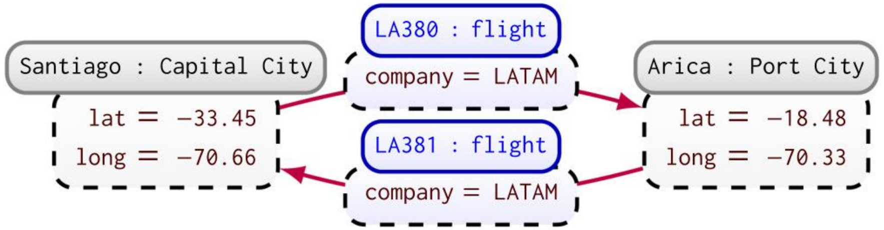
Is there anything beyond RDF triples?

PROPERTY GRAPHS

- Labels and property-value pairs can be associated with nodes and edges
- Not yet standardised
  (available in popular graph databases
  but particular implementations may differ)
- More intuitive representation, but requires more intricate query languages, formal semantics and inductive techniques



(a) Del graph



(b) Property graph

Source: A. Hogan et al. (2022), Knowledge Graphs, ACM Computing Surveys, 54(4), pp. 1-37.

Property Graph vs RDF Knowledge Graph

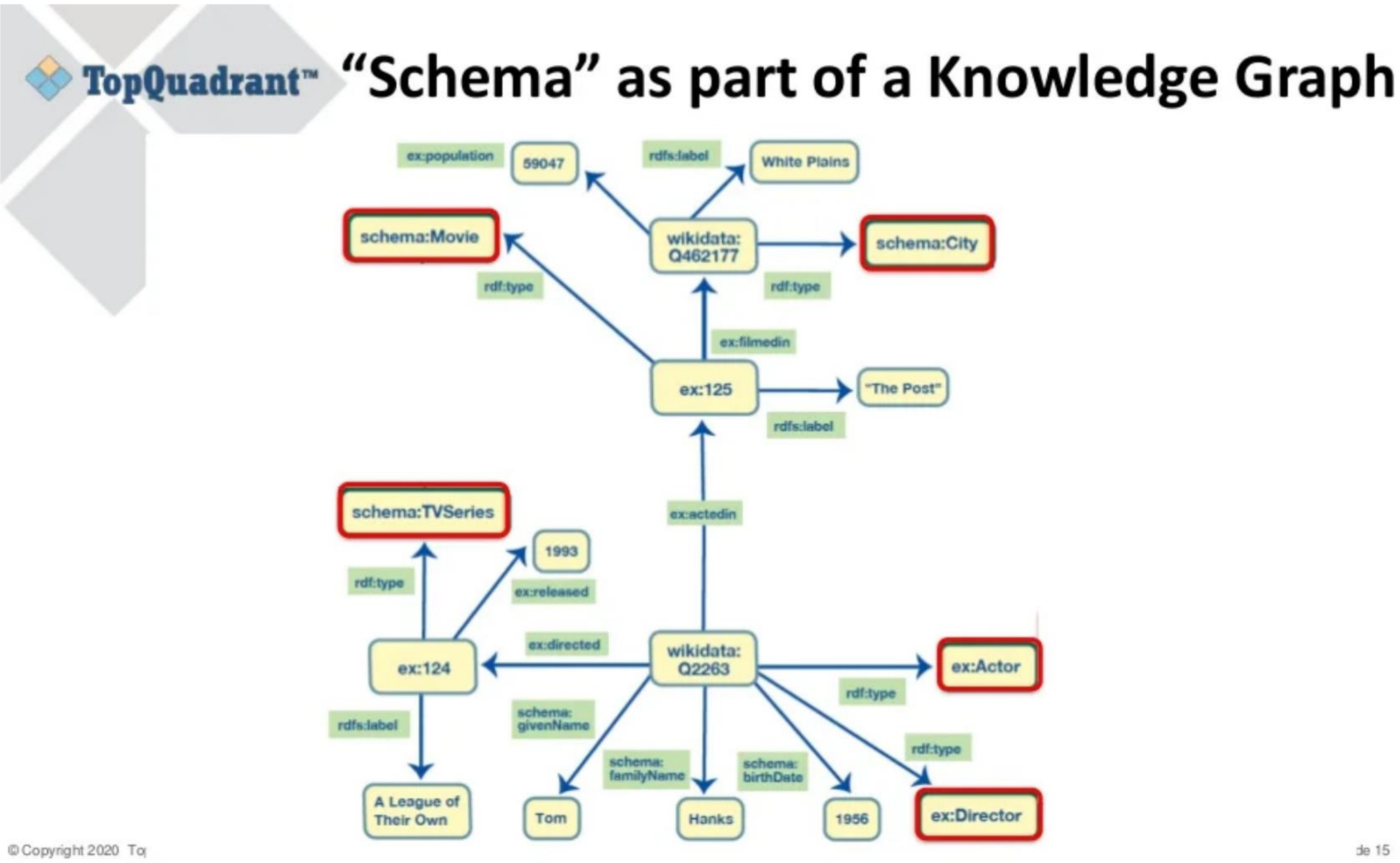
Property Graph	Knowledge Graph
IDs are internal to a graph database, user has no control over them.	IDs are global – URIs, meant to be under users control to enable combining different graphs
Properties are literal values. They are fundamentally different from nodes and relationships.	Canonical structure. Everything is stored as nodes and links connecting them. A literal value is a node like any other. Property is any link – to a resource or a literal.

Slide from Knowledge Graphs vs. Property Graphs talk by Irene Polikoff from TopQuadrant (2020).

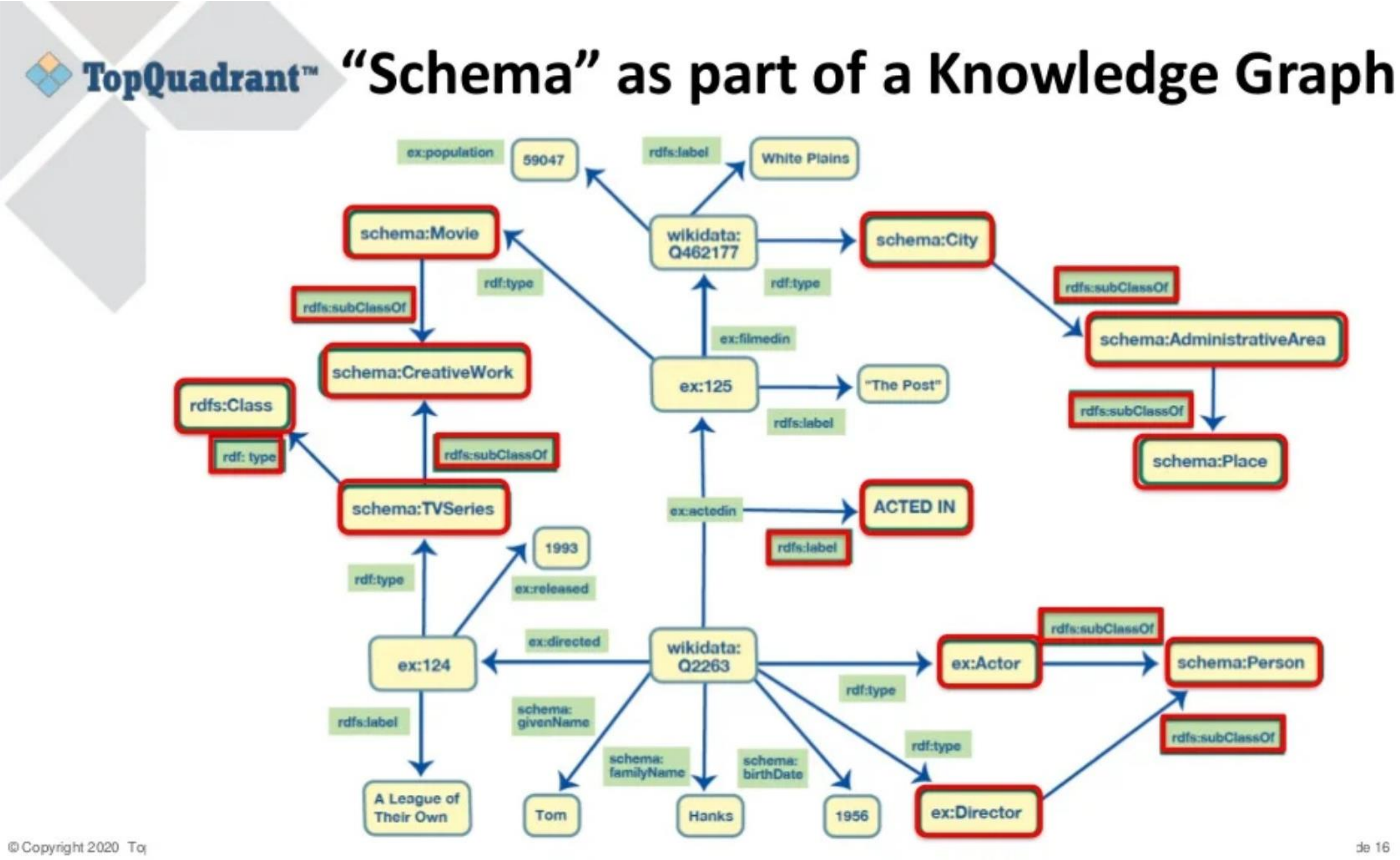
Property Graph vs RDF Knowledge Graph

Property Graph	Knowledge Graph
IDs are internal to a graph database, user has no control over them.	IDs are global – URIs, meant to be under user's control to enable combining different graphs.
Properties are literal values. They are fundamentally different from nodes and relationships.	Canonical structure. Everything is stored as nodes and links connecting them. A literal value is a node like any other. Property is any link - to a resource or a literal.
Schema (semantics of data) is not a part of the graph.	**Rich schemas, including rules are a part of the graph.**

Slide from Knowledge Graphs vs. Property Graphs talk by Irene Polikoff from TopQuadrant (2020).



Slide from Knowledge Graphs vs. Property Graphs talk by Irene Polikoff from TopQuadrant (2020).



Slide from Knowledge Graphs vs. Property Graphs talk by Irene Polikoff from TopQuadrant (2020).

Property Graph vs RDF Knowledge Graph

Property Graph	Knowledge Graph
IDs are internal to a graph database, user has no control over them.	IDs are global – URIs, meant to be under user's control to enable combining different graphs.
Properties are literal values. They are fundamentally different from nodes and relationships.	Canonical structure. Everything is stored as nodes and links connecting them. A literal value is a node like any other. Property is any link - to a resource or a literal.
Schema (semantics of data) is not a part of the graph.	Rich schemas, including rules are a part of the graph.
Each relationship (link or arch) uniquely identifies "node – link – node" combination. Relationships can be annotated with additional facts, but properties can't be.	**IDs of properties (links) are re-used. Thus, they do not uniquely identify a "node – link - node" combination that uses it. There is a way to give these triples identity. Any triple can be annotated with additional facts.**

Slide from Knowledge Graphs vs. Property Graphs talk by Irene Polikoff from TopQuadrant (2020).

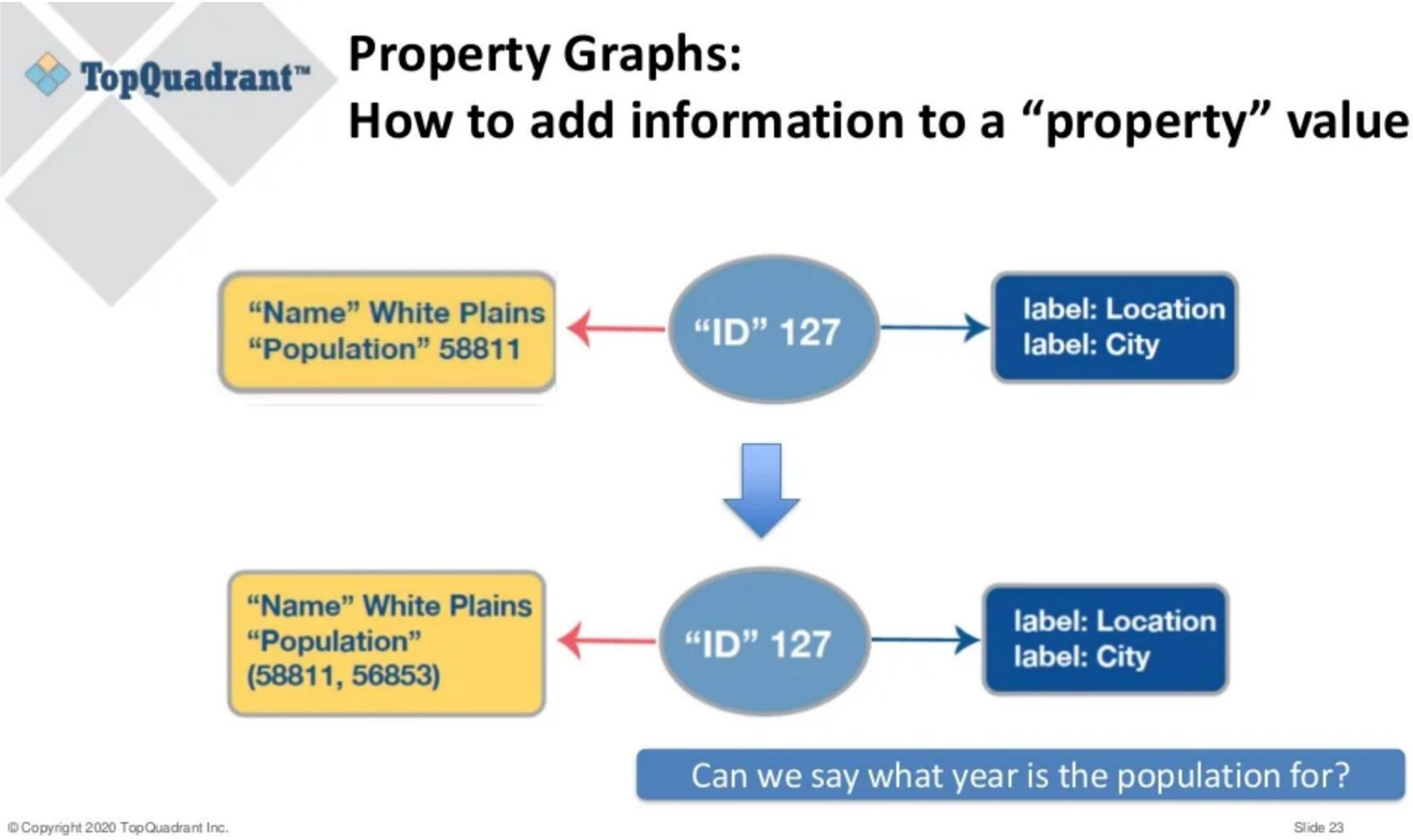
TopQuadrant™

Property Graph vs RDF Knowledge Graph

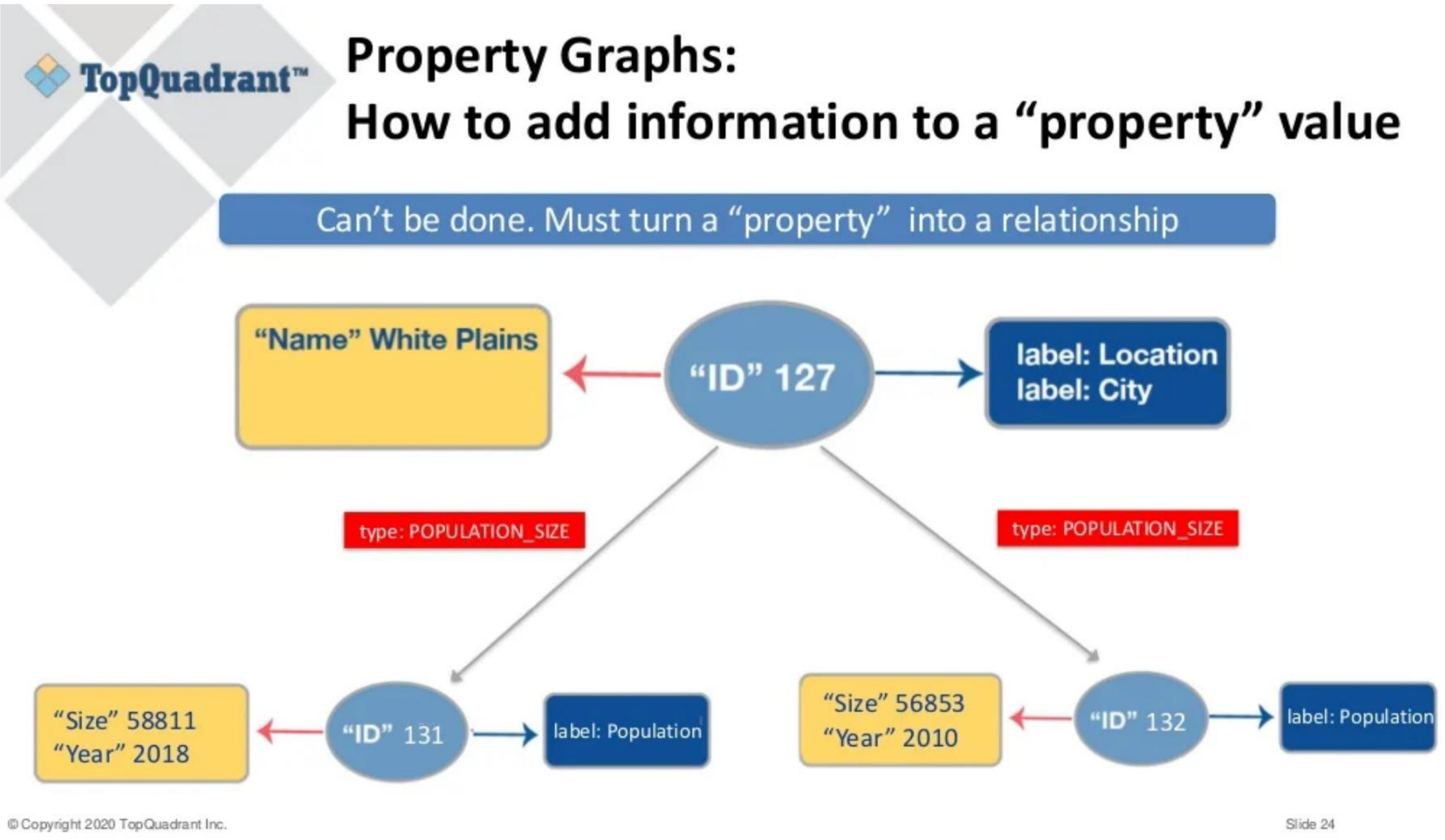
Property Graph	Knowledge Graph
IDs are internal to a graph database, user has no control over them.	IDs are global – URIs, meant to be under user's control to enable combining different graphs.
Properties are literal values. They are fundamentally different from nodes and relationships.	Canonical structure. Everything is stored as nodes and links connecting them. A literal value is a node like any other. Property is any link - to a resource or a literal.
Schema (semantics of data) is not a part of the graph.	Rich schemas, including rules are a part of the graph.
Each relationship (link or arch) uniquely identifies "node – link – node" combination. Relationships can be annotated with additional facts, but properties can't be.	IDs of properties (links) are re-used. Thus, they do not uniquely identify a "node – link - node" combination that uses it. There is a way to give these triples identity. Any triple can be annotated with additional facts.
Changes in the graph design require restructure/re-load of the data and changes to all impacted queries.	**Graphs can evolve and changes in the design can often be done with minimal impact on existing data and queries.**
Product-specific query languages, variants of Cypher, increasingly, GraphQL support, introspection not integrated	**Query standard – SPARQL. Increasingly, GraphQL support. In EDG: introspection and auto-generation of GraphQL Schemas.**

Slide 30

Slide from [Knowledge Graphs vs. Property Graphs] talk by Irene Polikoff from TopQuadrant (2020).



