ADVANCED TOPICS

Krzysztof Kutt, PhD
Semantic Internet, WFAIS UJ

OUTLINE

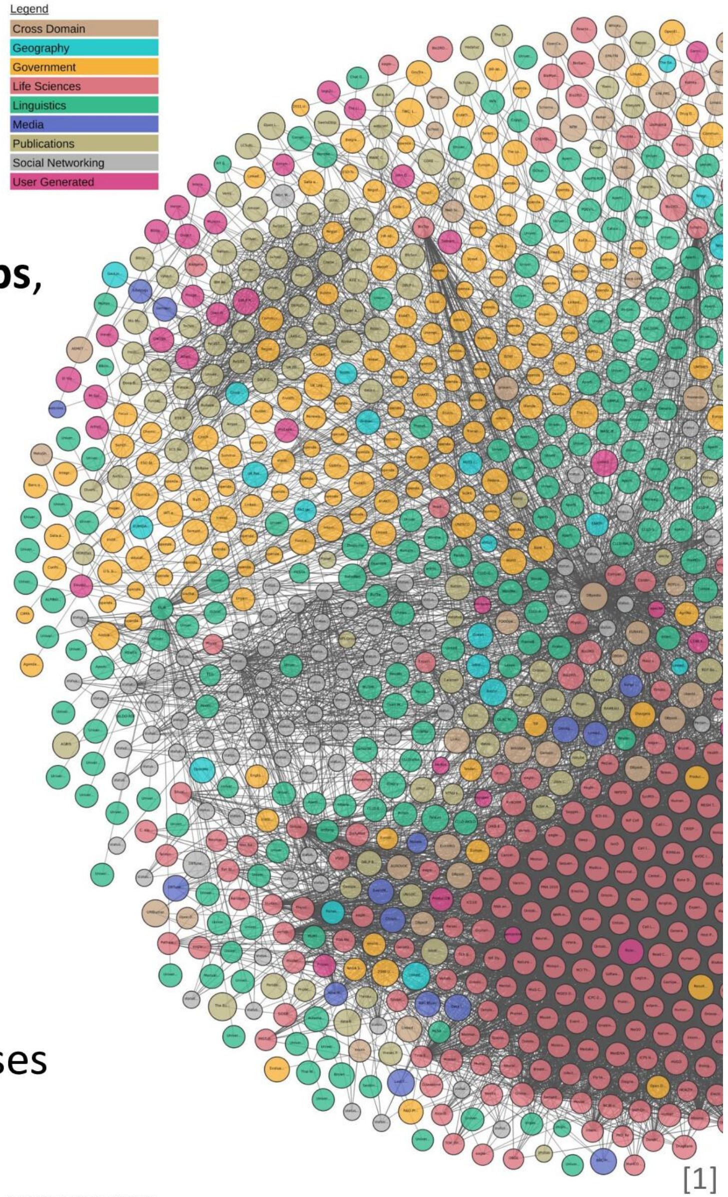
1. Knowledge graphs
2. Towards automated KG management
3. Semantic search and recommendations
4. Knowledge graph embeddings
5. Knowledge graph completion
6. ChatGPT is a bullshit. How can we fix it?

THE GRAPHS

How to compare them?

Knowledge Graph Recap

- A **Graph** consisting of **concepts**, **classes**, **properties**, **relationships**, and **entity descriptions**
- Based on **formal knowledge representations** (RDF(S), OWL)
- Data can be **open** (e.g. DBpedia, WikiData), **private** (e.g. supply chain data), or **closed** (e.g. product models)
- Data can be **original**, **derived**, or **aggregated**
- We distinguish
  - **instance data** (ground truth),
  - **schema data** (vocabularies, ontologies)
  - **metadata** (e.g. provenance, versioning, licensing)
- **Taxonomies** are used to categorize entities
- **Links** exist between internal and external data
- Including **mappings** to data stored in other systems and databases
- *Fully compliant to **FAIR Data principles***



The Linked Open Data Cloud from lod-cloud.net

[1]

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

Knowledge Base Definition

A Knowledge Graph is a **Knowledge Base** that is a Graph.

A **knowledge base** (**KB**) is a technology used to store complex structured and unstructured information used by a computer system. The initial use of the term was in connection with expert systems which were the first knowledge-based systems.

Wikipedia

knowledge base

Free Online Dictionary of Computing

<*artificial intelligence*>

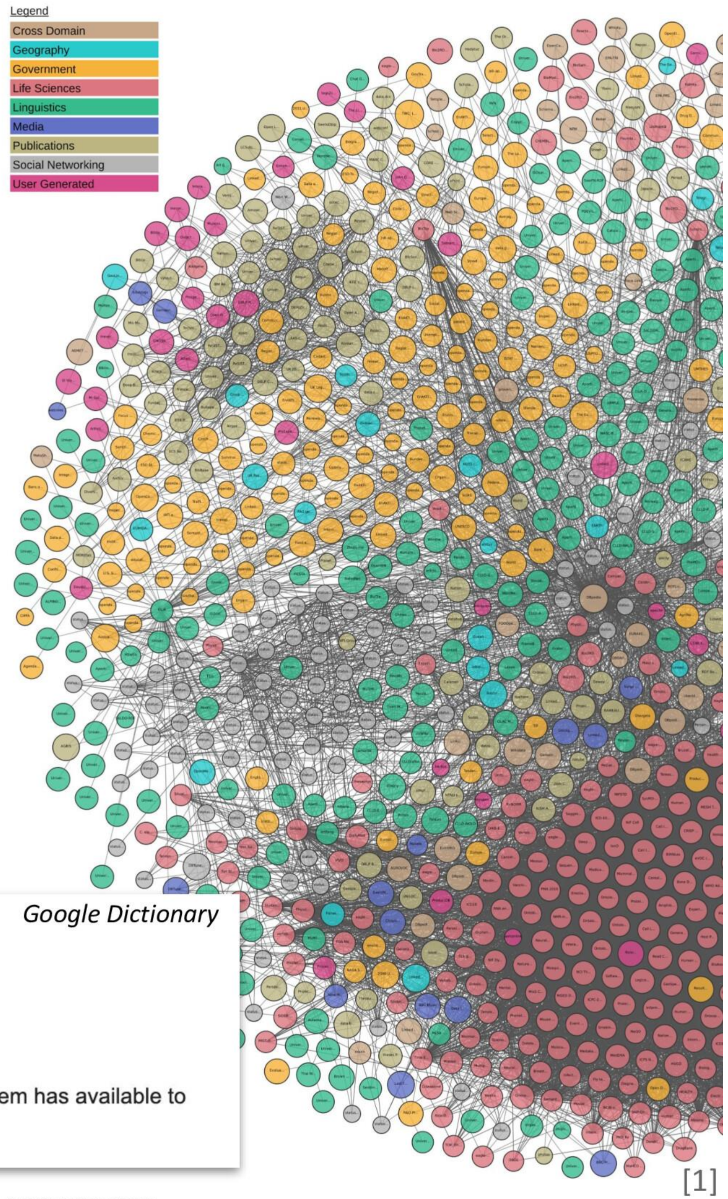
A collection of knowledge expressed using some formal knowledge representation language. A knowledge base forms part of a knowledge-based system (KBS).

Google Dictionary

knowledge base

noun

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.



The Linked Open Data Cloud from lod-cloud.net

[1]

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

Graph Definition

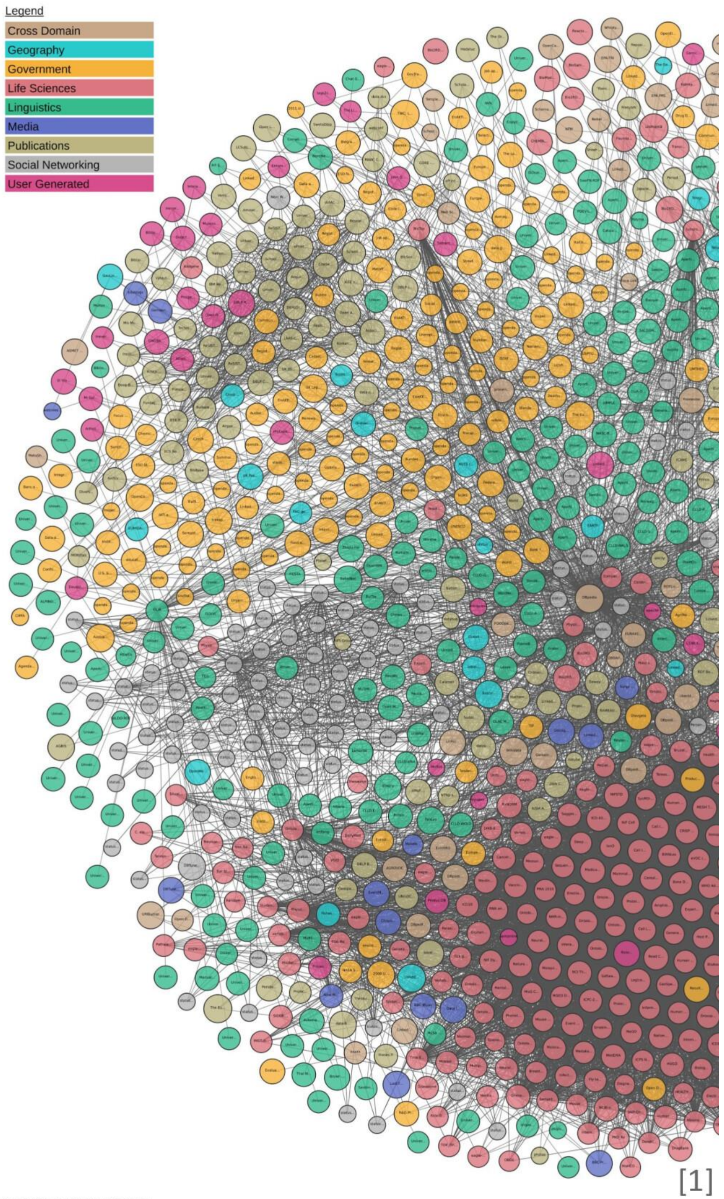
A Knowledge Graph is a Knowledge Base that is a **Graph**.

Definition 1.1

A **simple directed graph** **G=(V,E)** consists of a set **V** of **vertices**, $|V|=n$, and a set **E** of **directed edges**, $E \subseteq V \times V$, where each edge $e_i=(v_k, v_l)$, $e_i \in E$
is an ordered pair of two vertices (v_k, v_l) with $v_k, v_l \in V$.

Definition 1.2

- 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.
- An **edge-labelled graph** is a graph that has an additional **labelling function** $\lambda : E \rightarrow L$ that maps each edge in E to an element in a set of labels L (similarly for vertex-labelled graphs).



[1]

The Linked Open Data Cloud from lod-cloud.net

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

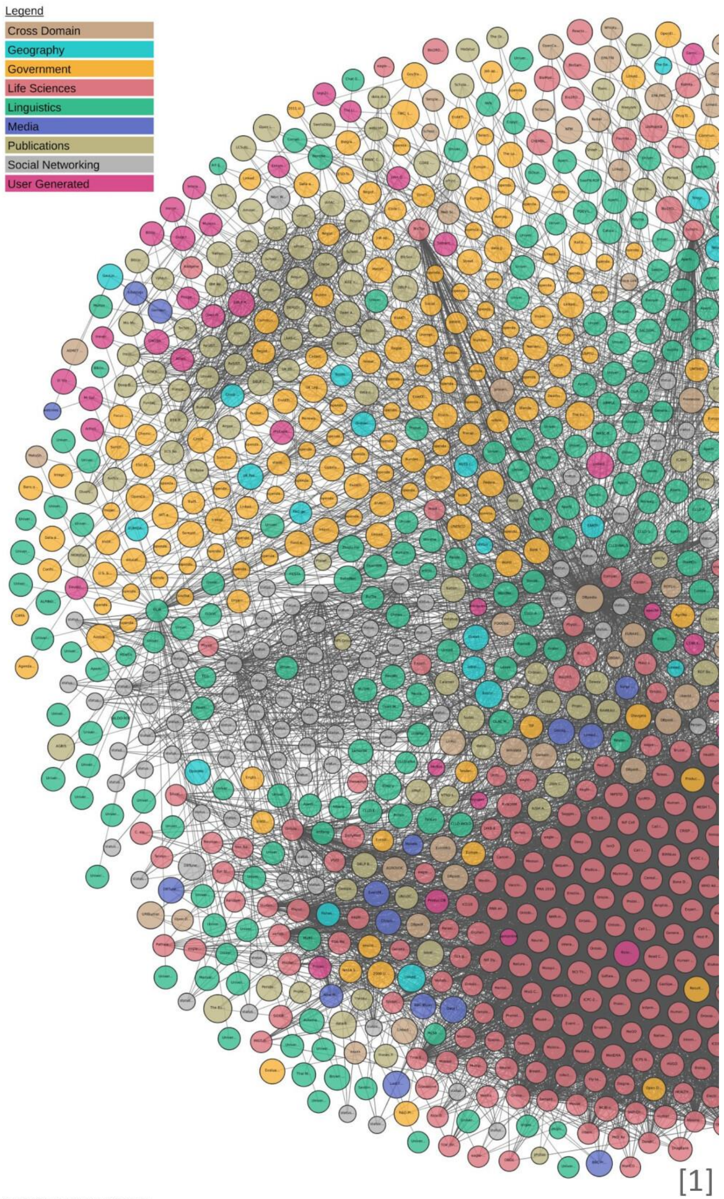
Graph Definition (cont.)

Definition 1.3

- An edge is 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 directed graph, the **in-degree** of a vertex is the number of edges pointing towards it; analogously for **out-degree**.

Definition 1.4

- A **directed path** in a directed graph is a sequence of consecutive edges $(e_1, e_2, \ldots, e_n)$ with $e_i=(v_l,v_k)$ and $e_{i+1}=(v_k,v_m)$.
- A directed graph is **strongly connected** if there is a directed path from any vertex to any other vertex.



[1]

The Linked Open Data Cloud from lod-cloud.net

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

How Can You Characterize a Knowledge Graph?

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

"Should I use Knowledge Graph A or Knowledge Graph B to solve my problem?"

- How to compare two Knowledge Graphs?
  - Size
  - Coverage
  - Completeness
  - Level of Detail
  - Accuracy
  - Reliability
  - etc.

- **Idea**: **Structural Comparison** by just comparing the Graphs

[1]

Graph Centrality Measures

- **Network analysis** has developed methods for **finding the most important vertices in a graph**.
- **Vertex importance** based on the structure of such graphs is called **centrality.**
- But, what makes a node important?

[1]

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

What makes a Node important?

- Many networks can be considered to describe a **flow** of something (goods, information, etc.)
- A node might be **important**, if
  - a lot flows from it (in a supply chain),
  - to it (in a network of links), or
  - through it (in a communication network)
- Flow might be modelled by (weighted) paths, possibly factoring in their length and/or number
- Paths might be more important if they pass through important nodes
- In knowledge graphs, the importance of edges and nodes may also depend on more complex features (e.g., edge or vertex labels)

[1]

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

What makes a Node important?

- **Wikidata Example:**
  - A Wikidata entity (node) might be important, if it is referenced by many Wikipedia pages
  - what are the most important Climatologists?

```
SELECT ?climatologistLabel (SUM(?link) AS ?importance)
WHERE {
  ?climatologist wdt:P106 wd:Q1113838 .
  ?climatologist wikibase:sitelinks ?link.
  ?climatologist rdfs:label ?climatologistLabel
   FILTER (lang(?climatologistLabel)="en")
} GROUP BY ?climatologistLabel
ORDER BY DESC(?importance)
```

[1]

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

Wikidata Query Service

[SPARQL query](#)

```
SELECT ?climatologistLabel (SUM(?link) AS ?importance)
WHERE {
  ?climatologist wdt:P106 wd:Q1113838 .
  ?climatologist wikibase:sitelinks ?link.
  ?climatologist rdfs:label ?climatologistLabel FILTER (lang(?climatologistLabel)="en")
} GROUP BY ?climatologistLabel
ORDER BY DESC(?importance)
```

256 results in 550 ms

climatologistLabel	importance
Alexander von Humboldt	117
Paul Jozef Crutzen	49
Wladimir Köppen	47
Michael E. Mann	16
Léon Teisserenc de Bort	15
Judith Curry	14
Stephen Schneider	14

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

Degree Centrality

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- A simple form of centrality restricts to incoming/outgoing paths of length one

Definition 1.5

- The **in-degree centrality** of a directed graph is given by the in-degree of each node.
- The **out-degree centrality** and the **degree centrality** (for undirected graphs) are defined analogously

- There are more sophisticated forms of centrality, as e.g.
  - Eigenvector centrality, Katz centrality, PageRank, etc.

[1]

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

Further Centrality Measures

- Further Measures to characterize a Knowledge Graph
  - Sizes
    - number of nodes
    - number of facts
    - avg number of facts per node
  - KG diameter

Definition 1.6

- The **eccentricity** of a node is the maximal distance between a certain node and any other node.
- The **diameter** of a graph is the maximum **eccentricity** of a graph, i.e. the greatest distance between any pair of nodes.
- To find the diameter of a graph, first find the **shortest path** between each pair of nodes. The greatest length of any of these paths is the **diameter of the graph**.

[1]

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

Further Centrality Measures

- Further Measures to characterize a Knowledge Graph
  - Sizes
    - number of nodes
    - number of facts
    - avg number of facts per node
  - KG diameter
  - KG radius

Definition 1.8

- The **radius** of a graph is the minimum eccentricity of a graph, i.e. the shortest of the maximum distances between any pair of nodes.

[1]

Further Centrality Measures

- Further (structural) measures to characterize a Knowledge Graph:
  - Sizes
    - number of nodes
    - number of facts
    - avg number of facts per node
  - KG diameter
  - KG radius
  - avg in/out degree
  - avg path length
  - and many more...

[1]

Knowledge Graphs and Important Nodes

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- In **Knowledge Graphs**, the importance of nodes might further be depending on
  - the properties (i.e. edge attributes)
  - the node labels (i.e. further attributes of nodes)
  - specific nodes or edges might be ignored, as e.g.
    - Basically for every entity in a (OWL encoded) knowledge graph the following fact holds:
      `:entity rdf:type owl:Thing`
    - Therefore, we might ignore this fact if we want to determine the importance of nodes

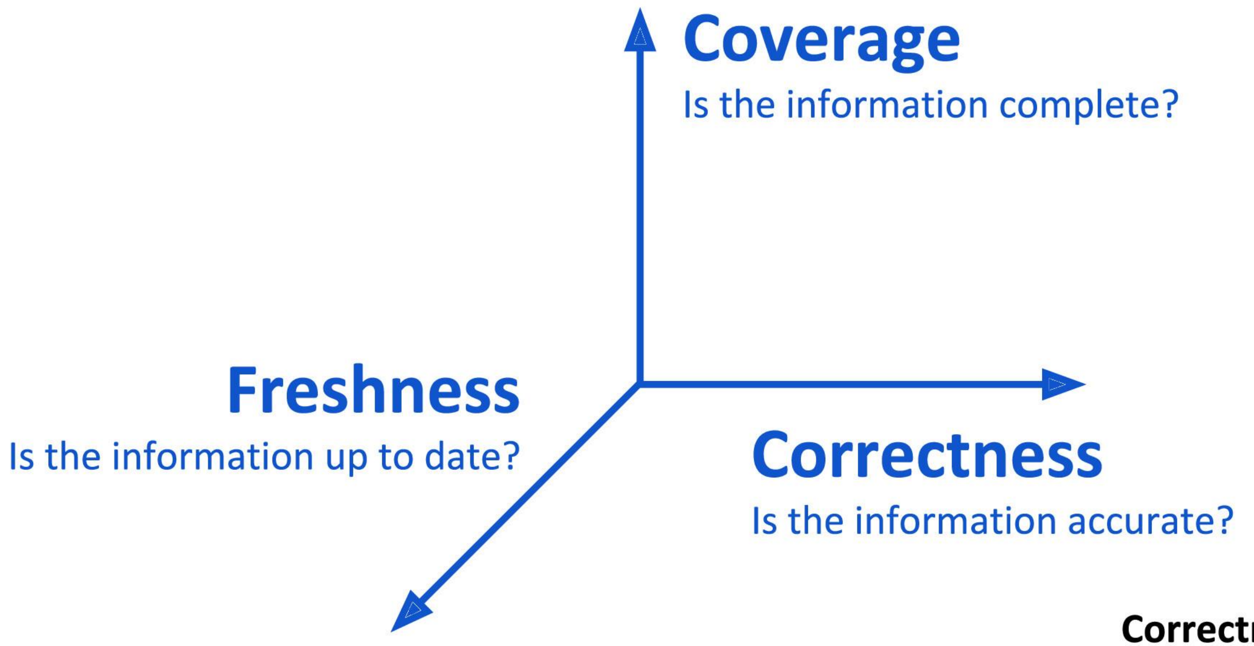
TOWARDS AUTOMATED KG MANAGEMENT

Mappings and alignment

Knowledge Graph Challenges

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- Building a small KG is easy but building a vast system like Google Knowledge Graph is a huge challenge