Slide from Knowledge Graphs vs. Property Graphs talk by Irene Polikoff from TopQuadrant (2020).



Slide from Knowledge Graphs vs. Property Graphs talk by Irene Polikoff from TopQuadrant (2020).

TopQuadrant™

Property Graph vs RDF Knowledge Graph

Property Graph	Knowledge Graph
IDs are internal to a graph database, user has no control over them.	IDs are global – URIs, meant to be under user's control to enable combining different graphs.
Properties are literal values. They are fundamentally different from nodes and relationships.	Canonical structure. Everything is stored as nodes and links connecting them. A literal value is a node like any other. Property is any link - to a resource or a literal.
Schema (semantics of data) is not a part of the graph.	Rich schemas, including rules are a part of the graph.
Each relationship (link or arch) uniquely identifies "node – link – node" combination. Relationships can be annotated with additional facts, but properties can't be.	IDs of properties (links) are re-used. Thus, they do not uniquely identify a "node – link - node" combination that uses it. There is a way to give these triples identity. Any triple can be annotated with additional facts.
Changes in the graph design require restructure/re-load of the data and changes to all impacted queries.	Graphs can evolve organically. Changes in the design can often be accomplished with minimal impact on existing data and queries.
Product-specific query languages, variants of Cypher, increasingly, GraphQL support, introspection not integrated	Query standard – SPARQL. Increasingly, GraphQL support. In EDG: introspection and auto-generation of GraphQL Schemas.
No standard serialization for export.	**Standard serializations supported by all products – RDF/XML, Turtle, N3 and JSON-LD formats.**

Slide from Knowledge Graphs vs. Property Graphs talk by Irene Polikoff from TopQuadrant (2020).

RDF 1.2 (A.K.A. RDF*)

- Properties on edges for RDF
- Reduces the mismatch between Linked Data (based on RDF) and Property Graphs
- Useful, e.g., for representing temporal context (<u>when</u> the particular property was true)
- Still under development by the community; for more details see [the dedicated page](#) (RDF*) and current working drafts of [the RDF 1.2 spec](#)

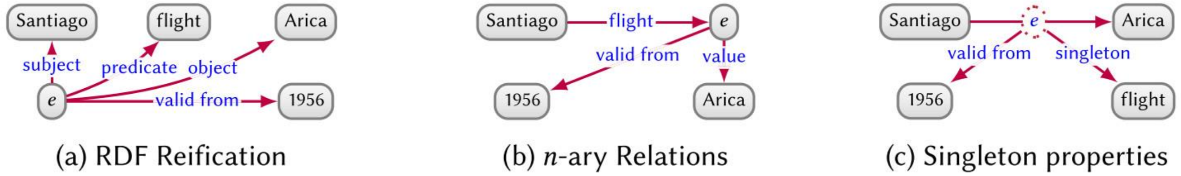


Fig. 9. Three representations of temporal context on an edge in a directed-edge labelled graph.

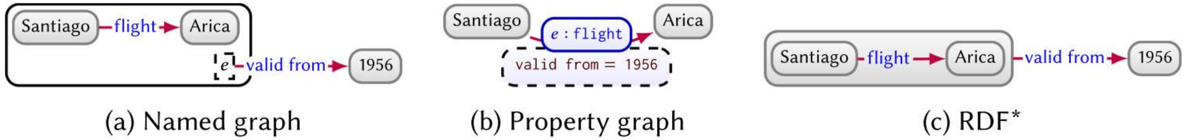


Fig. 10. Three higher-arity representations of temporal context on an edge.

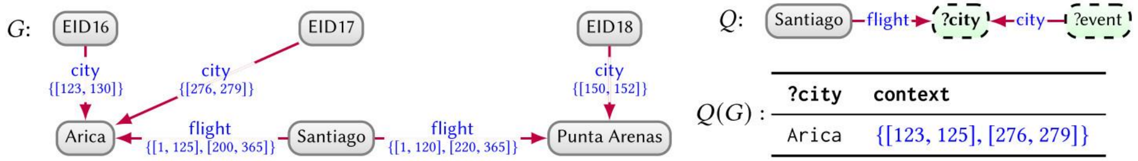


Fig. 11. Example query on a temporally annotated graph.

Source: A. Hogan et al. (2022), [Knowledge Graphs](#), ACM Computing Surveys, 54(4), pp. 1-37.

THERE IS NO SINGLE DEFINITION

- *James (1992)*: "A knowledge graph is a kind of semantic network... One of the essential differences between knowledge graphs and semantic networks is the explicit choice of only a few types of relations"
- *Zhang (2002)*: "A new method of knowledge representation, [which] belongs to the category of semantic networks. In principle, the composition of a knowledge graph is including concept (tokens and types) and relationship (binary and multivariate relation)"
- *Singhal (Google, 2012)*: "A graph that understands real-world entities and their relationships to one another: things, not strings"
- *Ehrlinger and Wöß (2016)*: "A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge"
- *Columbia University (2019)*: "An organized and curated set of facts that provide support for models to understand the world"

Source: Mike Bergman (2019), A Common Sense View of Knowledge Graphs.

WHAT THEY ARE NOT

- A specific language or data model, such as RDF, concept graphs, or OWL, is not required
- No specific schema or formal logic is required
- Types or type classification are not required
- Neither instances, nor attributes, nor concepts, nor specific relations are required, but one or two is
- A specific scope, broad or narrow, is not required
- Statements in the knowledge graph need not be 'triples', but they do need to be some form of knowledge assertion

Source: Mike Bergman (2019), A Common Sense View of Knowledge Graphs.

KNOWLEDGE GRAPH

So, we can go back to our definition from Knowledge Graphs 101…

Knowledge graph is a graph of data intended to accumulate and convey **knowledge of the real world,** whose **nodes represent entities** of interest and whose **edges represent** potentially different **relations** between these entities

Definition by: A. Hogan et al. (2022), Knowledge Graphs, ACM Computing Surveys, 54(4), pp. 1-37.

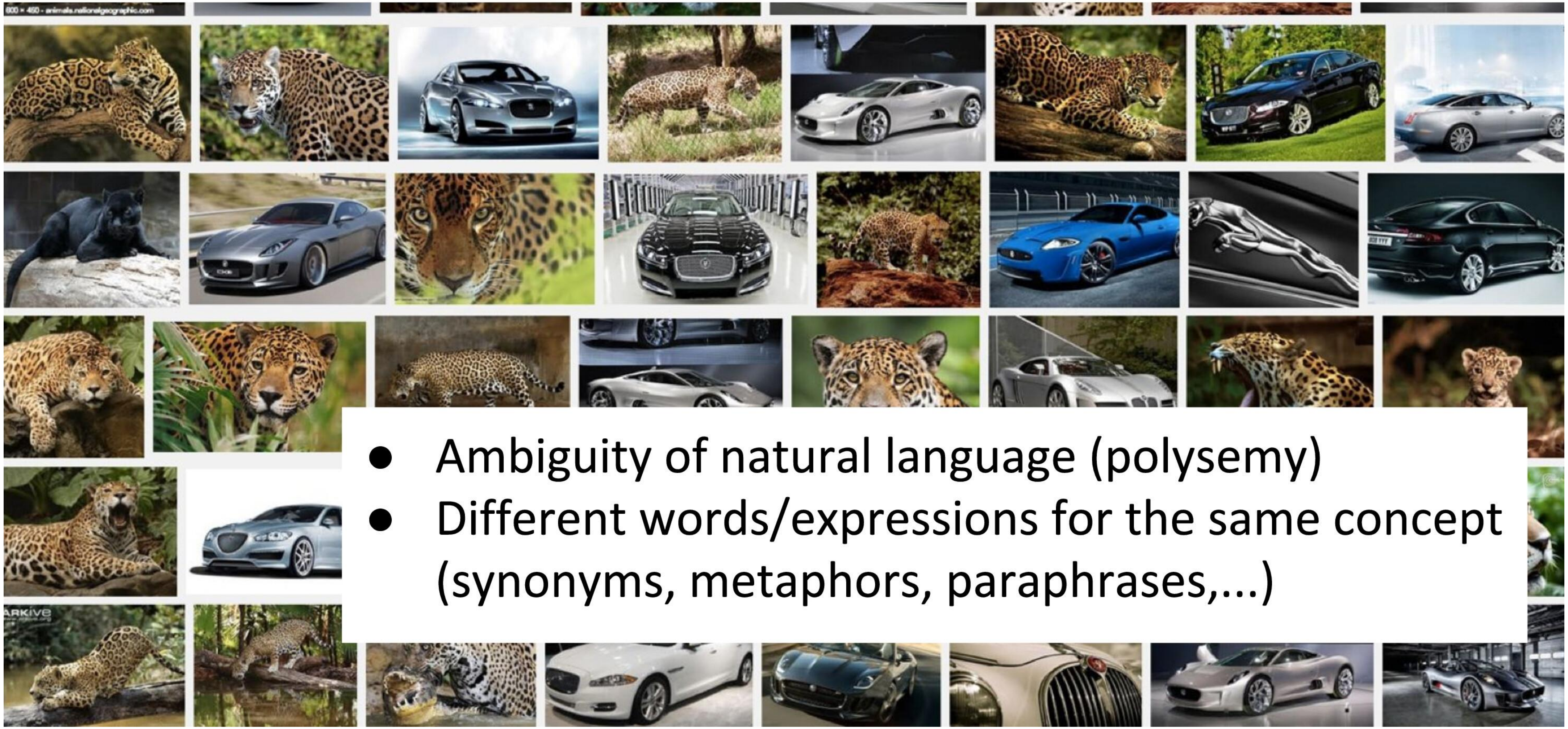
SEMANTIC SEARCH AND RECOMMENDATIONS

Would graphs help search engines?

The Information Retrieval Dilemma

Karlsruher Institut für Technologie

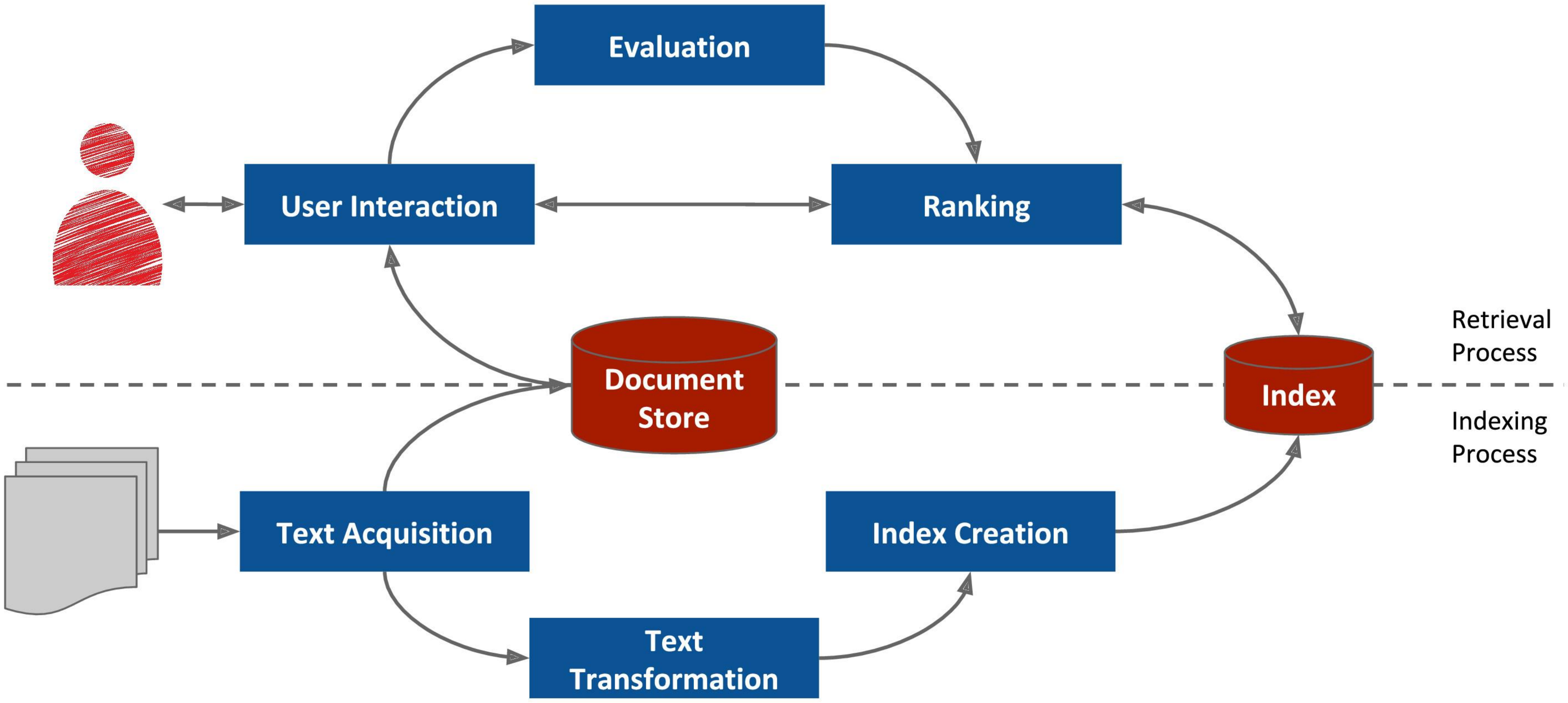
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure



- Ambiguity of natural language (polysemy)
- Different words/expressions for the same concept (synonyms, metaphors, paraphrases,...)