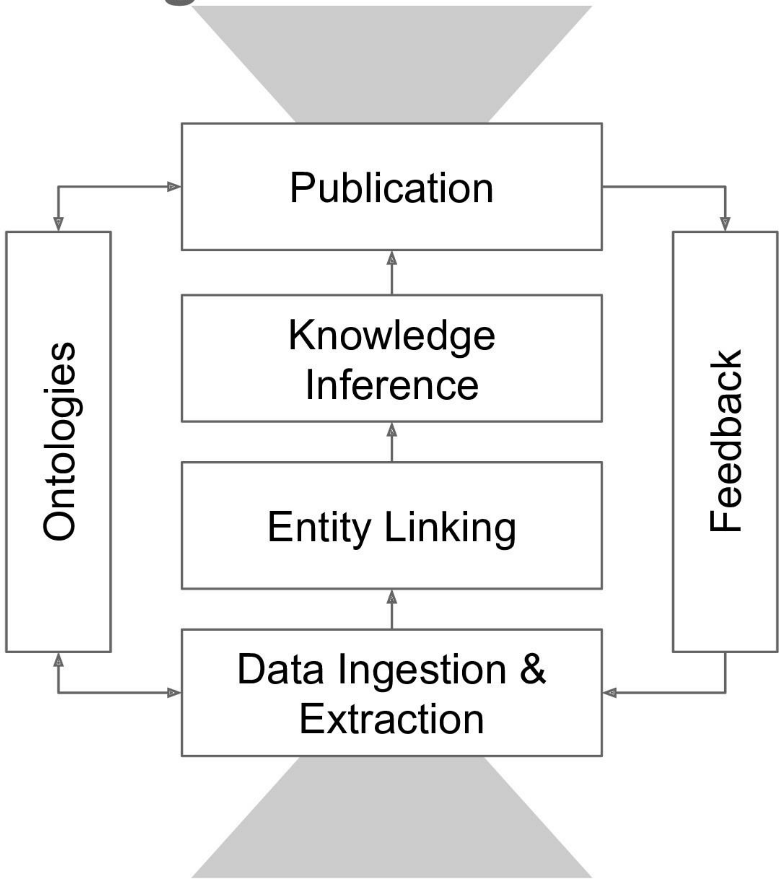
Increase **Freshness** & **Coverage**
▼
Harder to ensure **Correctness**

Increase **Correctness**
▼
Harder to ensure **Freshness** & **Coverage**

Correctness is always hard – what is true and correct?

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

Towards Automated Knowledge Graph Management



- Unsupervised knowledge extraction from unstructured data in open domain
- Semantic embedding via Ontologies
- Ultra-scale knowledge representations
- Large scale entity linking and disambiguation
- Autonomous knowledge inference & verification
- Knowledge Graph versioning and archiving
- Knowledge Precision vs Comprehensiveness

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

How to Automate Knowledge Graph Construction?

- Sound **Knowledge Graph Construction** relies on **Ontologies**
- Ontologies don't come for free, i.e. Ontology Design is very expensive wrt. time and resources
- Ontologies can be „learned" automatically

- **Ontology Learning** defines a set of methods and techniques
  - for **fundamental development** of new ontologies
  - for **extension or adaption** of already existing ontologies
- in a (partly) automated way from various resources.

Fundamental Types of Ontology Learning

- Ontology Learning from Text
  - automatic or semi-automatic generation of lightweight ontologies by means of text mining and information extraction
- Linked Data Mining
  - detecting meaningful patterns in RDF graphs via statistical schema induction or statistical relational learning
- Concept Learning in Description Logics and OWL
  - learning schema axioms from existing ontologies and instance data mostly based on Inductive Logic Programming
- Crowdsourcing Ontologies
  - combines the speed of computers with the accuracy of humans, as e.g. taxonomy construction via Amazon Turk or games with a purpose

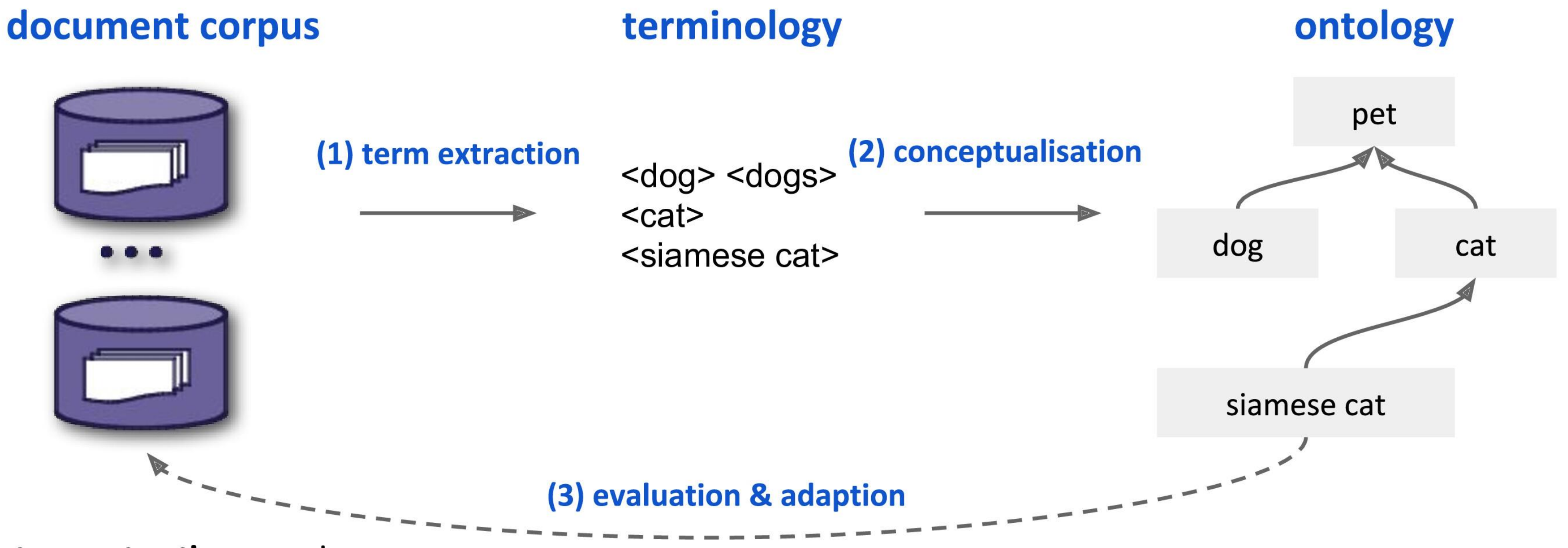
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Ontology Learning from Text

- **Ontology Learning from text** is the process of identifying terms, concepts, relations, and optionally axioms from textual information and using them to construct and maintain an ontology.
- Automatisation requires help from
    - Natural Language Processing (NLP)
    - Data Mining
    - Machine Learning techniques (ML)
    - Information Retrieval (IR)

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

Ontology Learning from Text - Basic Approach



term extractions requires linguistic processing (NLP) to identify important noun phrases and their internal semantic structure

terms: linguistic realisations of domain specific concepts

Concepts: clusters of semantically related terms

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

The Ontology Learning Layer Cake

KIT
Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

Country $\sqsubseteq \leq 1$ hasCapital.$\top$	General Axioms
River $\sqcap$ Mountain $\sqsubseteq \bot$	Axiomatic Schemata
capitalOf $\sqsubseteq$ locatedIn	Relation Hierarchies
flowThrough(dom:River, range:GeoEntity)	Relations
Capital $\sqsubseteq$ City , City $\sqsubseteq$ InhabitedGeoEntity	Concept Hierarchies
c:=country:=<description(c), uri(c)>	Concept Description
{country, nation, land}	Multilingual Synonyms
river, country, nation, city, capital, ...	Terms

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

Ontologies are NOT the Reality

- Ontologies are a **context-dependent projection (model) of the Reality**
- **Different ontologies** might represent the **same (or similar) knowledge**, as e.g. ontologies might
  - reflect different tasks and requirements for applications
  - follow different conventions or restrictions

[1]

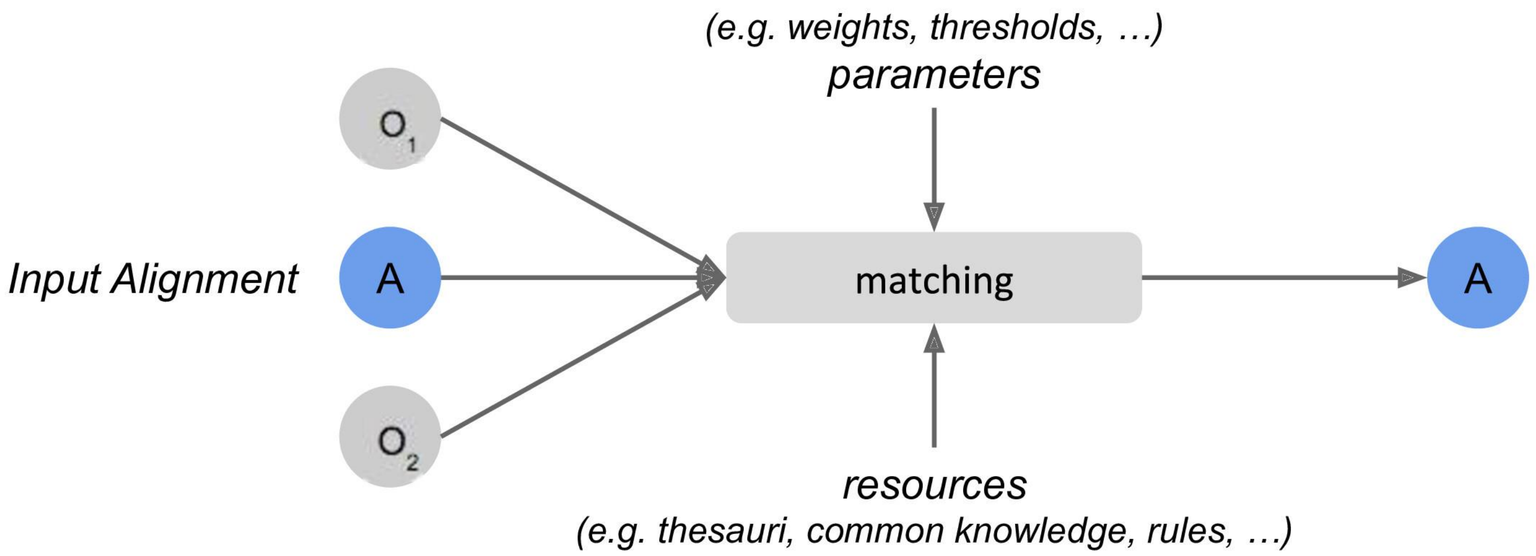
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

How Ontologies can Differ

- The **same term describes different concepts**
  - e.g. `Author` - *writer of a book vs. creator of a document*
- **Different terms describe the same concept**
  - e.g. `Author` vs. `Writer`
- **Different modeling conventions and paradigms**
  - e.g. `intervals` vs. `points` - *to describe temporal aspects*
- **Different level of granularity**
  - e.g. `Fiction` vs. `PoliticalFiction`, `ScienceFiction`, `RomanticFiction`, *etc. as literary Genres*
- **Different coverage or different point of view,** etc.

Ontology Alignment

- **Ontology Alignment** or **Ontology Matching** is the process of determining *correspondences* between ontological concepts



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

Correspondence or Mapping

- Given the ontologies O_1 and O_2, a **correspondence** or **mapping** among the entities $\mathbf{e_1}$ and $\mathbf{e_2}$ from O_1 and O_2 respectively, is defined as

$$\langle \mathbf{id}, \mathbf{e_1}, \mathbf{e_2}, \mathbf{r}, \mathbf{n} \rangle$$

- with
  - **id** ... a unique **identifier** of the correspondence
  - **r** ... a **relation**, as e.g. equivalence ($=$), more general ($\sqsupseteq, \geq$), less general ($\sqsubseteq, \leq$), disjointness($\perp$), part-of, etc...
  - **n** ... a **confidence measure** (typically in the range of **[0,1]**) holding for the correspondence between e_1 and e_2
- the correspondence $\langle \mathbf{id}, \mathbf{e_1}, \mathbf{e_2}, \mathbf{r}, \mathbf{n} \rangle$ asserts that the relation r holds between the entities e_1 and e_2 with confidence n

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

Complexity of Correspondences

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- Examples of **simple correspondences**:
  - `http://dbpedia.org/resource/Joseph_Fourier = https://www.wikidata.org/wiki/Q8772`
  - `Author = Writer`
  - Gas $\geq_{1.0}$ GreenhouseGas
  - rdfs:label $\geq_{0.9}$ dc:title

Complexity of Correspondences

- Examples of **more complex correspondences**:
  - $\text{speed} = \text{velocity} \times 2.237$
    $0.477 \times \text{speed} = \text{velocity}$
  - $\text{Book}(x) \land \text{author}(x,y) \land \text{Writer}(y) \Rightarrow_{.85} \text{writtenBy}(x, \text{concat}(y.\text{firstname}, y.\text{lastname}))$

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

Alignment Example

Book =1.0 Volume

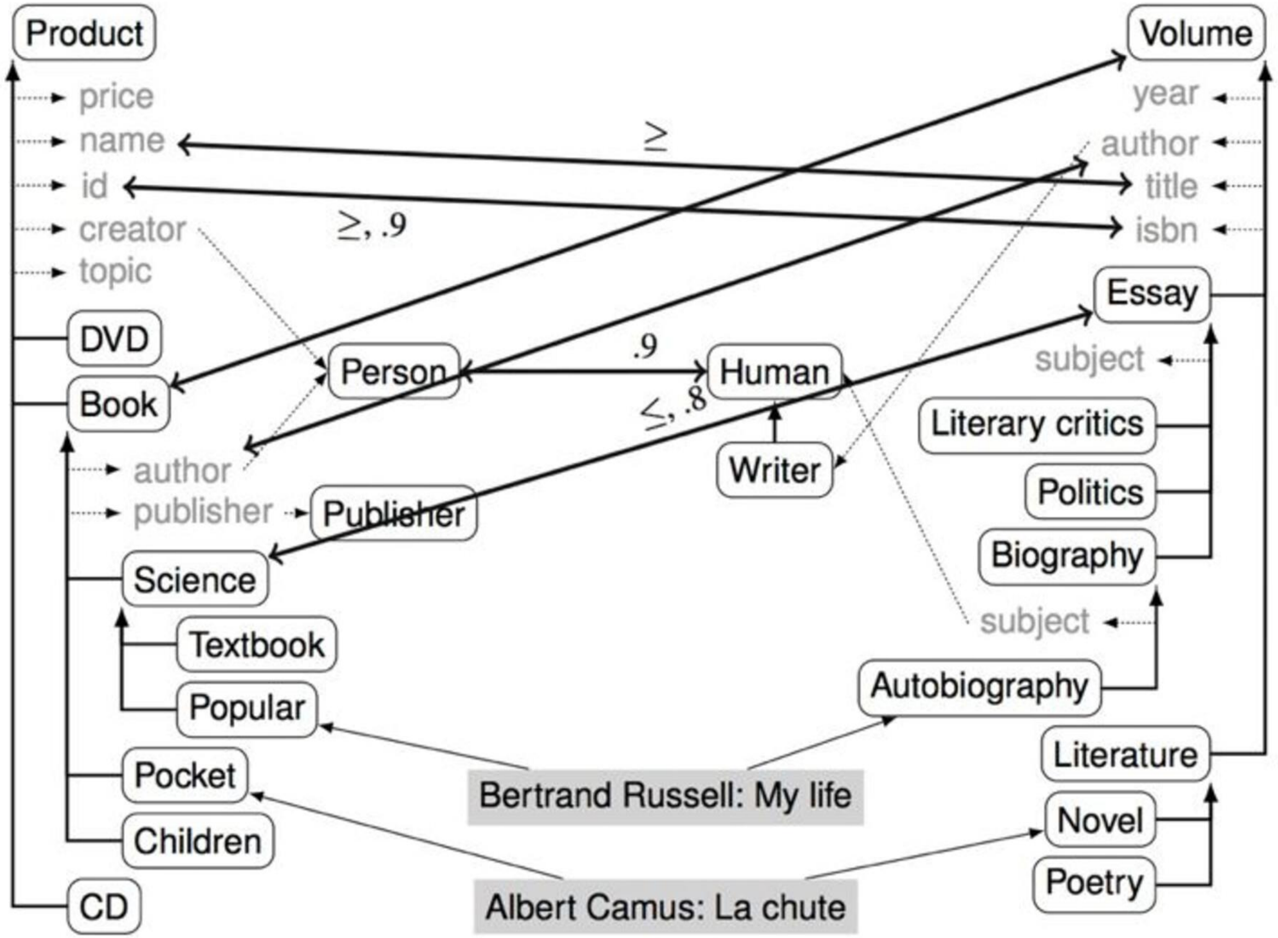
id ≥0.9 isbd

Person =0.9 Human

name ≥1.0 title

author =1.0 author

Science ≤0.9 Essay



Jérôme Euzenat, Pavel Shvaiko: Ontology Matching, Springer, 2007, p.48

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

Ontology Matching Techniques

- **Element-level Ontology Matching Techniques** consider ontology entities or their instances in isolation from their relations with other entities or their instances
  - **String-Based** - *matching names or descriptions of entities*
  - **Linguistic-Based** - *use NLP, lexicons, or domain specific thesauri to match words based on linguistic relations (homonymy, synonymy, partonomy, etc.), or exploiting morphological properties*
  - **Constrained-Based** - *take into account internal constraints applied to the definitions of entities, as e.g. types, cardinality of properties, etc.*
  - **Extensional-Based** - *use individual representation of classes, i.e. classes are considered similar if they share many instances*