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

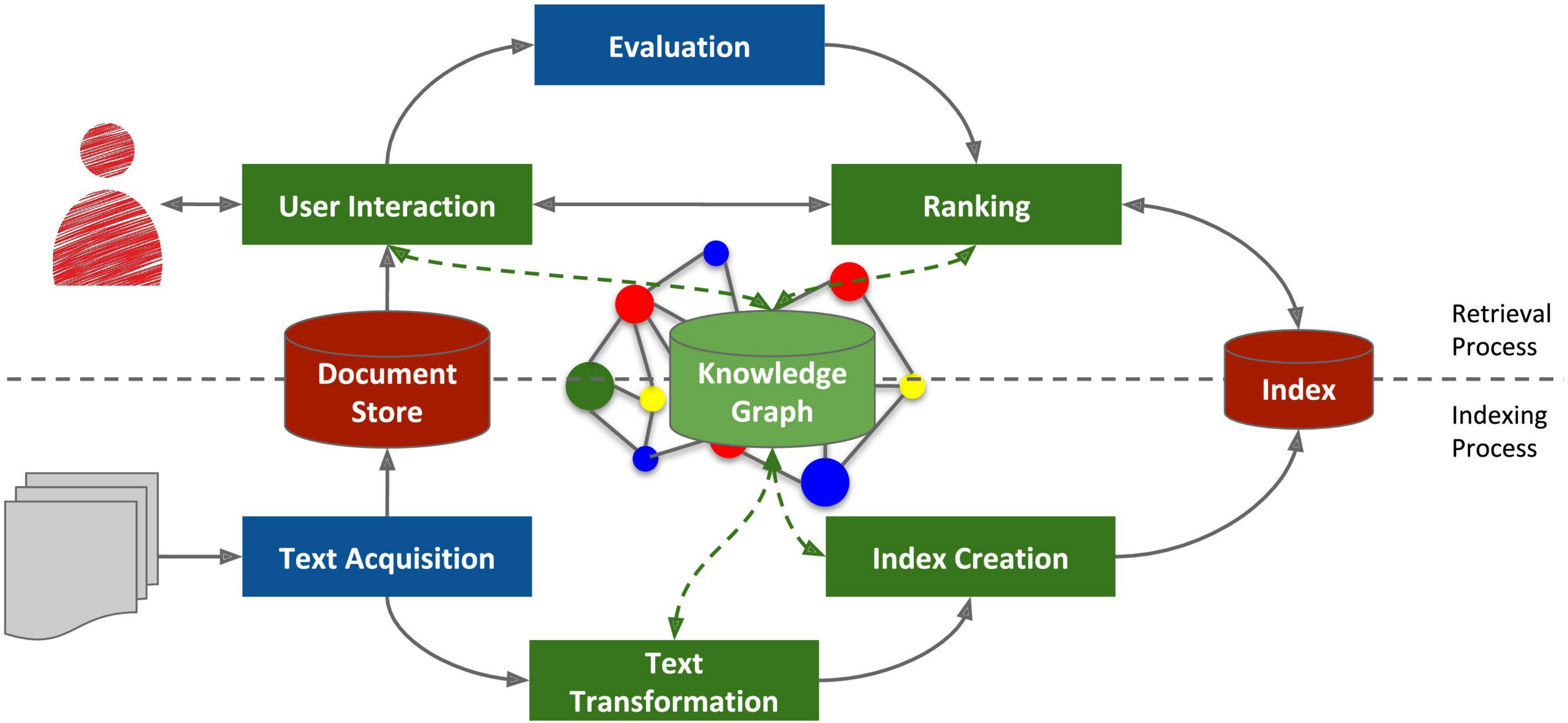
The Information Retrieval Process



[1]

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Knowledge Graph Supported Retrieval Process



[1]

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Knowledge Graph Supported Retrieval Process

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- **Prerequisite:**
  Document Annotation with explicit semantics, e.g. semantic entities

Example for Linked Data Based Document Annotation

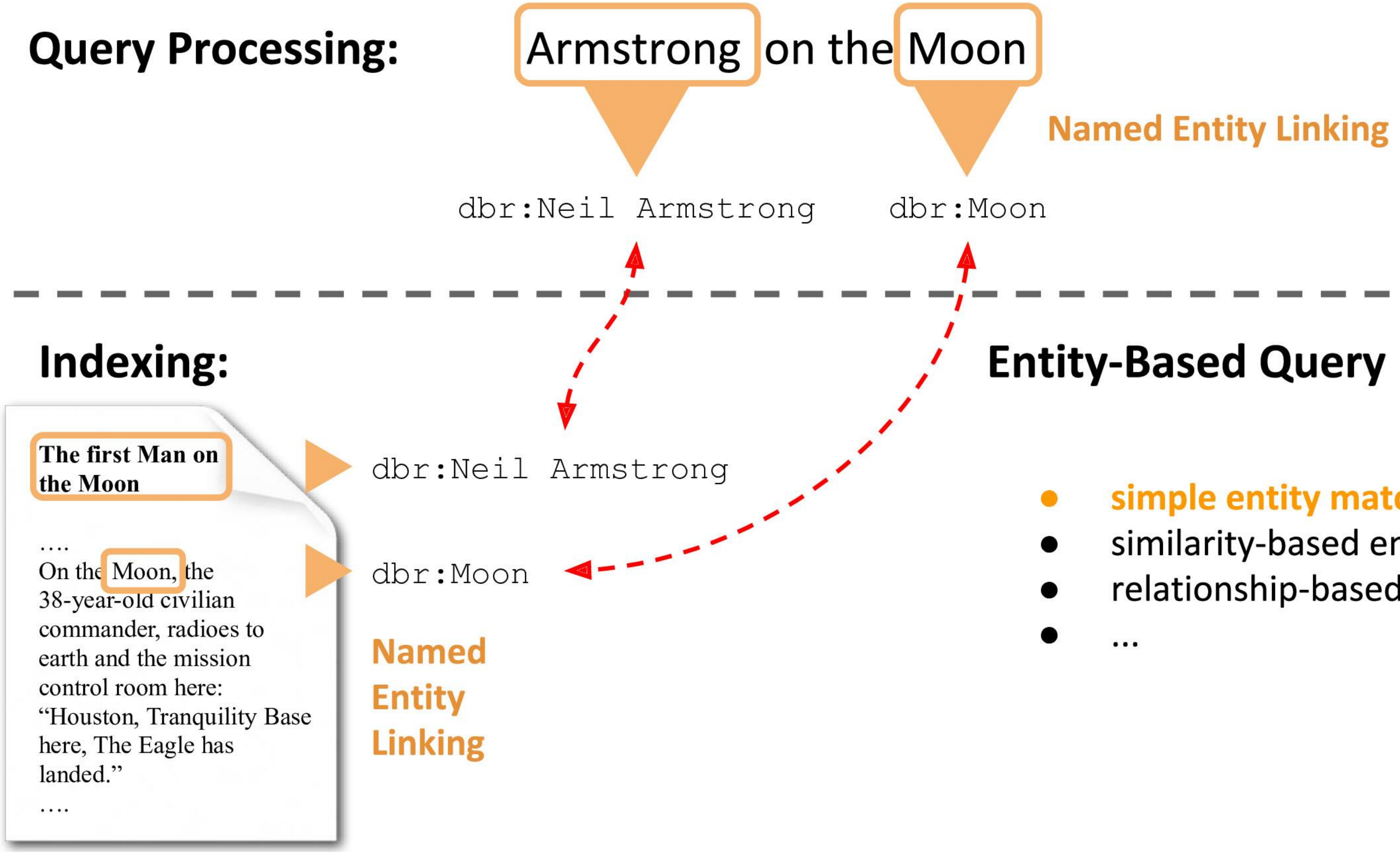
http://scihi.org/neil-armstrong/

- Enables **entity-based Information Retrieval**
  - Language independent

[1]

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Entity Based Search



Entity-Based Query Matching

- simple entity matching
- similarity-based entity matching
- relationship-based entity matching
- …

[1]

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Entity Based Search

Query Processing: Armstrong on the Moon

Named Entity Linking

dbr:Neil Armstrong dbr:Moon

Indexing

The 2nd Man on the Moon

….

Legendary astronaut Buzz Aldrin has revealed some captivating pieces of Apollo 11 memorabilia on social media in the last few days.

...

Named Entity Linking

dbr:Moon

dbr:Buzz_Aldrin

semantic similarity

dbr:Neil_Armstrong

Entity-Based Query Matching

- simple entity matching
- similarity-based entity matching
- relationship-based entity matching
- ...

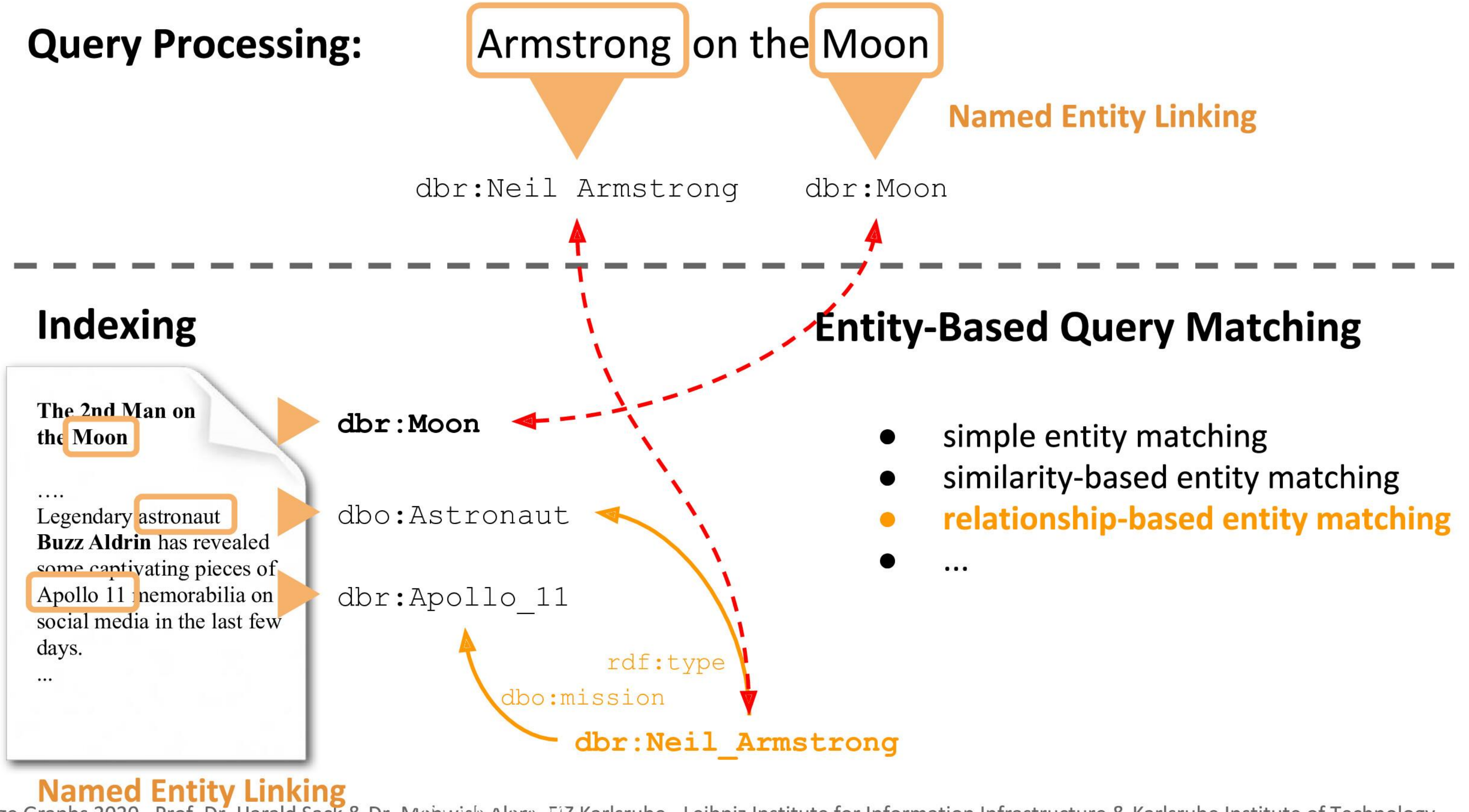
Two entities are considered **semantically similar**

- if they share property/value pairs
- if they share properties with similar values

[1]

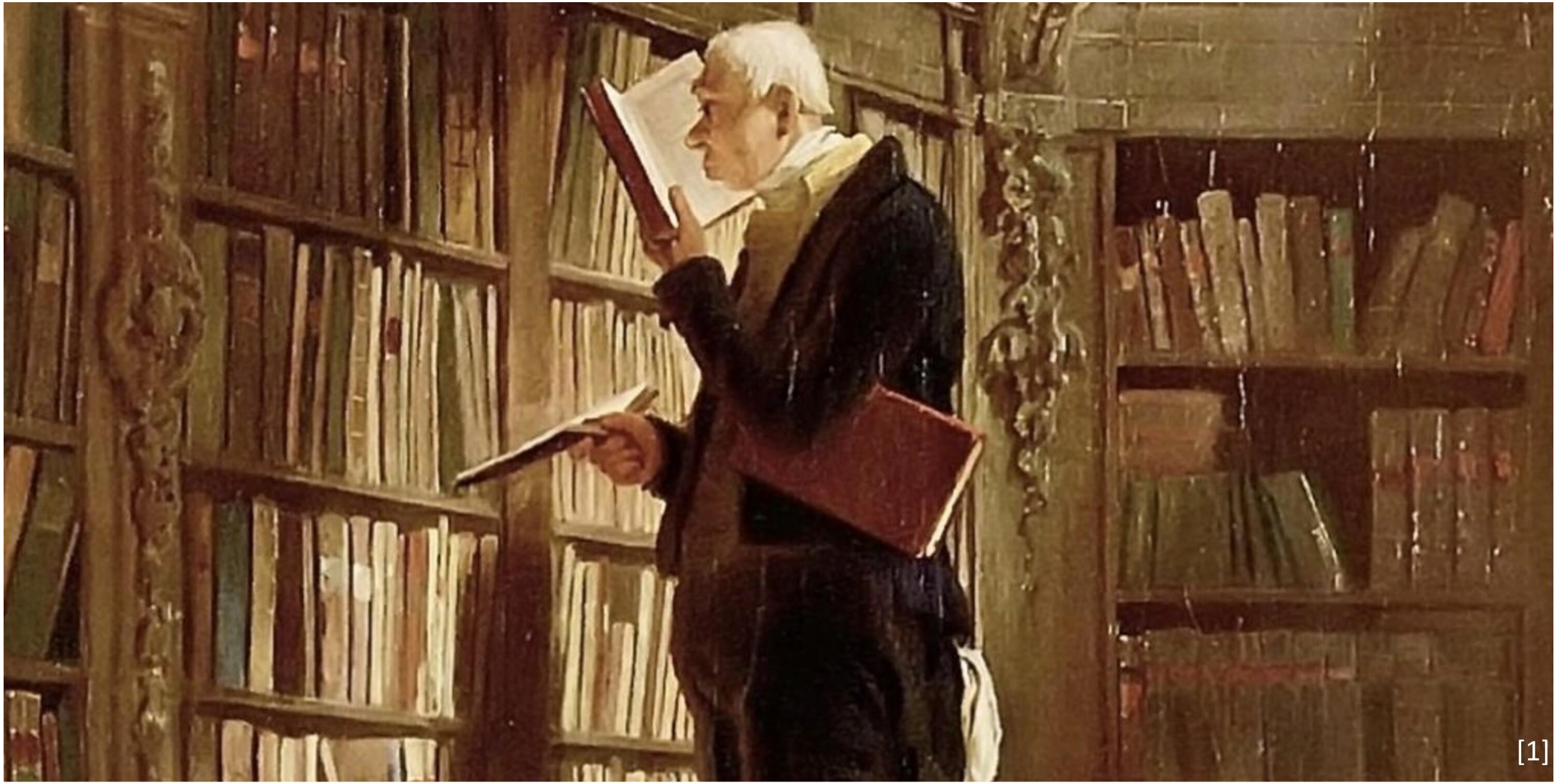
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Entity Based Search



[1]

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).