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

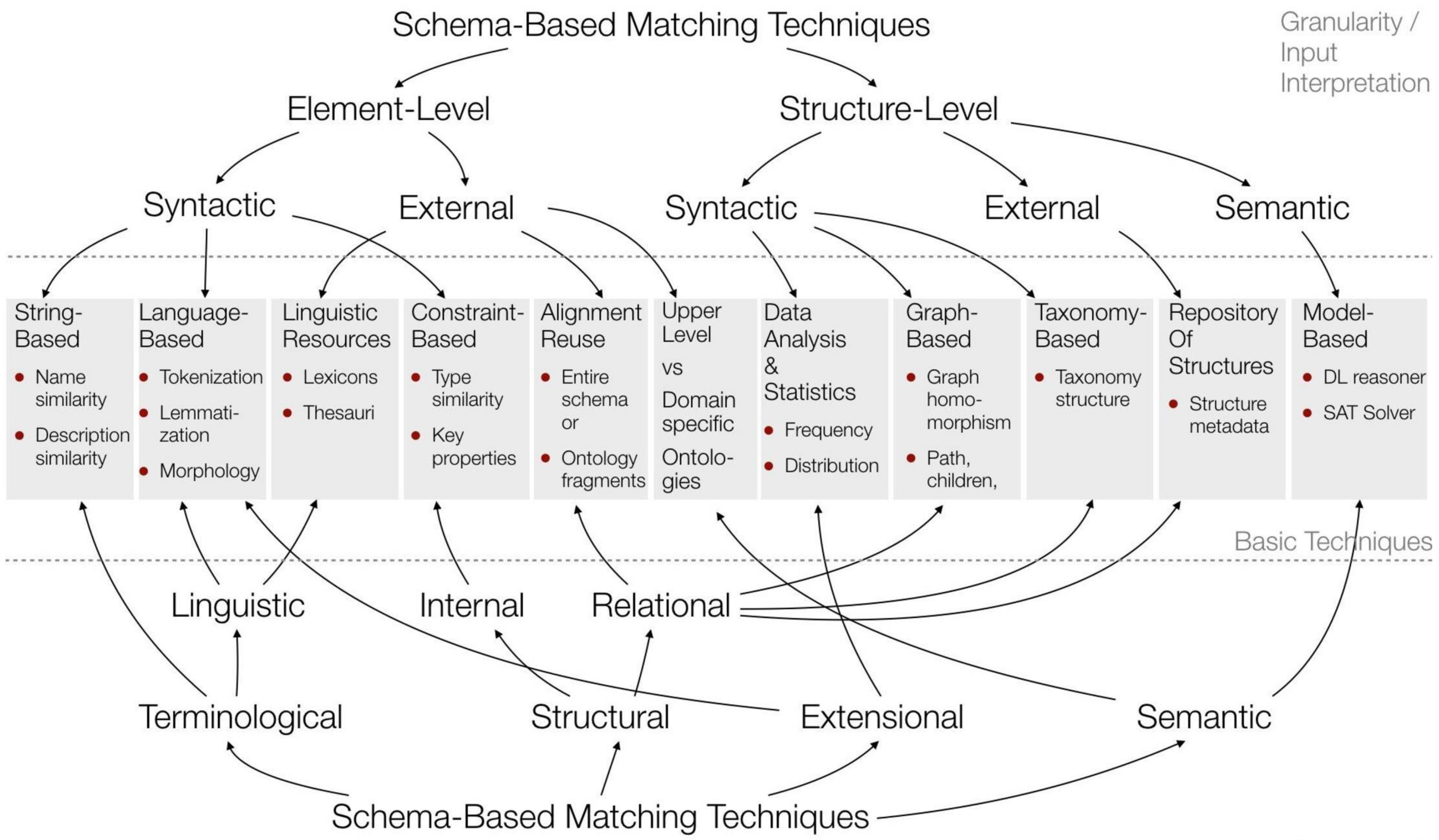
Ontology Matching Techniques

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- **Structure-level Ontology Matching Techniques** consider ontology entities or their instances to compare their relations with other entities or their instances
  - **Graph-Based** - *consider ontologies as labeled graphs, assumption: if nodes are similar, then also their neighbors must be similar*
  - **Taxonomy-Based** - *like graph-based algorithms, but consider only specialization/generalization relation*
  - **Method-Based** - *take into account semantic interpretation of the ontologies, assumption: if two entities are the same, then they share the same interpretation*
  - **Data Analysis and Statistics** - *take a large sample, try to find regularities, discrepancies, allows grouping or determining distance metrics, ...*

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

Ontology Alignment



Euzenat, Shvaiko: Ontology Matching, Springer 2007

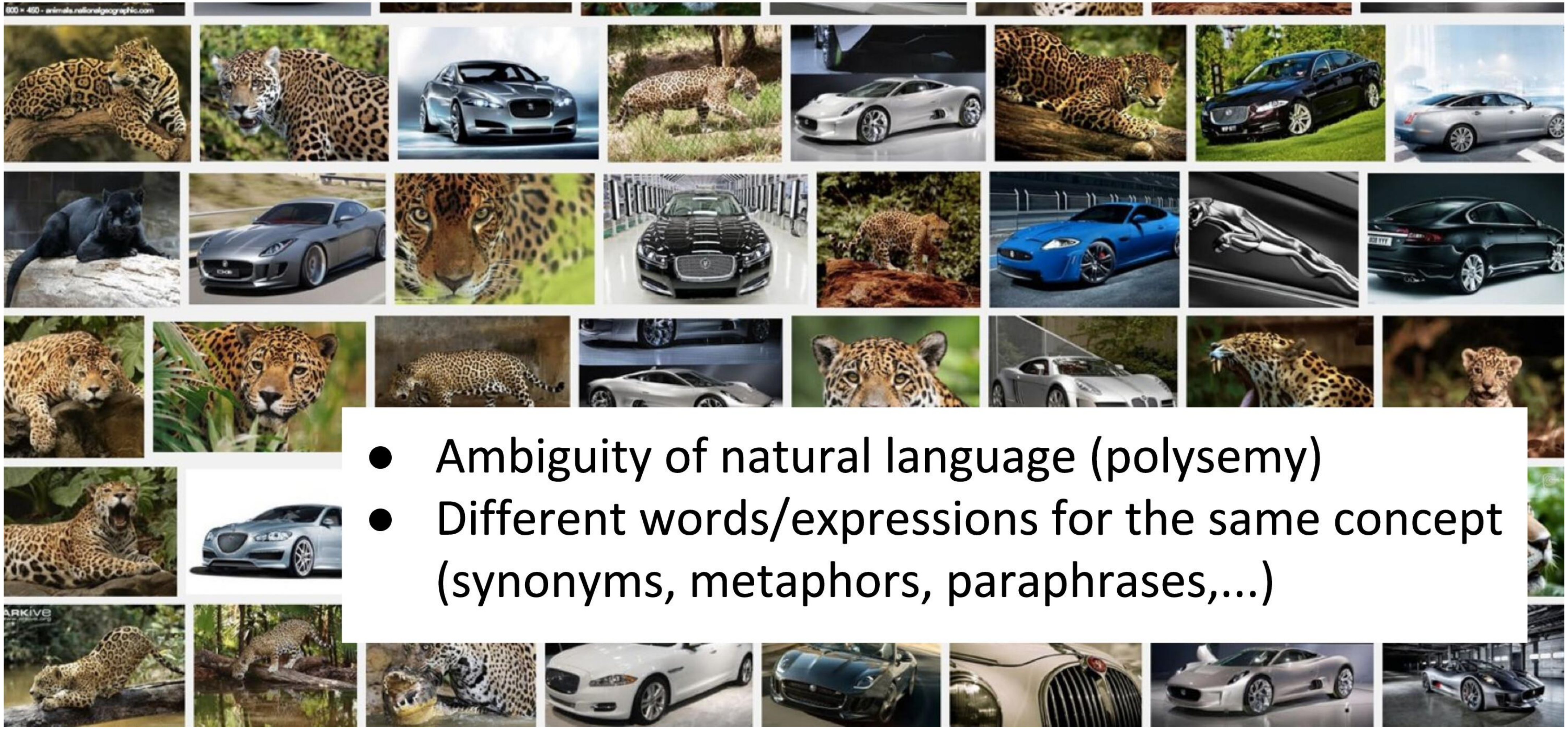
Slide from [Knowledge Graphs course] by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

SEMANTIC SEARCH AND RECOMMENDATIONS

Would graphs help search engines?

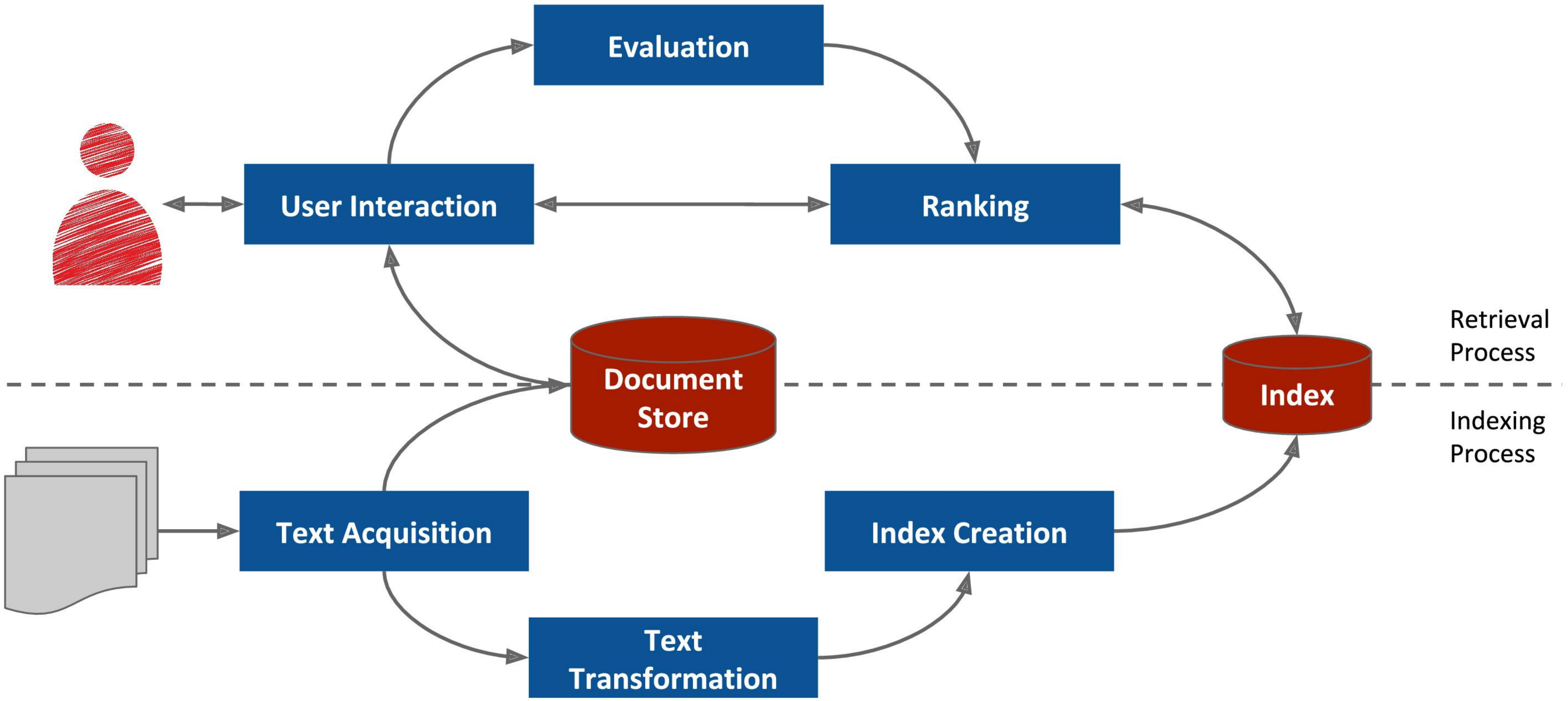
The Information Retrieval Dilemma

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,...)

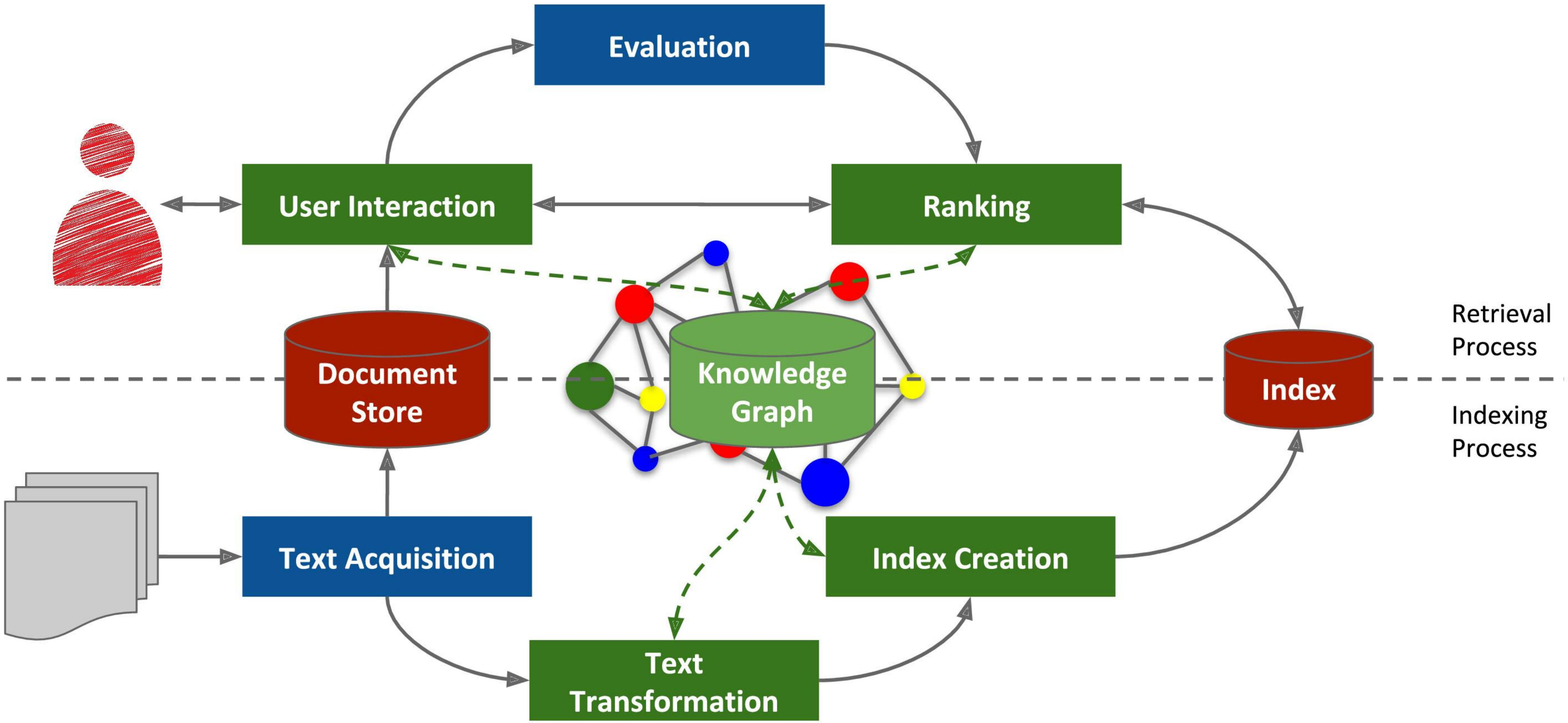
The Information Retrieval Process



[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

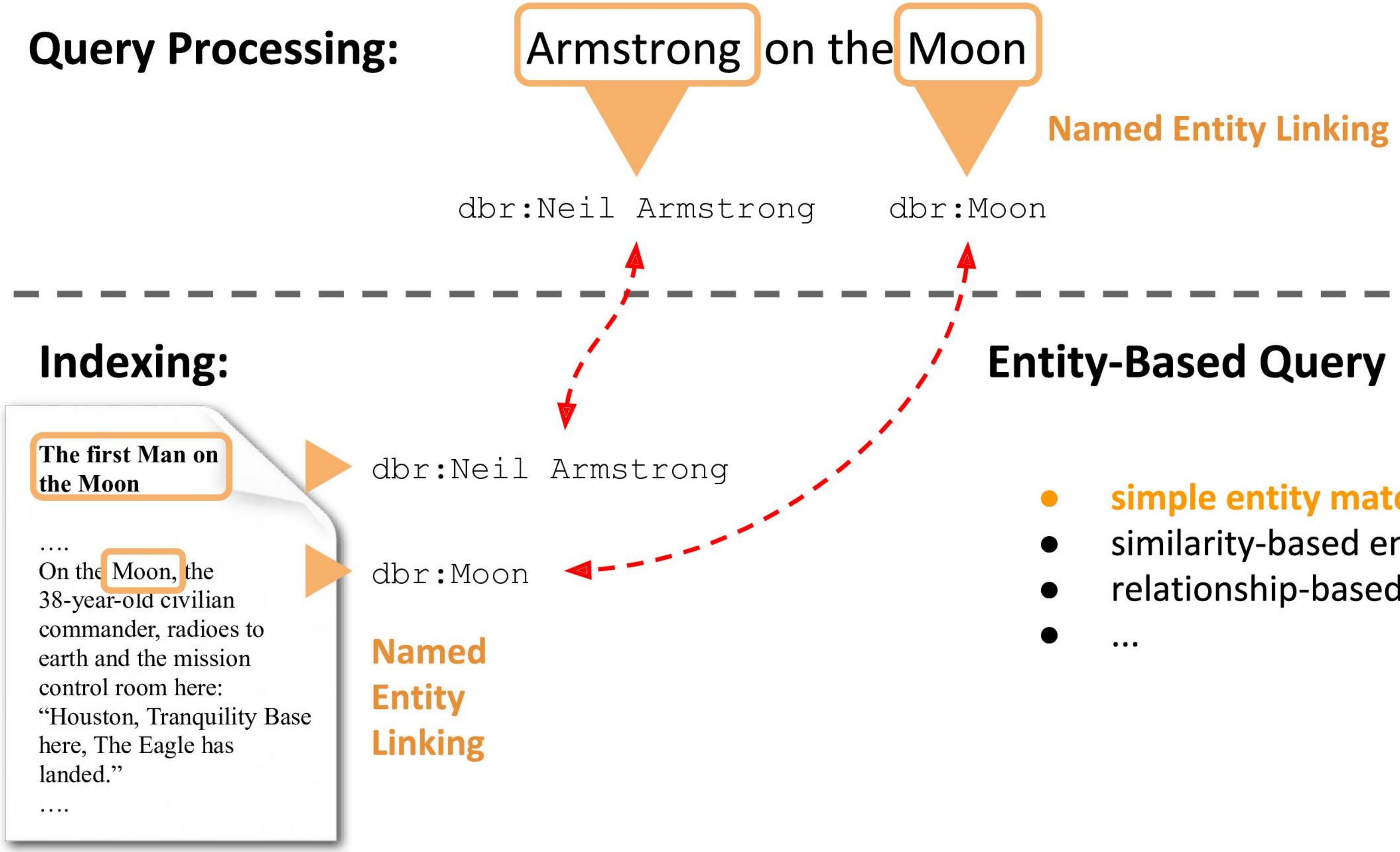
http://scihi.org/neil-armstrong/

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

[1]