Retrieval vs. Exploration

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

The Retrieval Problem

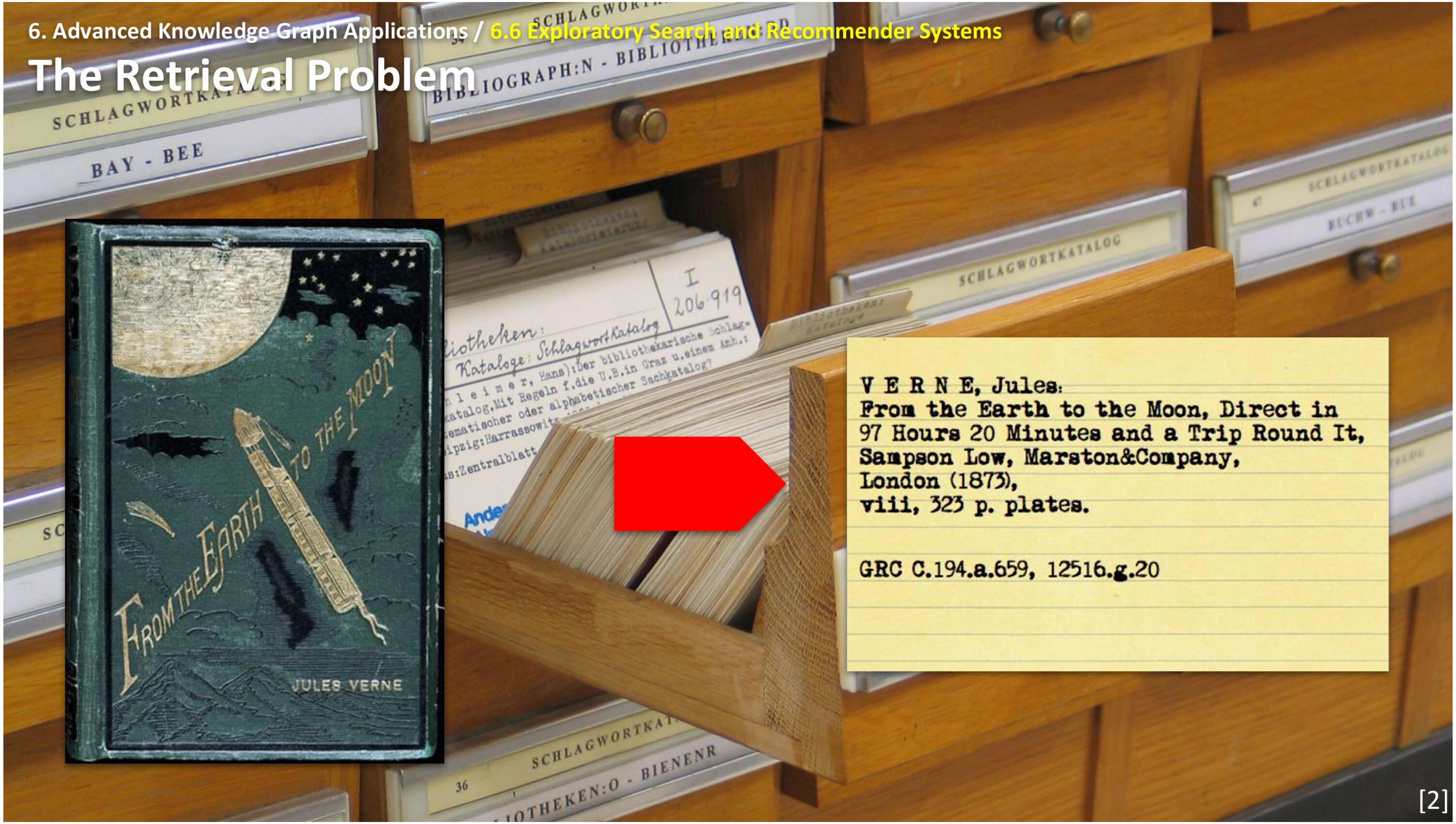
- **Retrieval Problem:**
  - you are looking for **something specific**
    i.e. you know what you are looking for
- How to **specify your search request**?
  - e.g. for a (specific) book:
    author name, title, etc.
- Often you are using
  - (unique) identifier
  - descriptive metadata

[3]

Author: Jules Verne
Title: From the Earth to the Moon

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

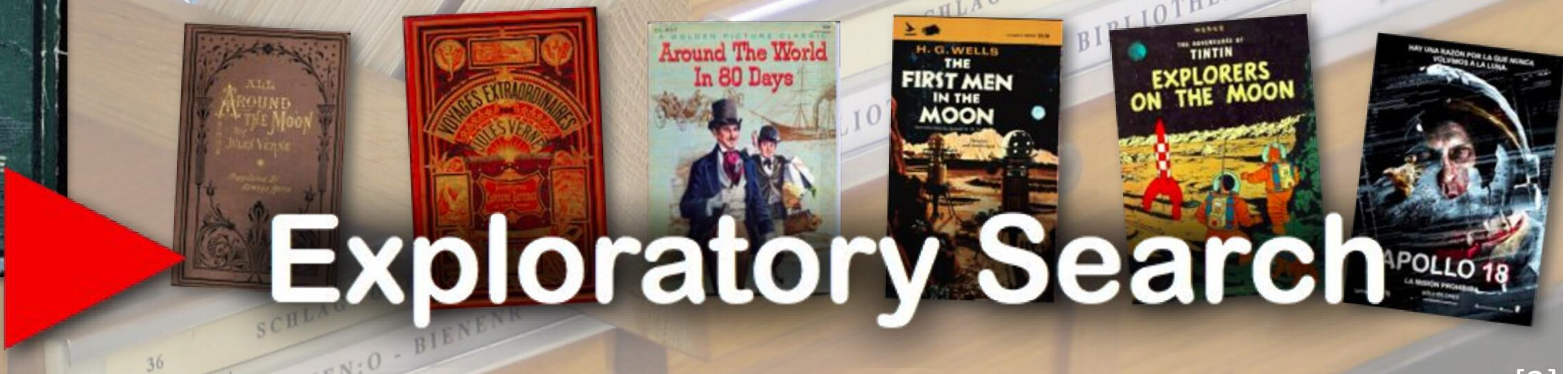
The Retrieval Problem



[2]

Retrieval vs. Exploration

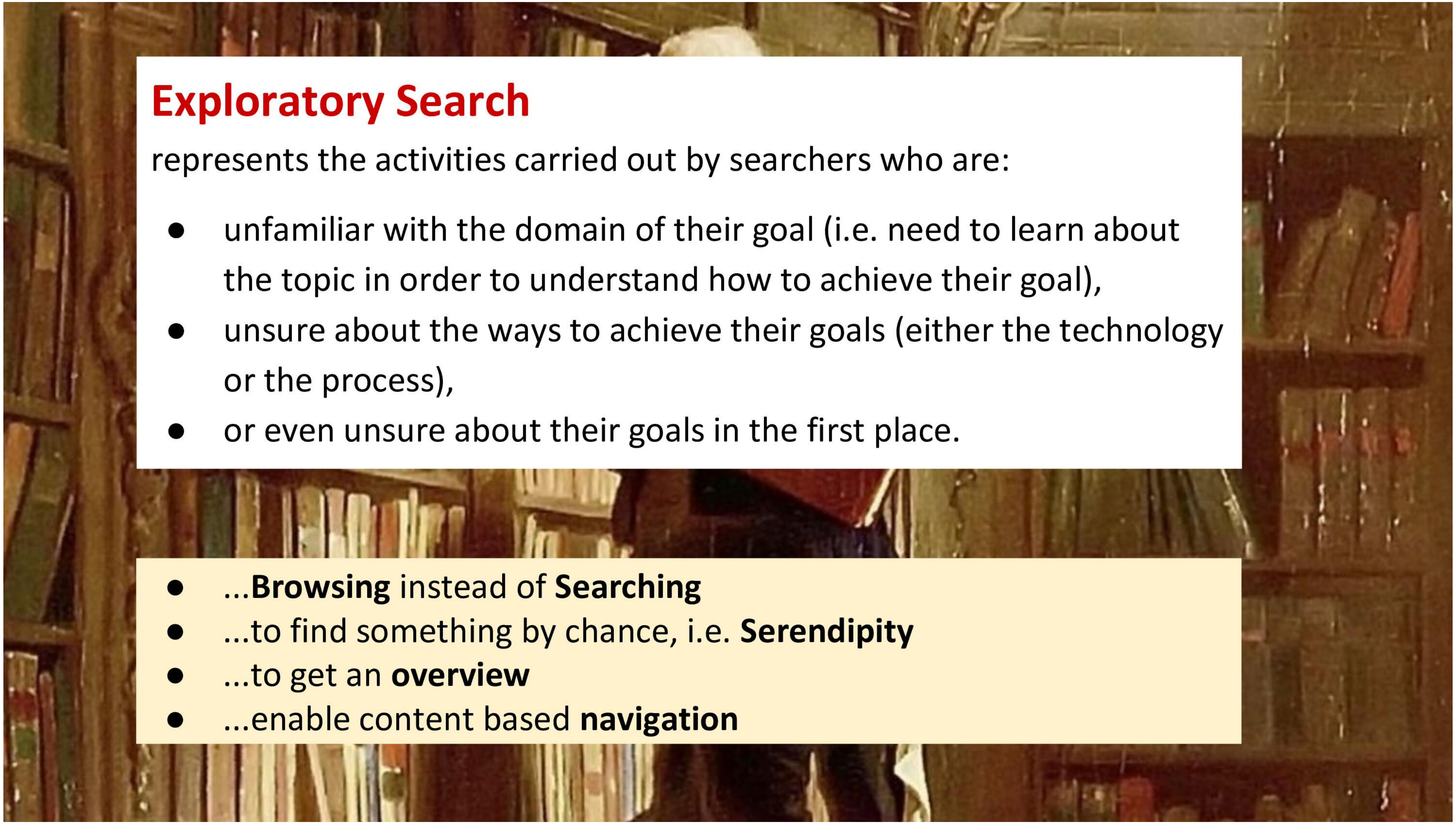
- *Find another („comparable“) book,*
  (that will be of interest for me...)
- *Find books of the same or of related topics*
- *How did the author / the topic develop over time?*
- *What else would I like to read?*
- *...*



[2]

Slide from [Knowledge Graphs course]() by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).



Exploratory Search

represents the activities carried out by searchers who are:

- unfamiliar with the domain of their goal (i.e. need to learn about the topic in order to understand how to achieve their goal),
- unsure about the ways to achieve their goals (either the technology or the process),
- or even unsure about their goals in the first place.

- ...**Browsing** instead of **Searching**
- ...to find something by chance, i.e. **Serendipity**
- ...to get an **overview**
- ...enable content based **navigation**

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Exploratory Search via Knowledge Graphs

http://dbpedia.org/resource/From_the_Earth_to_the_Moon

DBpedia   Browse using ▾   Formats ▾   Faceted Browser   Sparql Endpoint

About: From the Earth to the Moon

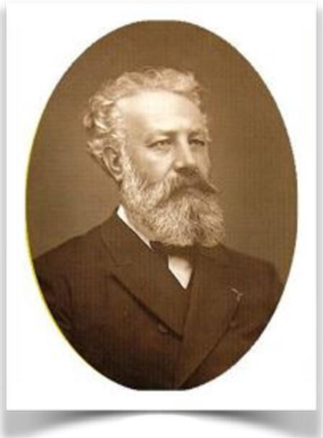
An Entity of Type : work, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

From the Earth to the Moon (French: De la terre à la lune) is an 1865 novel by Jules Verne.

Property	Value
dbo:abstract	• Von der Erde zum Mond ist ein Roman des französischen Autors Jules Verne. Der Roman wurde erstmals 1865 unter dem französischen Titel De la Terre à la Lune von dem Verleger Pierre-Jules Hetzel veröffentlicht. Die erste deutschsprachige Ausgabe erschien 1873 unter dem Titel Von der Erde zum Mond. Der englische Titel des Romans lautet From the Earth to the Moon.Es handelt sich um ein frühes Werk des Science-Fiction-Genres, das die Mondfahrt um etwa hundert Jahre vorwegnimmt. Allerdings geht es hier vor allem noch um die Vorbereitung des Abenteuers. Der Roman Reise um den Mond (Autour de la Lune) von 1870 setzte die Geschichte fort. (de) • From the Earth to the Moon (French: De la terre à la lune) is an 1865 novel by Jules Verne. It tells the story of the Baltimore Gun Club, a post-American Civil War society of weapons enthusiasts, and their attempts to build an enormous sky-facing Columbiad space gun and launch three people—the Gun Club's president, his Philadelphian armor-making rival, and a French poet—in a projectile with the goal of a moon landing.The story is also notable in that Verne attempted to do some rough calculations as to the requirements for the cannon and, considering the comparative lack of any data on the subject at the time, some of his figures are surprisingly close to reality. However, his scenario turned out to be impractical for safe manned space travel since a much longer muzzle would have been required to reach escape velocity while limiting acceleration to survivable limits for the passengers.The character of Michel Ardan, the French member of the party in the novel, was inspired by the real-life photographer Félix Nadar. (en)
dbo:author	• dbr:Jules_Verne
dbo:illustrator	• dbr:Henri_de_Montaut
dbo:literaryGenre	• dbr:Science_fiction
dbo:mediaType	• dbr:Hardcover
dbo:publisher	• dbr:Pierre-Jules_Hetzel
dbo:series	• dbr:Voyages_extraordinaires
dbo:thumbnail	• wiki-commons:Special:FilePath/From_the_Earth_to_the_Moon_Jules_Verne.jpg?width=300

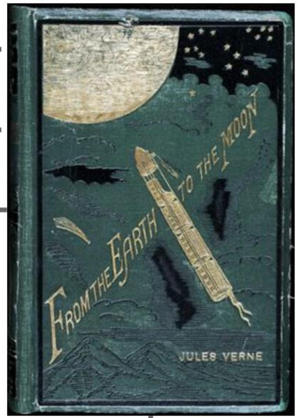
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Exploratory Search via Knowledge Graphs



:Jules_Verne

:From_the_Earth_to_the_Moon



rdf:type → dbo:Book

dbo:author

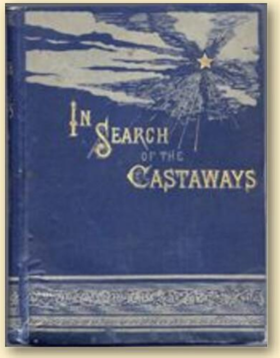
dct:subject

dbo:influenced

:H._G._Wells

category:1865_novels
category:Frence_science_fiction_novels
category:Novels_by_Jules_Verne
category:Moon_in_fiction
category:Fictional_rivalries
category:Novels_set_in_Florida
category:1860s_science_fiction_novels
...