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

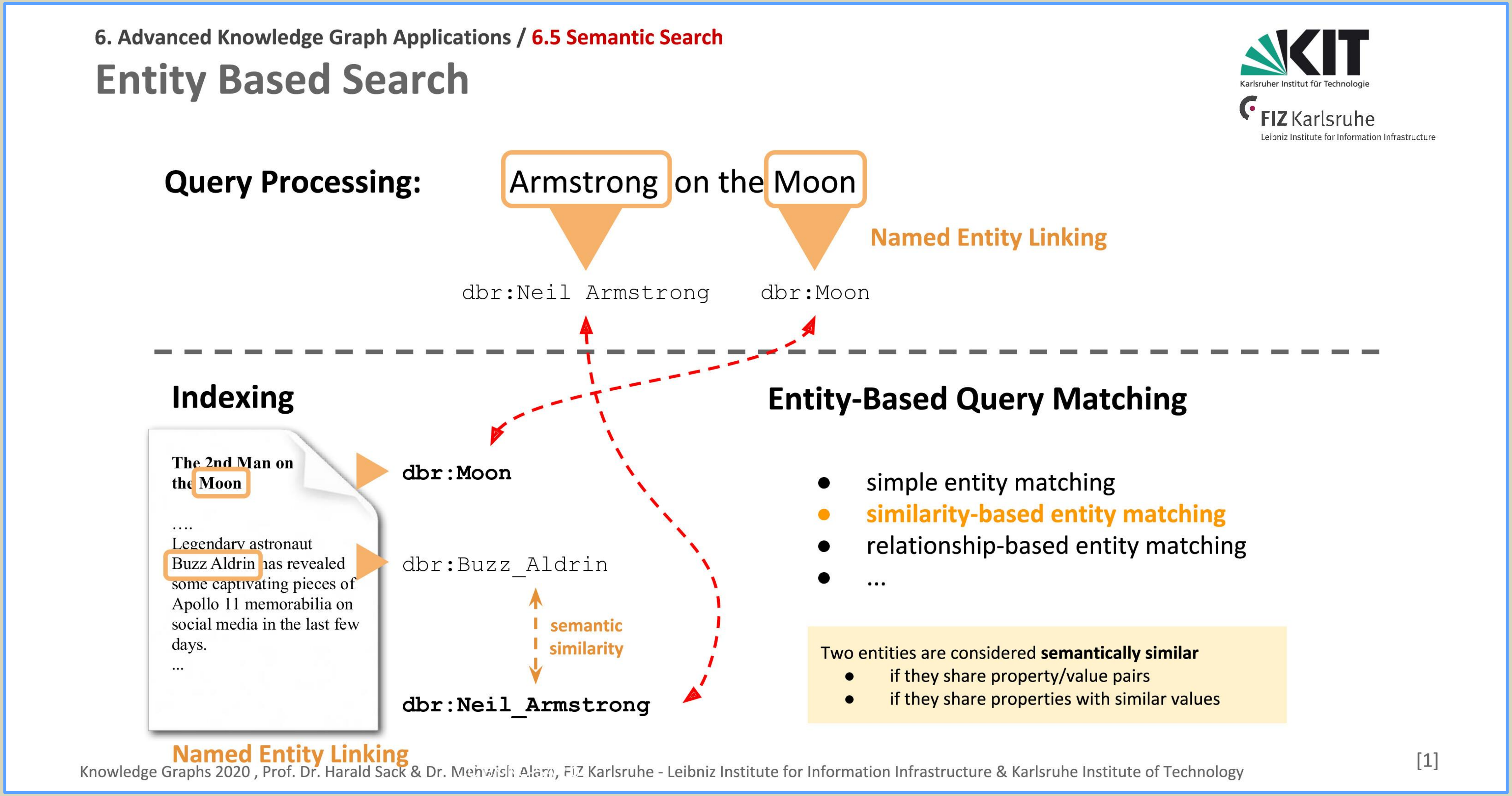
Entity Based Search



Entity-Based Query Matching

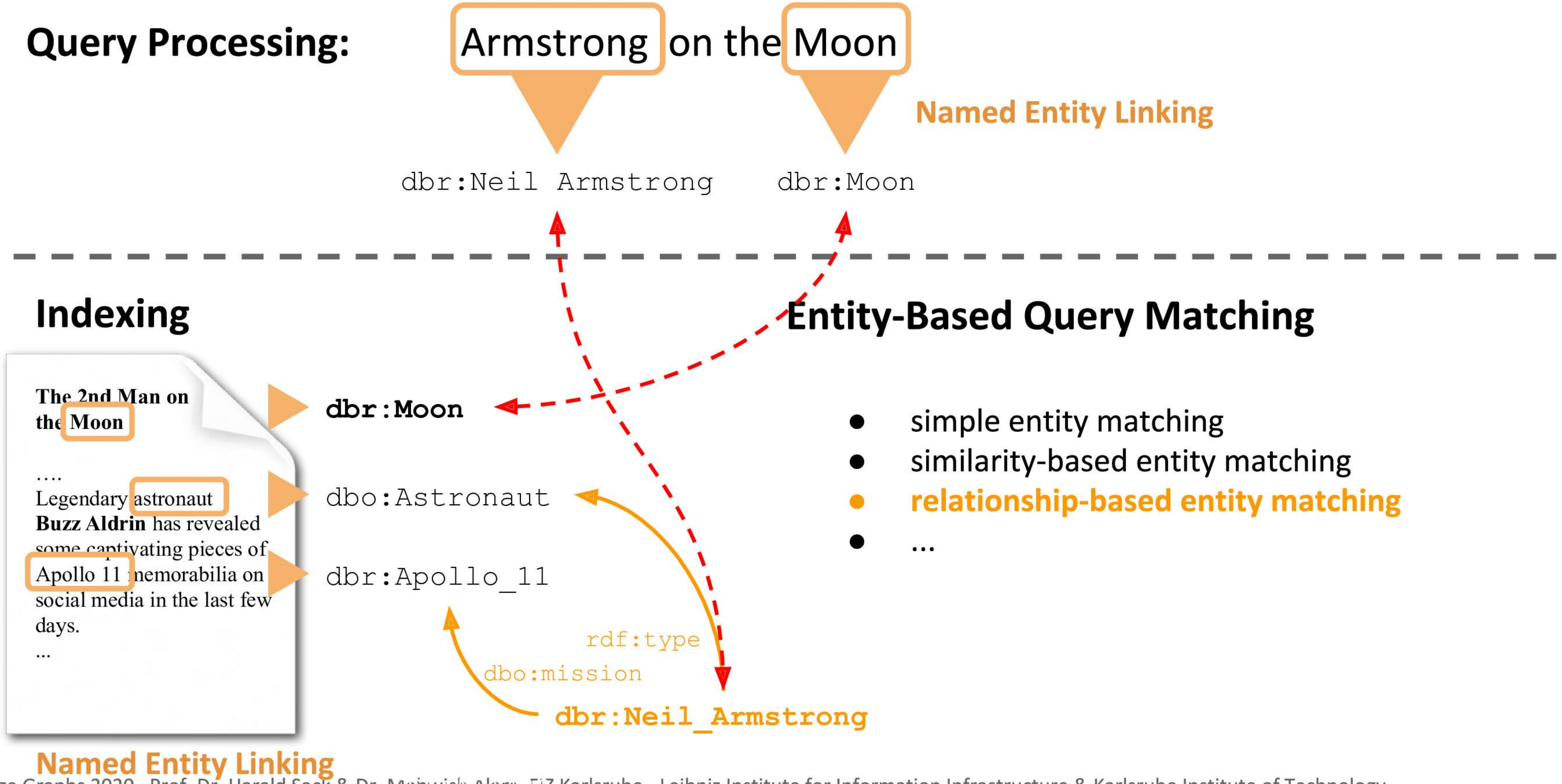
- simple entity matching
- similarity-based entity matching
- relationship-based entity matching
- ...