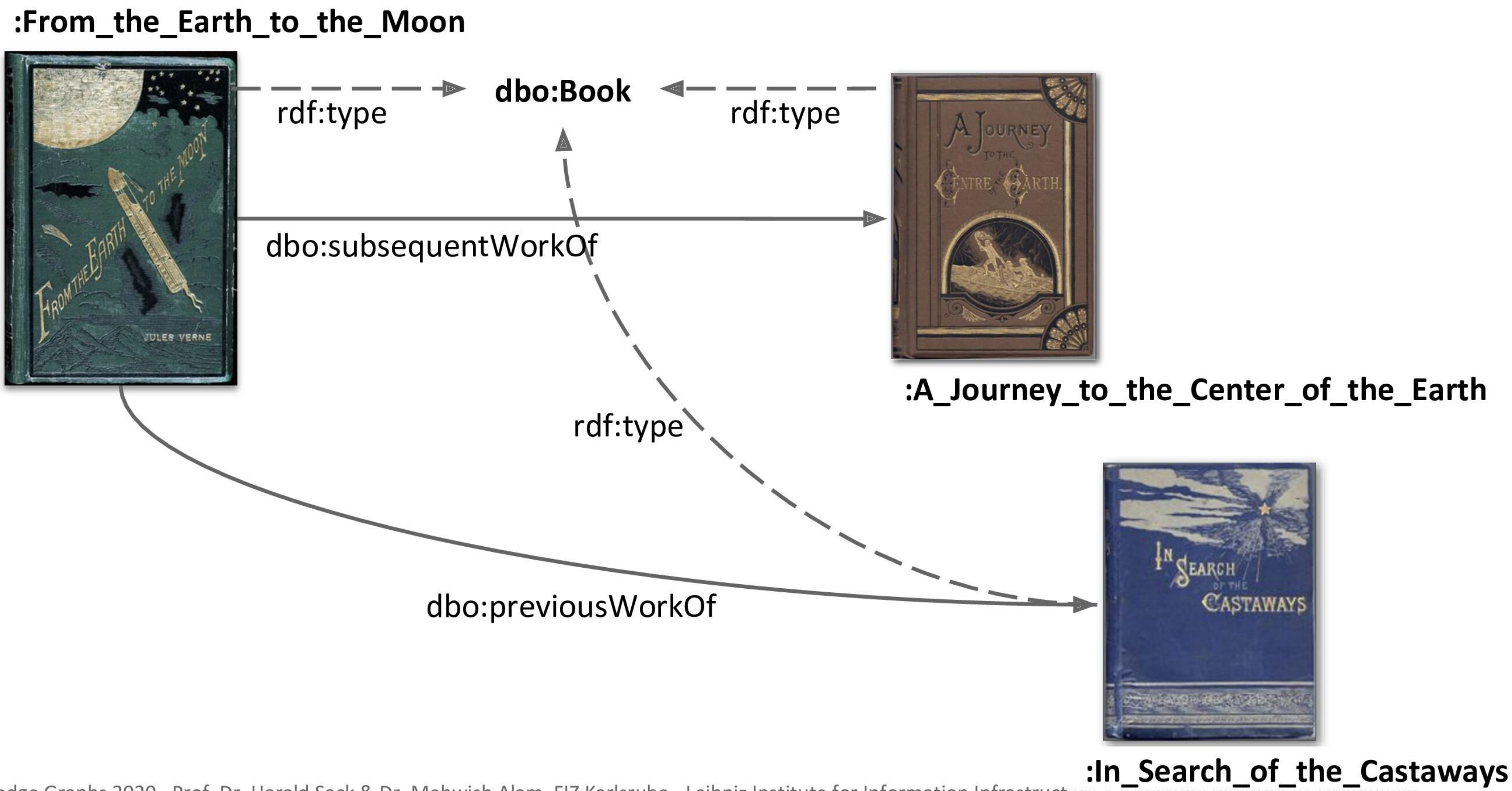
dbo:previousWorkOf



:In_Search_of_the_Castaways

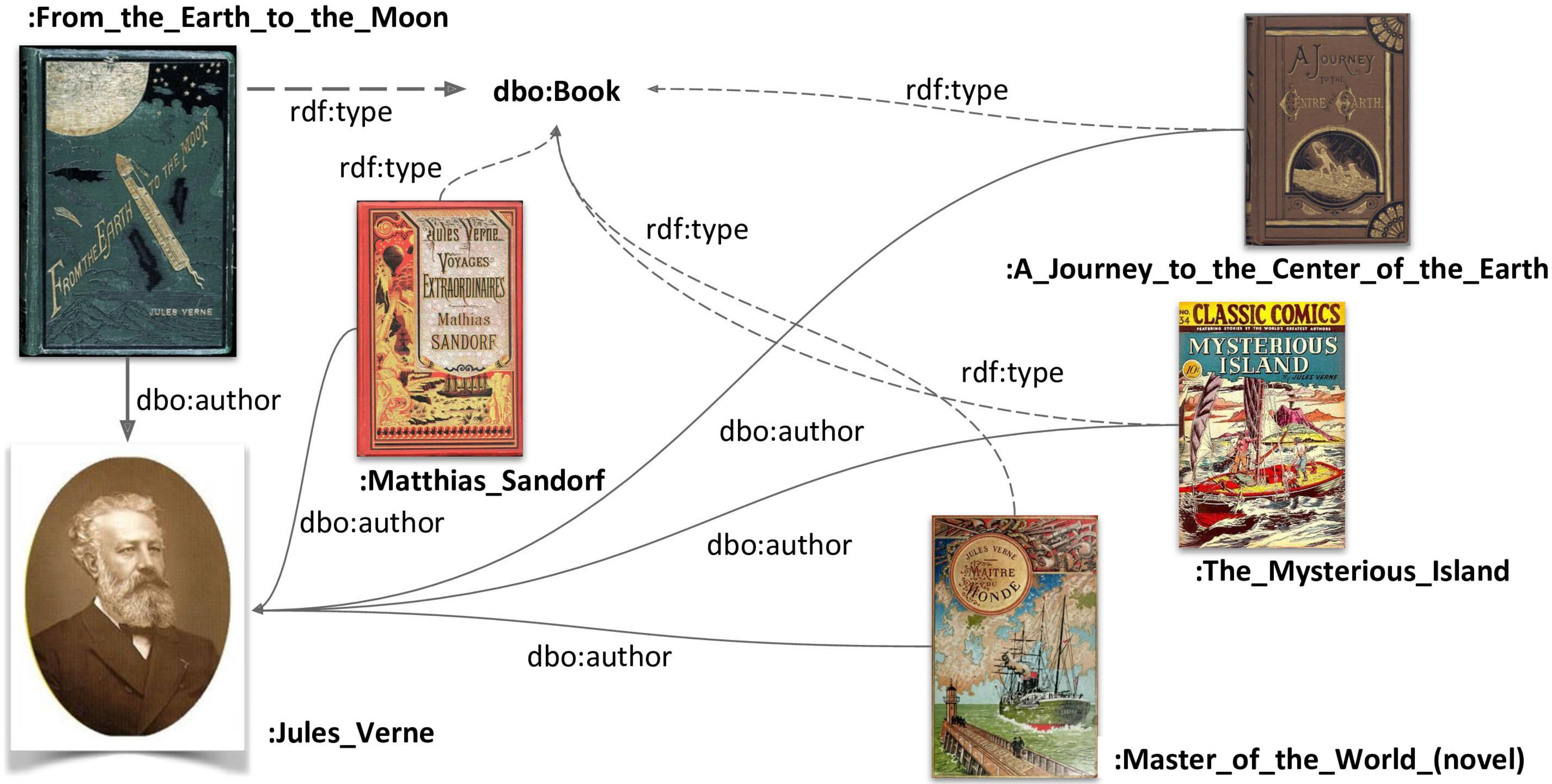
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Exploratory Search via Knowledge Graphs



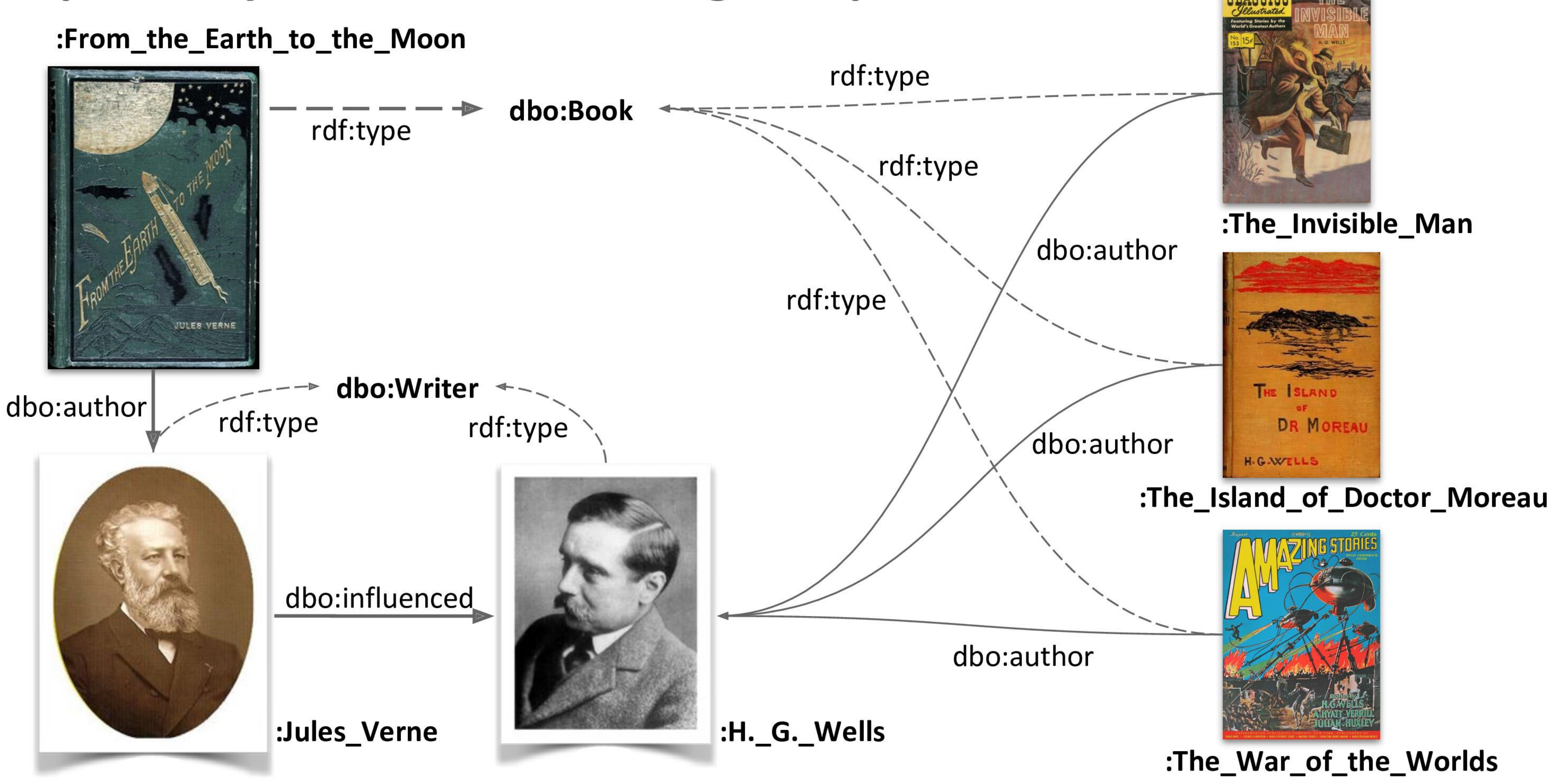
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Exploratory Search via Knowledge Graphs



Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Exploratory Search via Knowledge Graphs



Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Exploratory Search via Knowledge Graphs

- **Exploratory Search** represents the activities carried out by searchers who are either:
    - **unfamiliar with the domain** of their goal (i.e. need to learn about the topic in order to understand how to achieve their goal),
    - **unsure about the ways** to achieve their goals (either the technology or the process)
    - or even **unsure about their goals** in the first place.

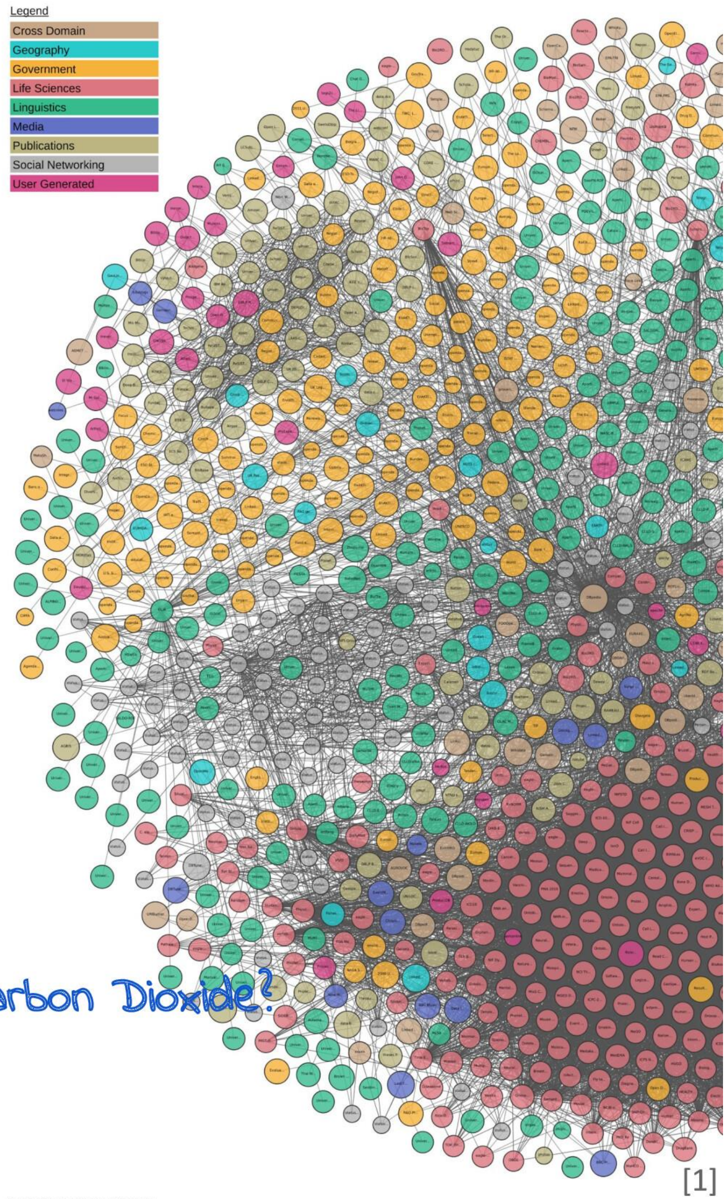
- **Recommender Systems** seek to predict the preference a user would give to an item.

KNOWLEDGE GRAPH EMBEDDINGS

The graphs are vectors
if you need it

Semantic Similarity

- For search and retrieval systems, **semantic similarity of entities** is an important feature, as e.g.
    - Given an entity find the most similar entities
    - Given an entity find the most similar documents
    - Given a document find the most similar documents, etc.
- **When are two entities (semantically) similar?**
    - If they can be described by the same/similar facts, as e.g.
    - Carbon Dioxide is a Greenhouse Gas and water vapour is a Greenhouse Gas
    - Albert Einstein is a Physicist and Stephen Hawking is a Physicist
    - Is Stephen Hawking more similar to Albert Einstein or to Carbon Dioxide?

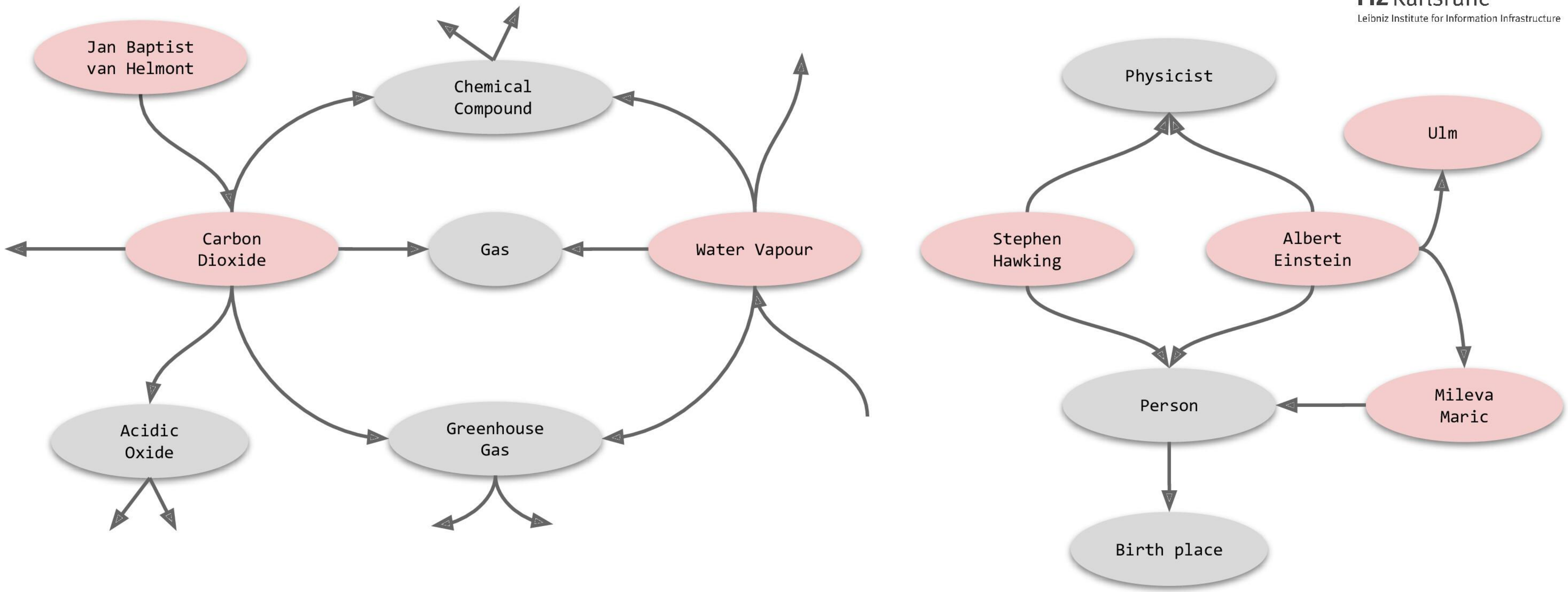


[1]

The Linked Open Data Cloud from lod-cloud.net

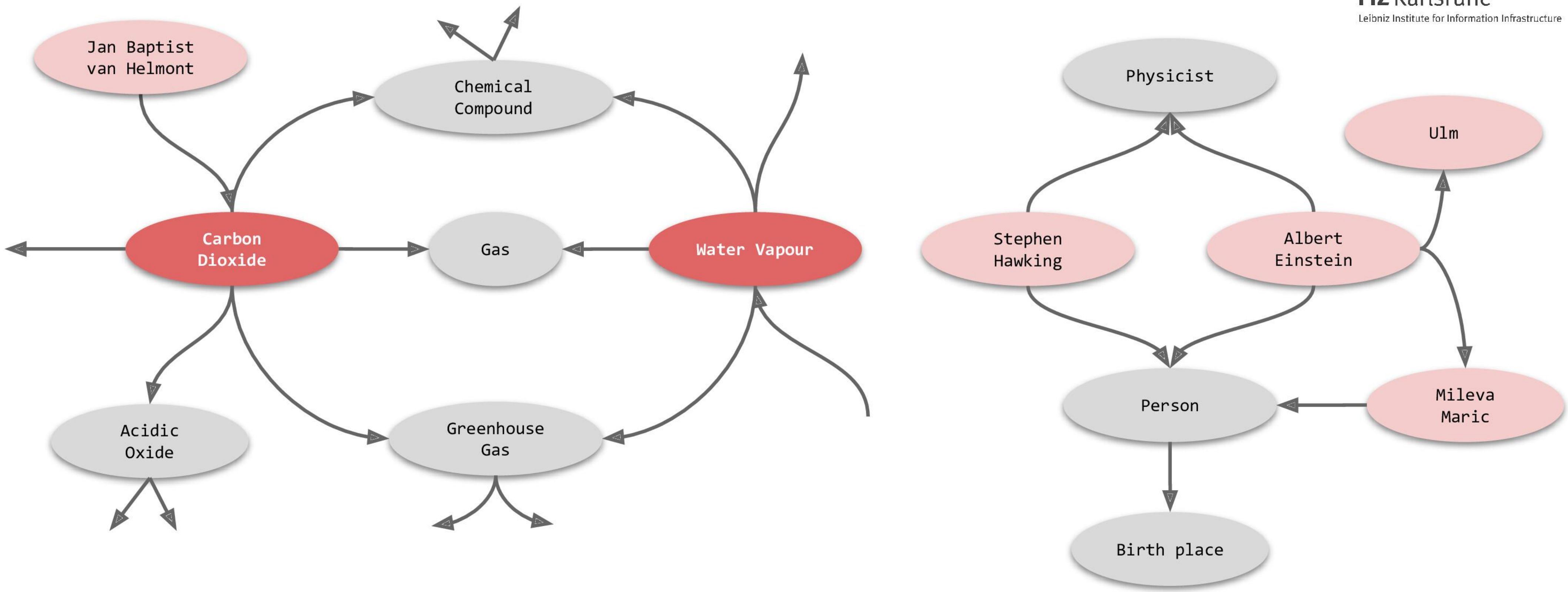
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Semantic Similarity



Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

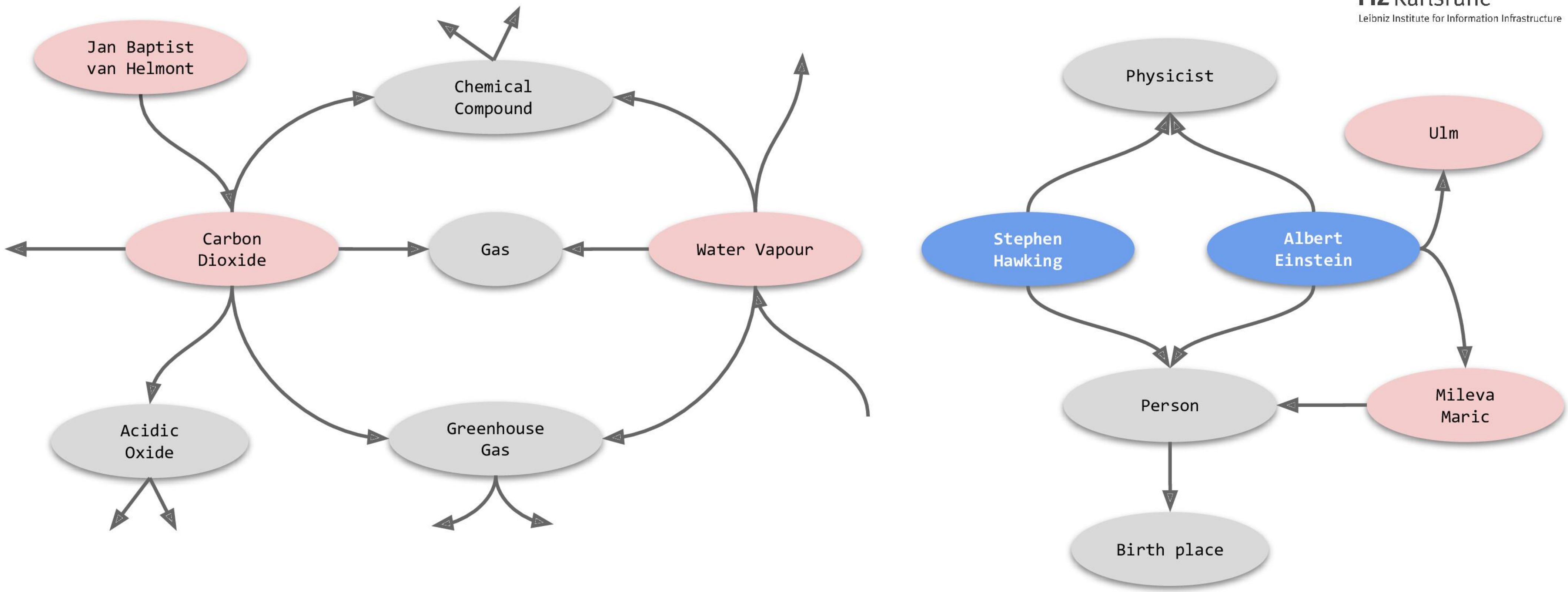
Semantic Similarity



- Carbon Dioxide and water vapour share similar (structural) context in the graph

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

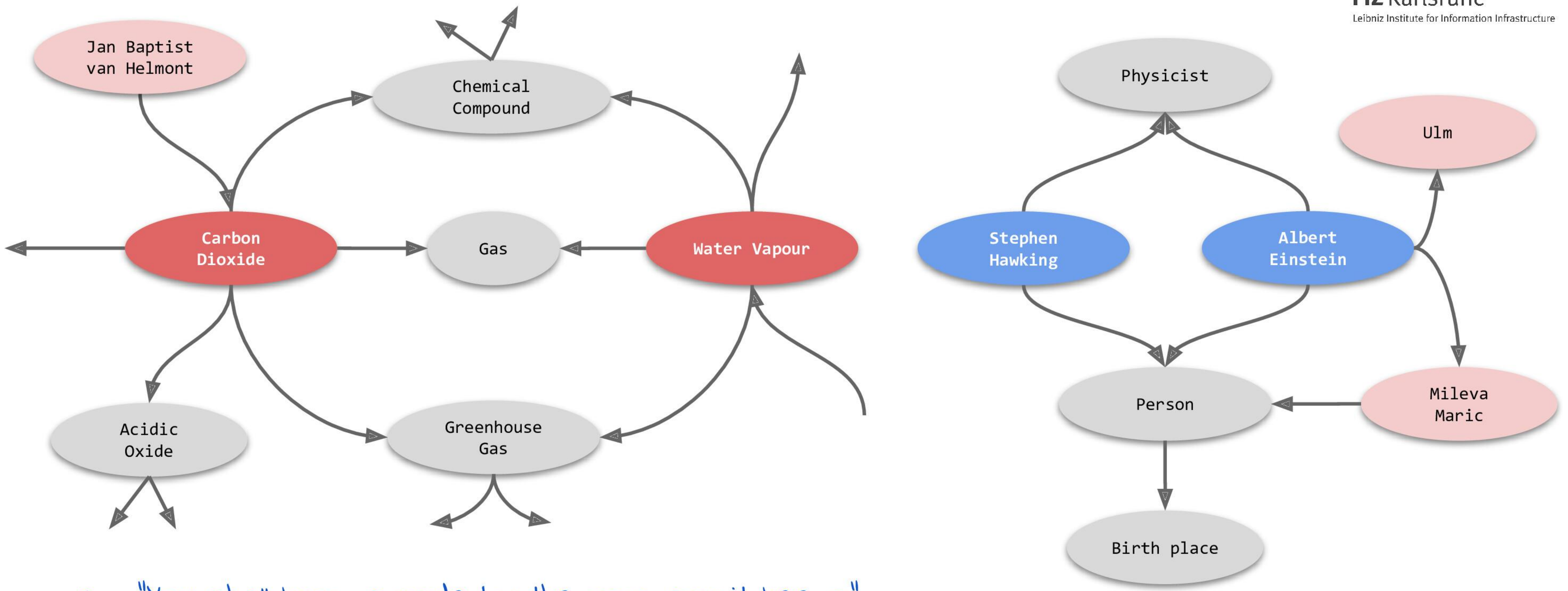
Semantic Similarity



- Stephen Hawking and Albert Einstein share similar (structural) context in the graph

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Semantic Similarity



- "You shall know a node by the company it keeps"
- i.e. similar nodes can be identified by having the same/similar environment (context)
- adjacency based similarity

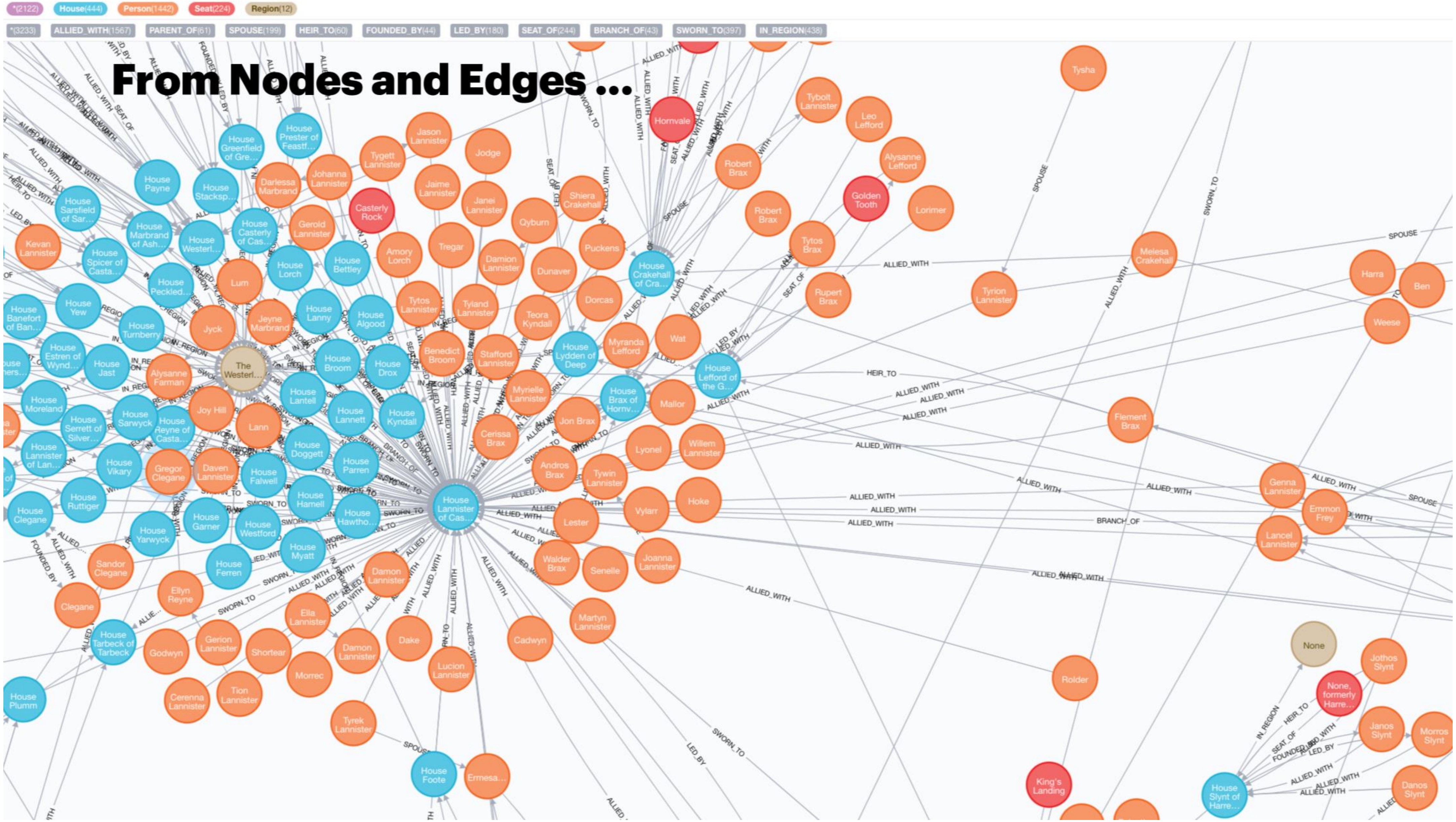
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Semantic Similarity

- In a Knowledge Graph,
  - **similar entities** are represented by nodes that are connected to **similar/same facts**
  - i.e. that are connected to **similar graph structures**
  - To identify **similar entities**, we have to identify **similar graph structures**
- **Problem:**
  - Algorithms to determine semantic similarity in graphs are of high complexity, i.e. with large KGs, as e.g. Wikidata, they don’t work efficiently.
- **Idea**:
  - Approximate the problem by transferring it from graph structures to vector spaces That are easier to handle.

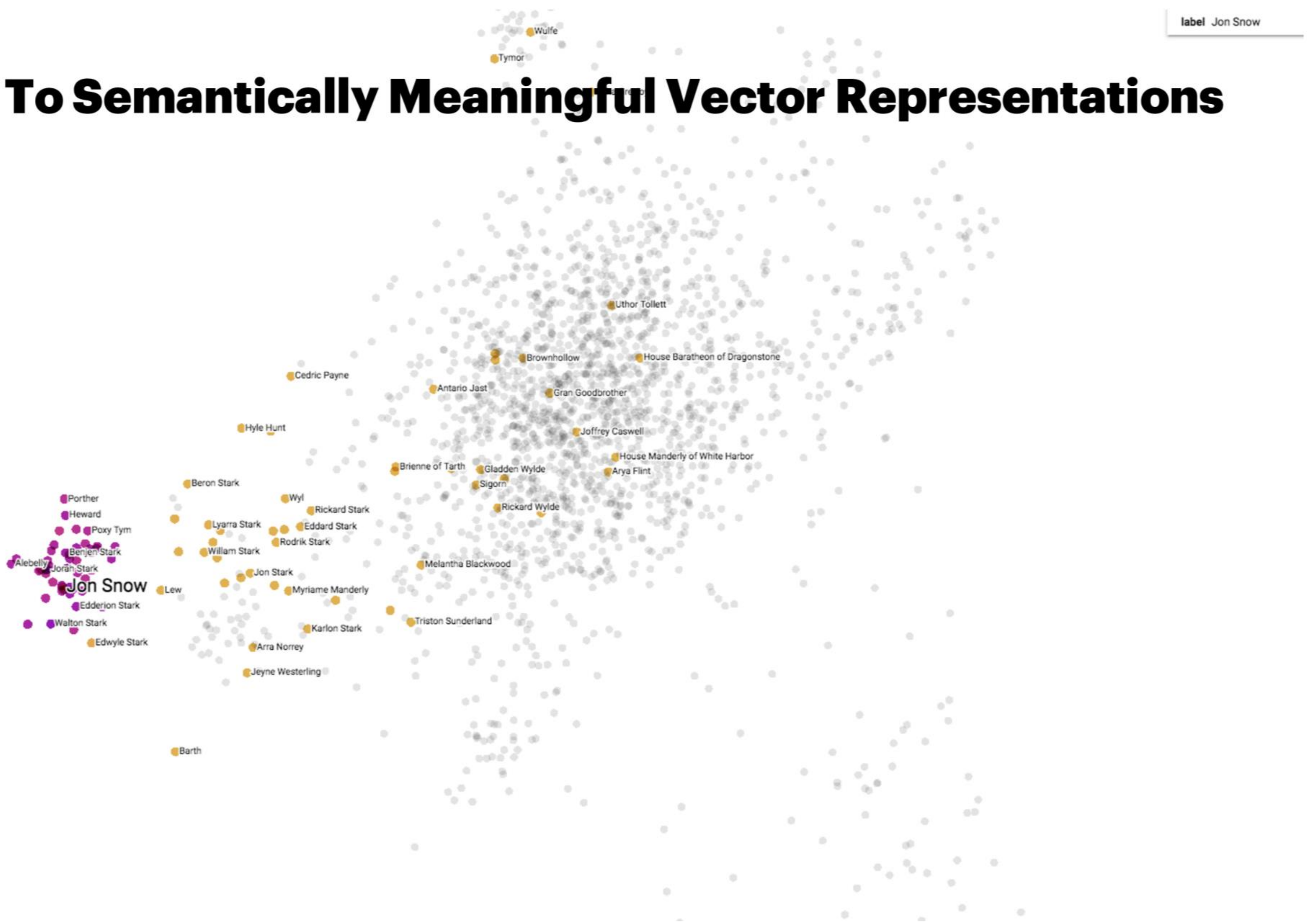
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

From Nodes and Edges ...



Slide from Knowledge Graph Embeddings Tutorial by L. Costabello et al. (ECAI 2020).

… To Semantically Meaningful Vector Representations



Slide from Knowledge Graph Embeddings Tutorial by L. Costabello et al. (ECAI 2020).

Excursion: Word Embeddings

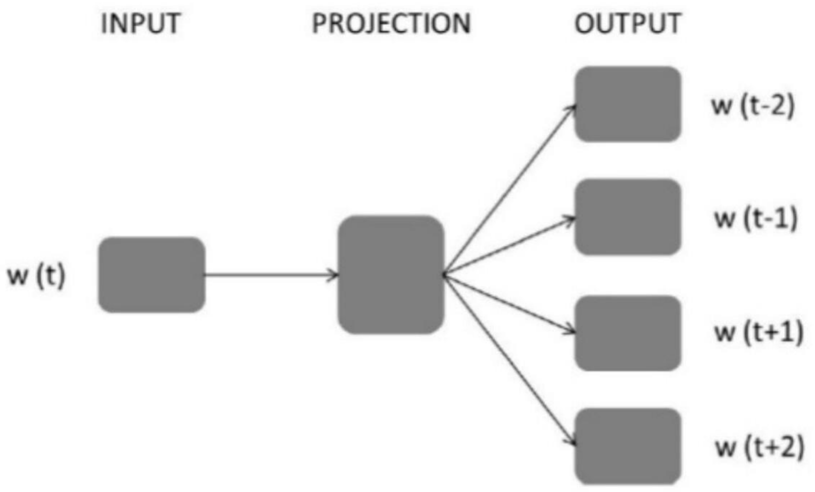
- **Word Embeddings** map natural language words to a dense vector representation
- **Basic Assumption:** Similar words occur in similar contexts:
  (Carbon Dioxide, Water Vapour, Methane) is one of the driving agents of climate change.
  Climate change is caused by greenhouse gases like (Carbon Dioxide, Water Vapour, Methane)
- **Basic idea:** instead of counting co-occurrences of words, predict the likelihood of the appearance of words in the neighborhood of others
- Train a predictor (neural network) that can predict a word from its context (**CBOW**) or the context from a given word (**Skip Gram**)

Mikolov, Tomas; et al. (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

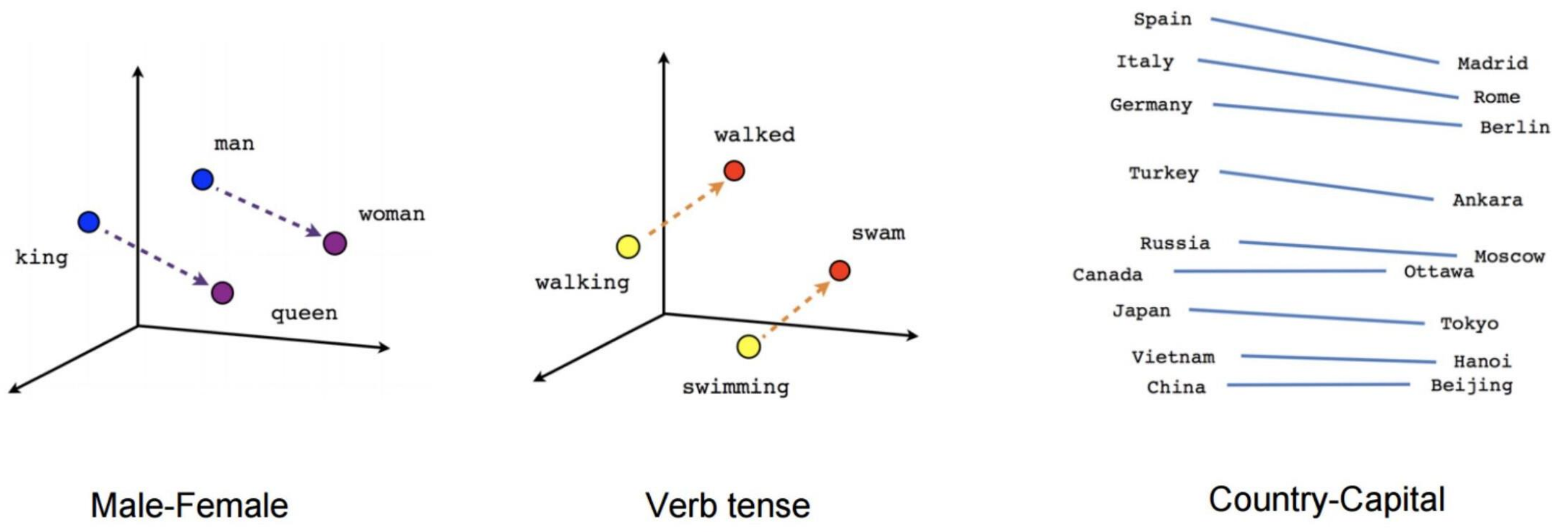
Excursion: Word Embeddings

- **Skip Gram:**
  - Train a neural network with one hidden layer
  - Use output at hidden layer as vector representations
- **Observation:**
  - *Carbon Dioxide, Water Vapour, Methane* will activate similar context words
  - i.e. their output weights at the projection layer have to be similar



Mikolov, Tomas; et al. (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781

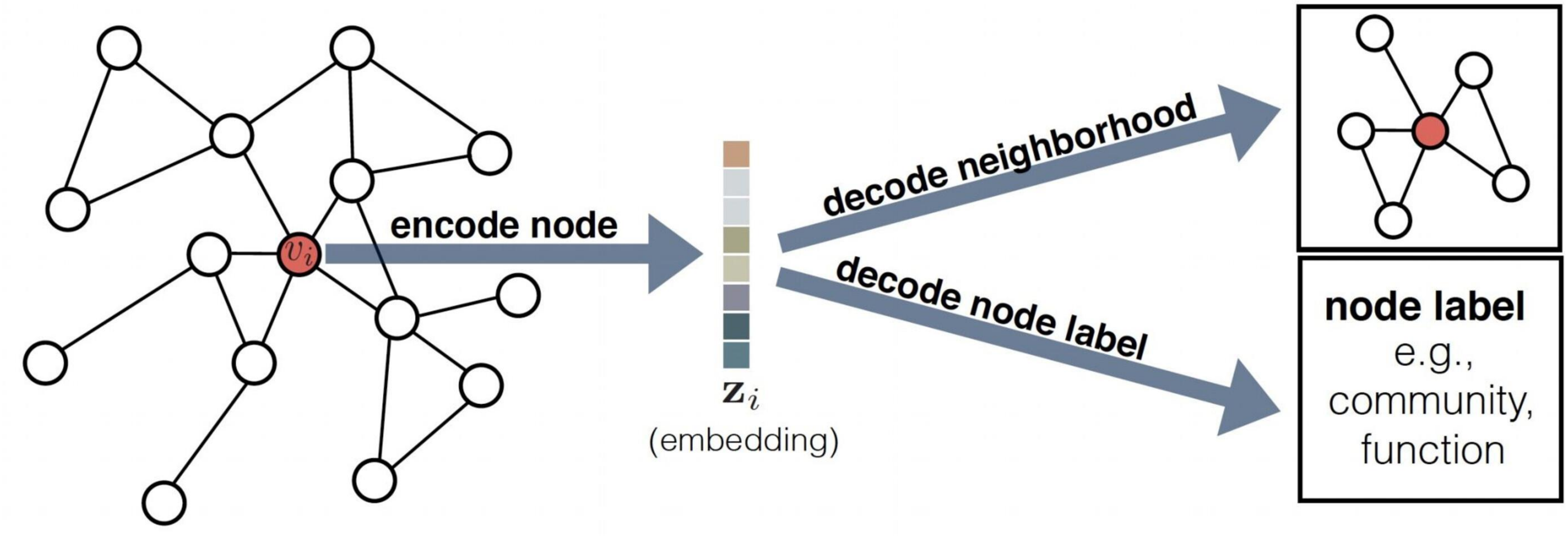
Word Embeddings



- Semantics of words is preserved, i.e. it enables semantic arithmetic operations as e.g. analogies
    - “king” - “man” ≈ “queen” - “woman”
    - “king” - “man” + “woman” ≈ “queen”

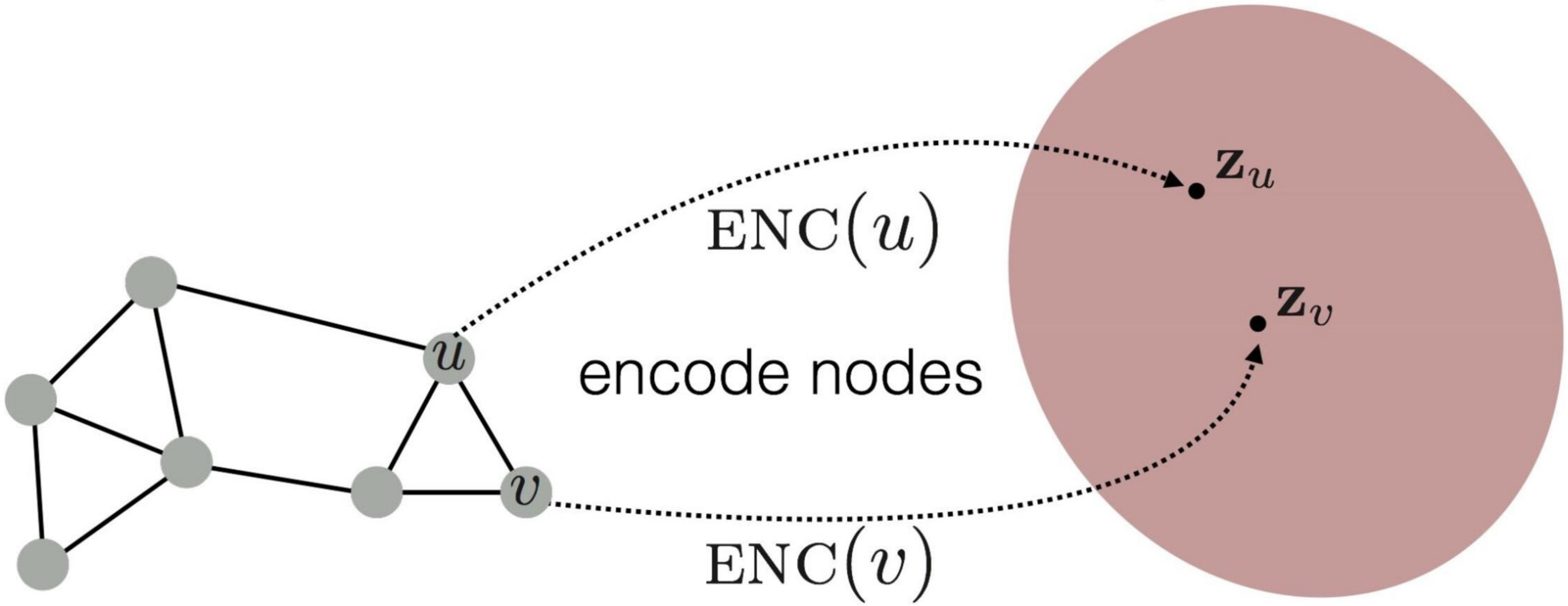
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Graph Embeddings



Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Graph Embeddings - Encoder-Decoder Approach



- The goal is to encode the nodes of the graph in a way so that **similarity in the embedding space** (e.g., dot product) **approximates similarity in the original network**.
- ENC: $N \rightarrow \mathbb{R}^d$, $u,v \in N$, $\mathrm{ENC}(u) = z_u \in \mathbb{R}^d$, $\mathrm{ENC}(v) = z_v \in \mathbb{R}^d$
- DEC: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$, $\mathrm{DEC}(\mathrm{ENC}(u), \mathrm{ENC}(v)) = \mathrm{DEC}(z_v , z_u) \approx$ similarity (u,v)

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Learning Graph Embeddings

1) Define an **encoder ENC** (i.e., a mapping from nodes to embeddings)
2) Define a **node similarity function** that specifies how relationships in vector space map to relationships in the original network.
3) Optimize the parameters of the encoder so that:

$$\text{similarity}(u, v) = z_v^T z_u$$

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Knowledge Graph Embeddings

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

Many ways to generate Knowledge Graph Embeddings:

- **Translational Methods**: TransE, TransH, TransR, TransEdge, …
- **Rotation Based**: RotatE
- **Graph Convolutional Networks**: R-GCN, TransGCN
- **Walk-Based Methods**: DeepWalk, RDF2Vec

Slide from [Knowledge Graphs course] by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

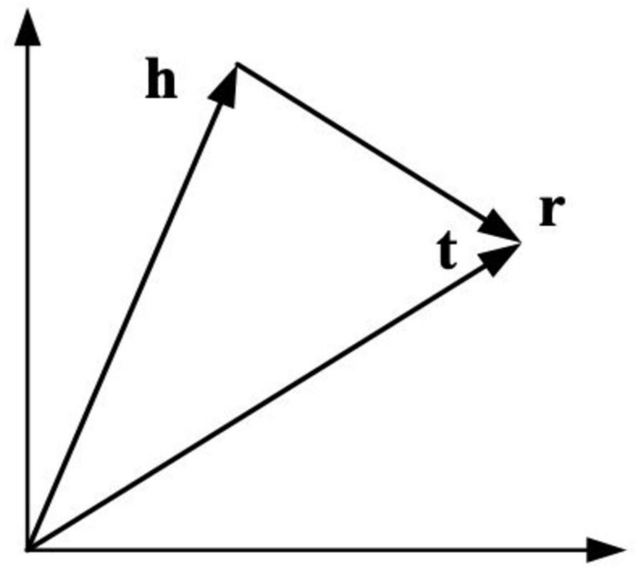
Translational Distance Models

- Exploit distance-based scoring functions
- Measure the **plausibility of a fact** as the **distance between two entities**
- A translation carried out by the relation.
- **Models**: TransE, TransH, TransR, TransD, TransSparse, TransM, TransEdge

Wang et al., Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2017.

TransE

- Entities and relations are embedded into **same vector space**.
- h = head, t = tail, r = relation
- Relation r is considered as translation from h to t
- Learning Assumption **h+r≈t**
- **Problem:** Symmetric functions,
  1-N / N-1 / N-N functions

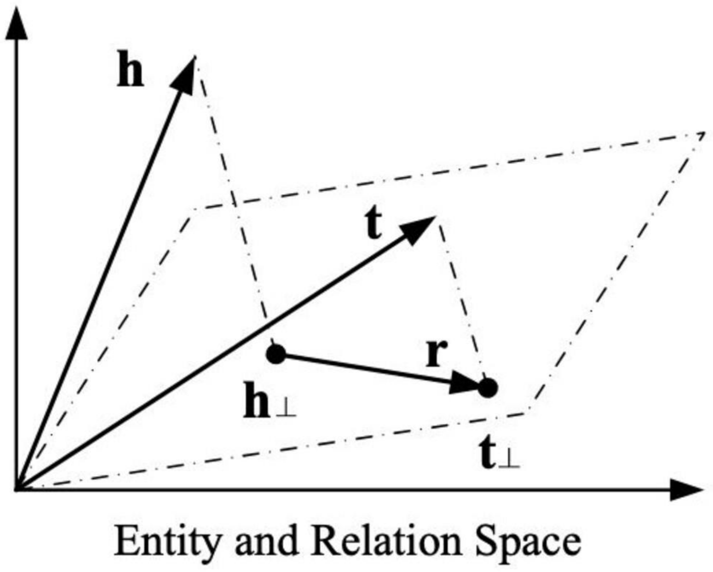


Bordes et al, Translating Embeddings for Modeling Multi-relational Data, NIPS 2013

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

TransH

- From original space to Hyperplane
- TransH enables **different roles of an entity in different relations**.
- Entities h and t are projected into specific **hyperplane of relation r**.
- Then predict new links based on translation on hyperplane.



Wang et al., Knowledge graph embedding by translating on hyperplanes. AAAI, 2014.

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Graph Convolutional Network

- **Graph Convolutional Networks (GCN)**
  - modeling structured neighborhood information of **unlabeled** and **undirected** graphs with **convolution operations**
- **Relational Graph Convolutional Network (R-GCN)**
  - Models Relational Data using GCN where Knowledge Graphs are considered as **directed labeled multigraphs**.
  - Models in RGCN
    - **Link Prediction:**
      - **an encoder:** an R-GCN producing latent feature representations of entities,
      - **a decoder:** a tensor factorization model exploiting these representations to predict labeled edges

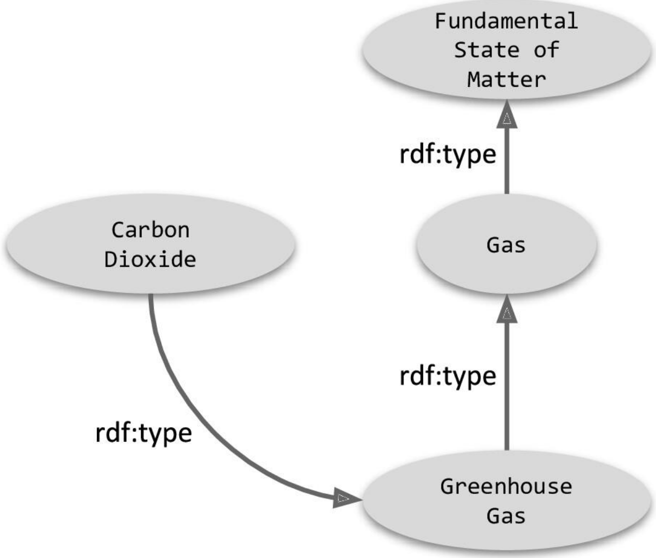
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

RDF2Vec

- Word2vec operates on sentences, i.e. sequences of words
- **RDF2Vec Basic Idea**:
  - Generate “sentences” from knowledge graph, i.e. sequences of interconnected RDF triples

```
:CarbonDioxide rdf:type :GreenhouseGas.
:GreenhouseGas, rdf:type, :Gas.
:Gas, rdf:type, :FundamentalStateOfMatter.
```

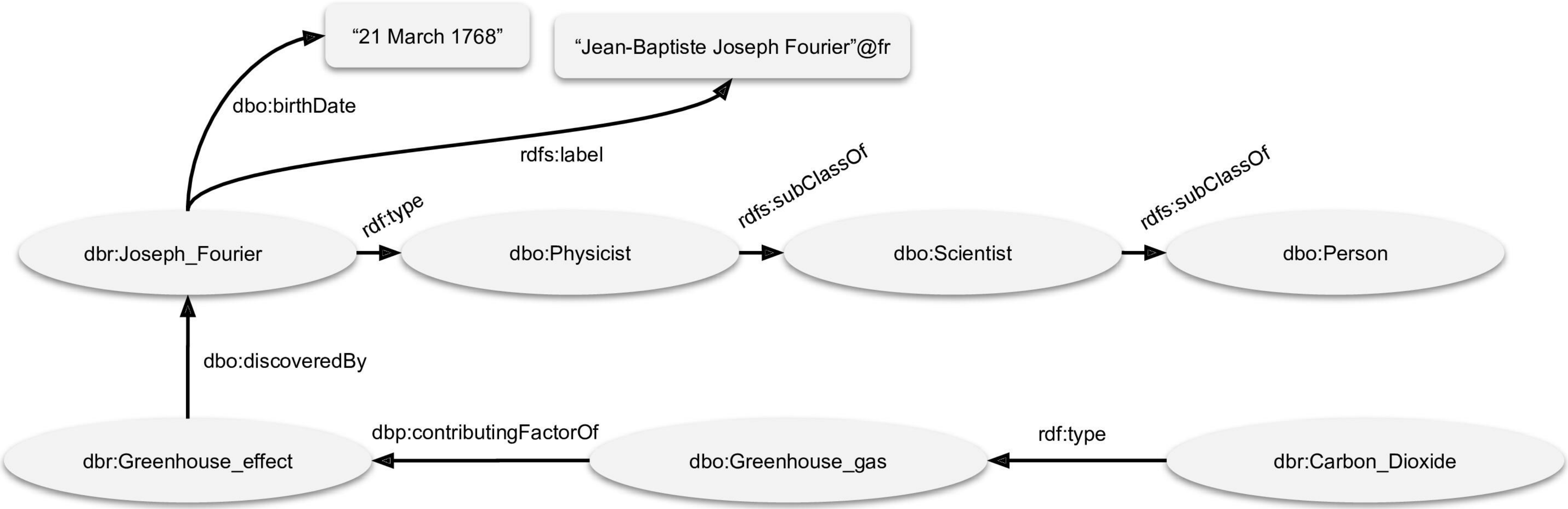
  - Selection strategies:
    - Depth first search
    - Breadth first search
    - Random walk
    - RDF Graph Kernels



Petar Ristoski and Heiko Paulheim RDF2Vec: RDF graph embeddings for data mining, ISWC 2016

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Graph Walks RDF2Vec



Generated Sequences of depth = 3:

- dbr:Carbon_Dioxide→ rdf:type→dbo:Greenhouse_gas → dbp:contributingFactorOf → dbr:Greenhouse_effect → dbo:discoveredBy → dbr:Joseph_Fourier

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Libraries for KG Embedding

PyTorch BigGraph

https://github.com/facebookresearch/PyTorch-BigGraph

AmpliGraph

https://github.com/Accenture/AmpliGraph

PyKeen

https://github.com/SmartDataAnalytics/PyKEEN

OpenKE

http://openke.thunlp.org/

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

KNOWLEDGE GRAPH COMPLETION

How to guess the missing triples?

Knowledge Graph Refinement

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- As a model of the real world or a part of it, **knowledge graphs cannot reasonably reach full coverage**, i.e., contain information about each and every entity in the universe.
- **It is unlikely,** in particular if heuristic methods are applied for knowledge graph construction, **that the knowledge graph is fully correct**.
- To address those shortcomings, various methods for **Knowledge Graph Refinement** have been proposed, as e.g.
  - Deduplicating entity nodes (entity resolution)
  - Collective reasoning (probabilistic soft logic)
  - **Link prediction** or **Knowledge Graph Completion**
  - Dealing with missing values
  - Anything that improves an existing knowledge graph

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Completion vs. Error Detection

- **Knowledge Graph Completion:**
  Adding missing knowledge to the Knowledge Graph

  E.g. adding a triple:
  <JosephFourier, occupation, Physicist>

- **Error Detection:**
  Identifying wrong information in the Knowledge Graph

  E.g. finding inconsistencies:
  <JosephFourier, isA, Human>
  <JosephFourier, isA, FictionalCharacter>

Slide from [Knowledge Graphs course] by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Knowledge Graph Completion

- A promising approach for **Knowledge Graph Completion** is
  - to embed Knowledge Graphs into latent spaces (via Knowledge Graph Embeddings) and
  - make inferences by learning and operating on latent representations.
- Such embedding models, however, **do not make use of any rules** during inference and hence have limited accuracy.
- E.g. predict that in wikidata the following fact may be complemented:

  (AtsumoOmuhura occupation Climatologist)

  `wd:Q462297 wdt:P106 wd:Q1113838 .`

  Tail Prediction

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Link Prediction

	Task	Example	Result
Link Prediction	Triple Classification	(JosephFourier, occupation, physicist)?	(yes, 95%)
	Tail Prediction	(JosephFourier, occupation, ?)	(1, physicist, 0.95), (2, chemist, 0.93) ...
	Head Prediction	(?, occupation, physicist)	(1, AlbertEinstein, 0.91) (2, StephenHawking, 0.90)
	Relation Prediction	(JosephFourier, ?, physicist)	(1, occupation, 0.95)
	Entity Classification (Type Prediction)	(JosephFourier, isA, ?)	(1, Person, 0.99) (2, Human, 0.99),...

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Type Prediction

- **Predicting a type or class** for an entity given some characteristics of the entity is a very common problem in machine learning, known as **classification**.

  `<JosephFourier, isA, ?>`

- **Supervised Learning Approach**:
  - Type Prediction can be addressed via a **classification model** based on **labeled training data**,
  - typically the set of entities in a Knowledge Graph which have types attached.

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

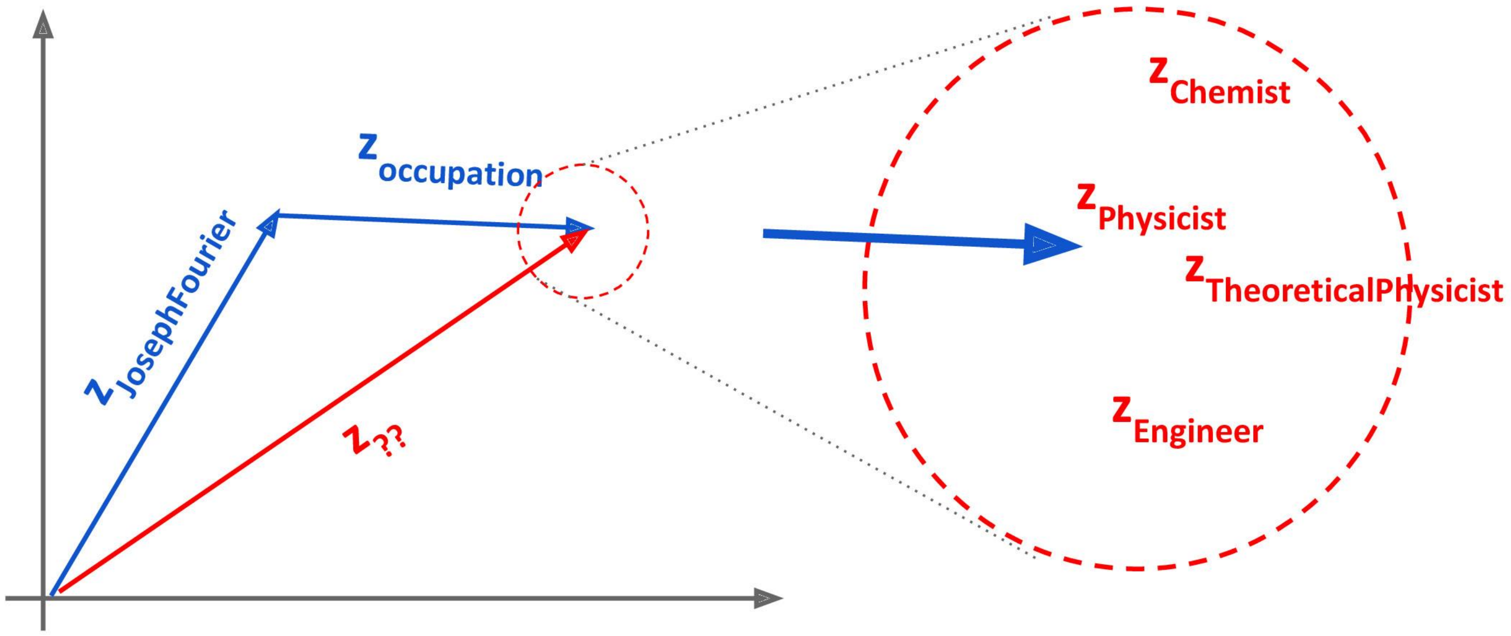
Type Prediction

- **Multi-Class Prediction:**
  - In Knowledge Graphs usually there are more than two types/classes of entities to distinguish
    E.g. Classes Physicists, Chemists, Climatologists, etc.
- **Single-Label Classification:**
  - Only one type/class can be assigned per entity
    E.g.: `<JosephFourier, isA, Person>`
- **Multi-Label Classification:**
  - In Knowledge Graphs some entities might allow for the assignment of more than one type
    E.g.: `<electron, isA, Particle>` and
    `<electron, isA, Wave>`

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Methods for Knowledge Graph Link Prediction

- Use **Translational Embeddings**
  - **Unsupervised** methods, e.g. **TransE**, use $z_s + z_p$ to predict z_o
  - **Supervised** Methods for prediction based on embedding vectors



Ranked List for z_o:

1. Physicist, *0.95*
2. TheoreticalPhysicist, *0.89*
3. Chemist, *0.83*
4. Chemist, *0.62*
5. ...

score

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Industrial applications:

Pharmaceutical Industry:
Drug Side-effects Prediction

Human Resources:
Career Paths Prediction

Products:
Product Recommendation

Food & Beverage:
Flavor Combinations

Slide from Knowledge Graph Embeddings Tutorial by L. Costabello et al. (ECAI 2020).

CHATGPT IS A BULLSHIT

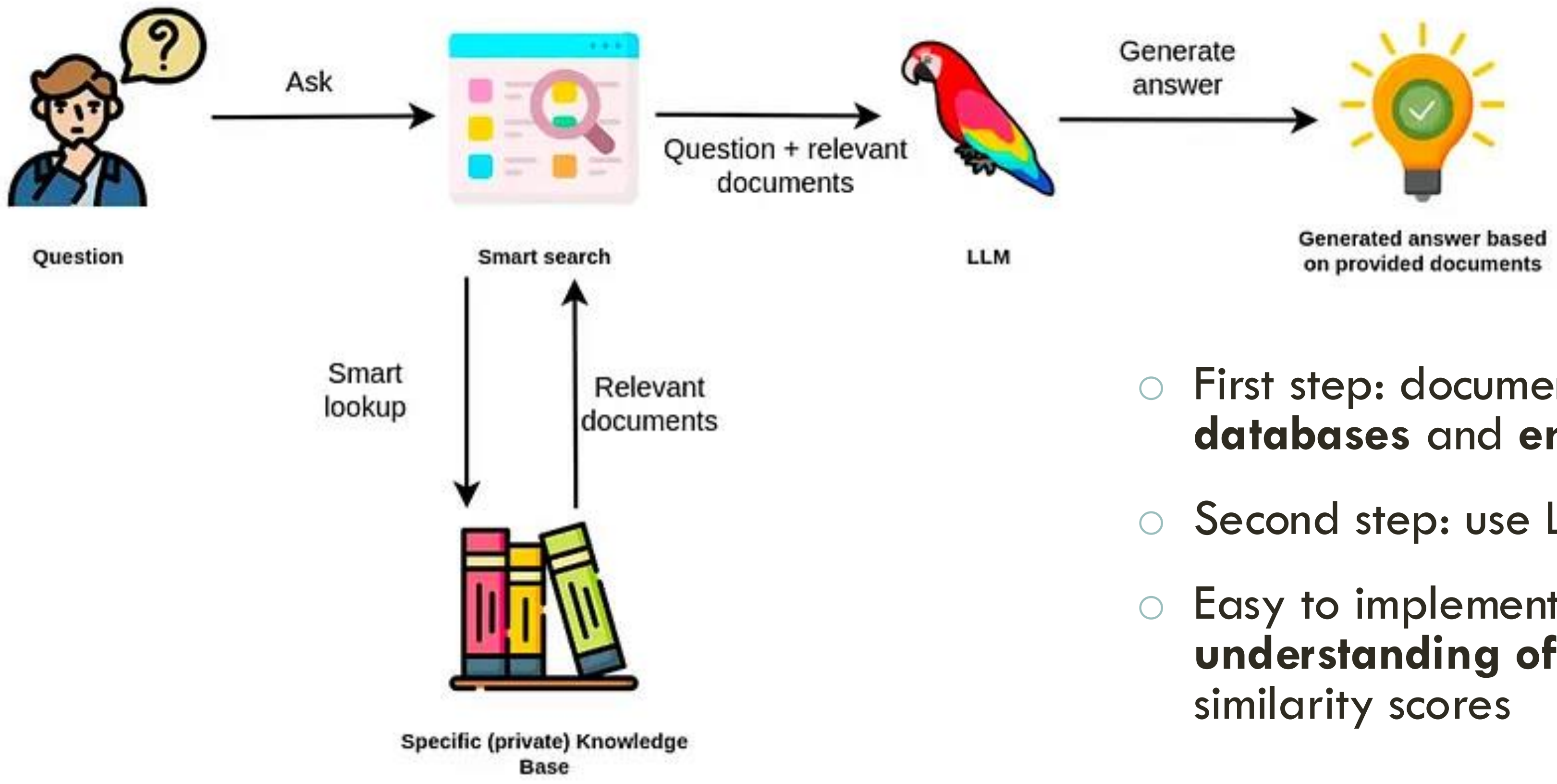
How can we fix it?

IT'S NOT ABOUT HALLUCINATIONS…

We argue against the view that when ChatGPT and the like produce false claims they are lying or even hallucinating, and in favour of the position that the activity they are engaged in is bullshitting, in the Frankfurtian sense (Frankfurt, 2002, 2005). Because ***these programs cannot themselves be concerned with truth****, and because they are designed to produce text that looks truth-apt* ***without any actual concern for truth****, it seems appropriate to call their outputs* ***bullshit.***

Source: M.T. Hicks, J. Humphries & J. Slater (2024), ChatGPT is bullshit, Ethics Inf Technol, 26, pp. 38.

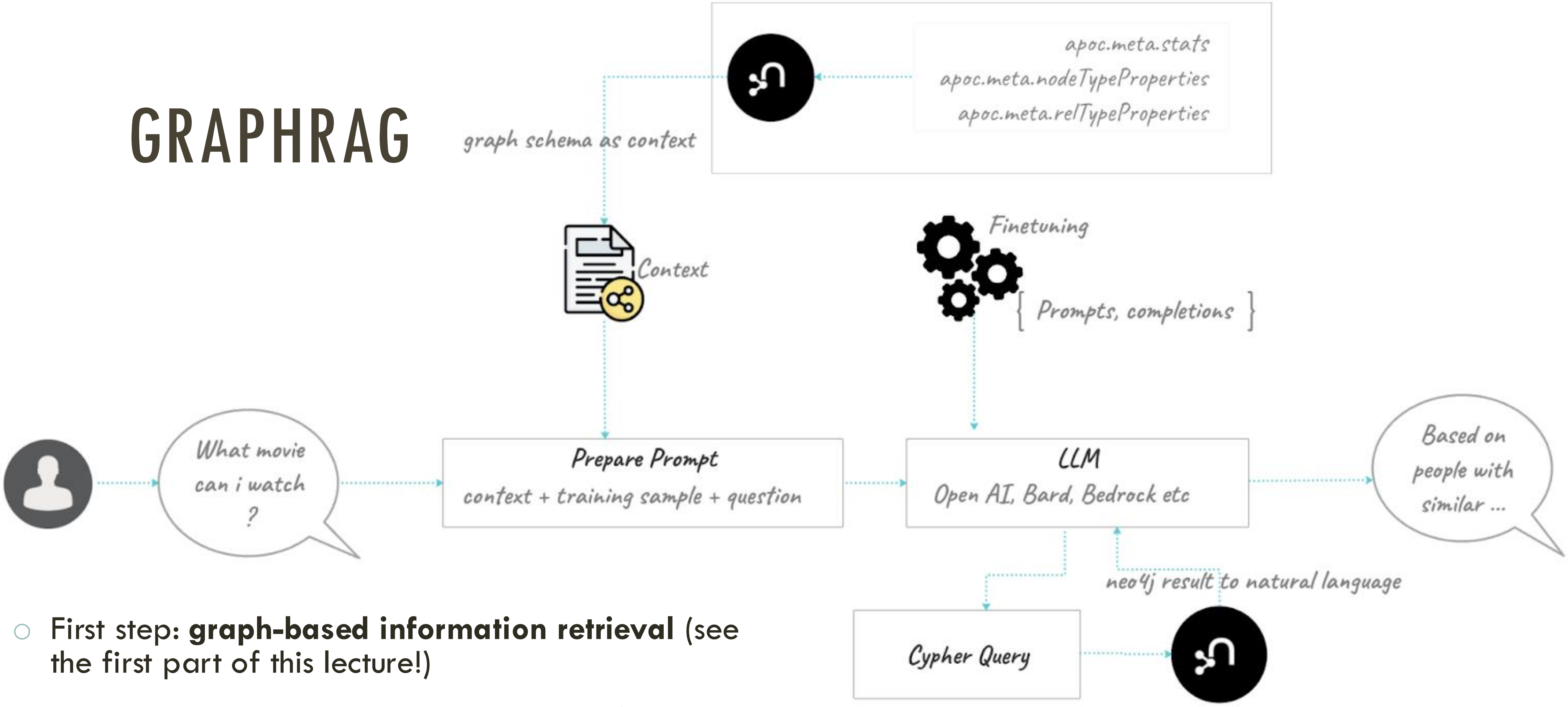
RETRIEVAL-AUGMENTED GENERATION (RAG)



- First step: documents retrieval (based on **vector databases** and **embeddings**)
- Second step: use LLM to generate output for user
- Easy to implement, but **lacks a comprehensive understanding of data**, relying primarily on similarity scores

Sources: (1) M. Gupta (2024), GraphRAG vs RAG: Which is Better?
(2) Z. Blumenfeld & E. Htet (2024), What Is Retrieval-Augmented Generation (RAG)?

GRAPHRAG



- First step: **graph-based information retrieval** (see the first part of this lecture!)
- Second step: use LLM to generate output for user
- More complicated, but **offers enhanced data understanding by capturing the context** (associated information and related entities)

Sources: (1) M. Gupta (2024), [GraphRAG vs RAG: Which is Better?]
(2) M. Hunger (2024), [Get Started With GraphRAG: Neo4j's Ecosystem Tools]

KG 201 RECAP

- Knowledge graphs are everywhere!
- SPARQL and GQL are the only languages you need to know
- Graphs are great for information retrieval (search) and exploration (recommendations)
- The graphs are vectors if you need it (for ML tasks)
- ChatGPT is a bullshit, but combination of LLMs and graphs (GraphRAG) is a reliable tool

THANK YOU FOR YOUR ATTENTION!

GEIST Research Group: https://geist.re/

Krzysztof Kutt: https://krzysztof.kutt.pl/

keep-calm.net