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

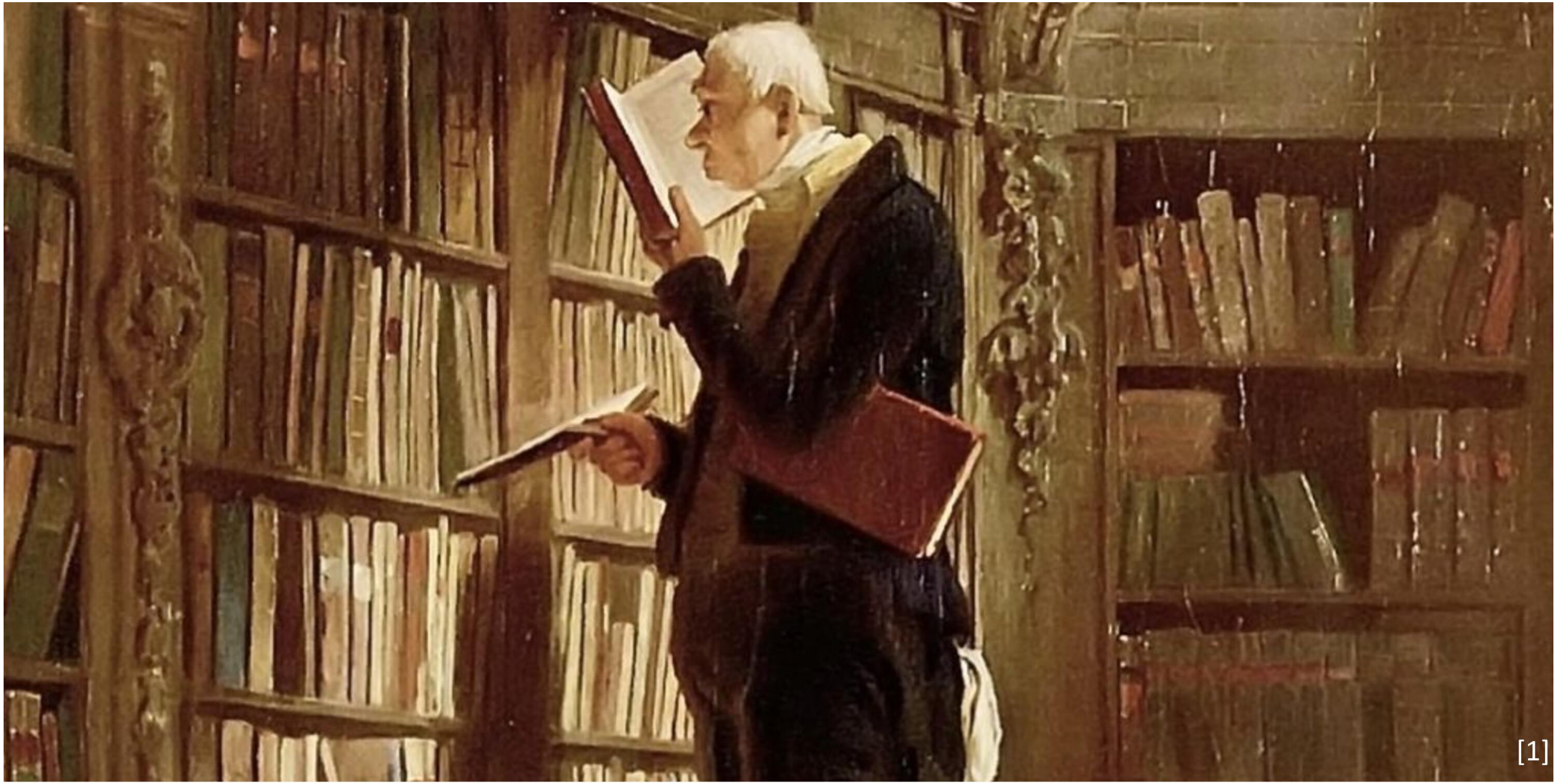


Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Entity Based Search



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

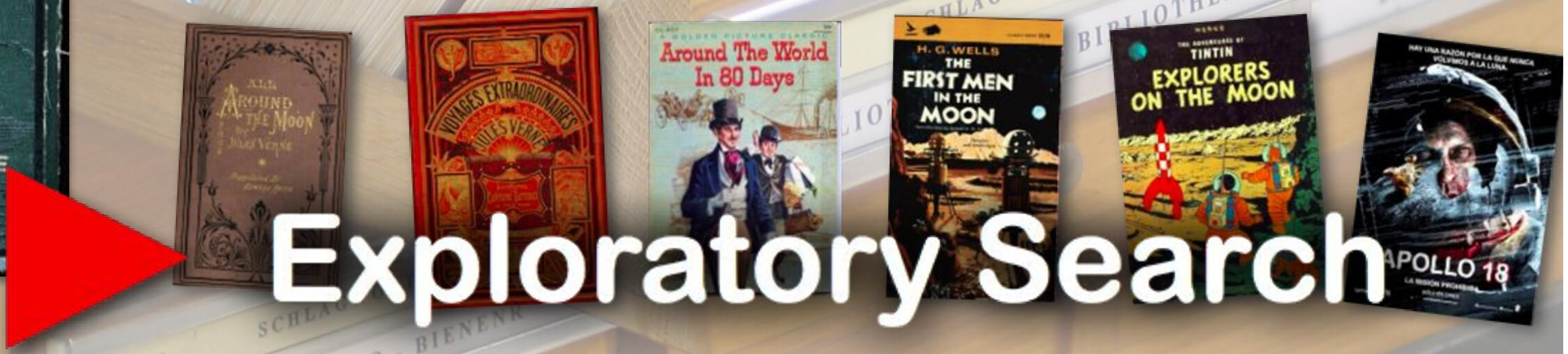
Author: Jules Verne
Title: From the Earth to the Moon

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).

Retrieval vs. Exploration

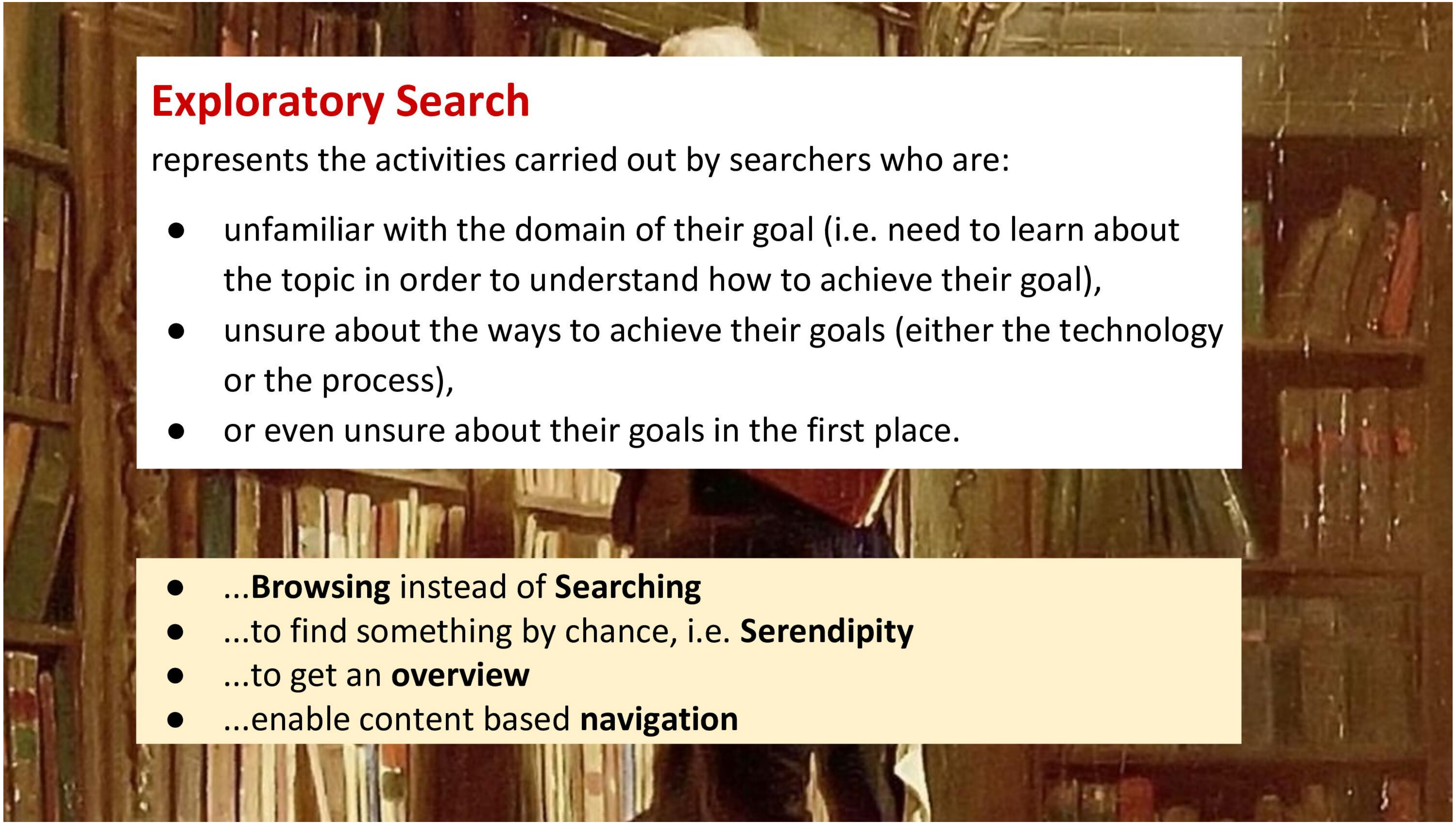
- *Find another („comparable“) book,*
  (that will be of interest for me...)
- *Find books of the same or of related topics*
- *How did the author / the topic develop over time?*
- *What else would I like to read?*
- ...



[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

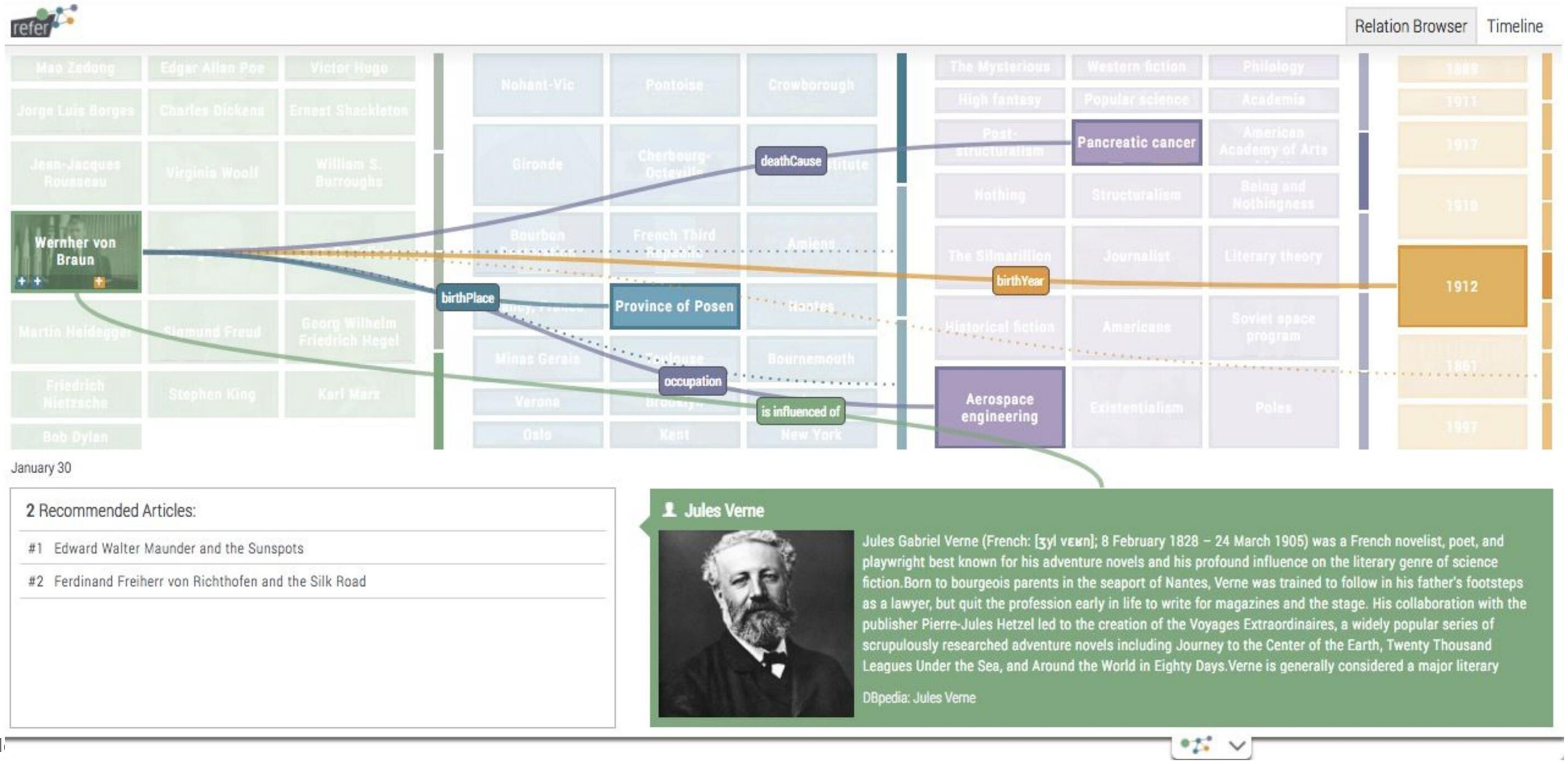
represents the activities carried out by searchers who are:

- 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).

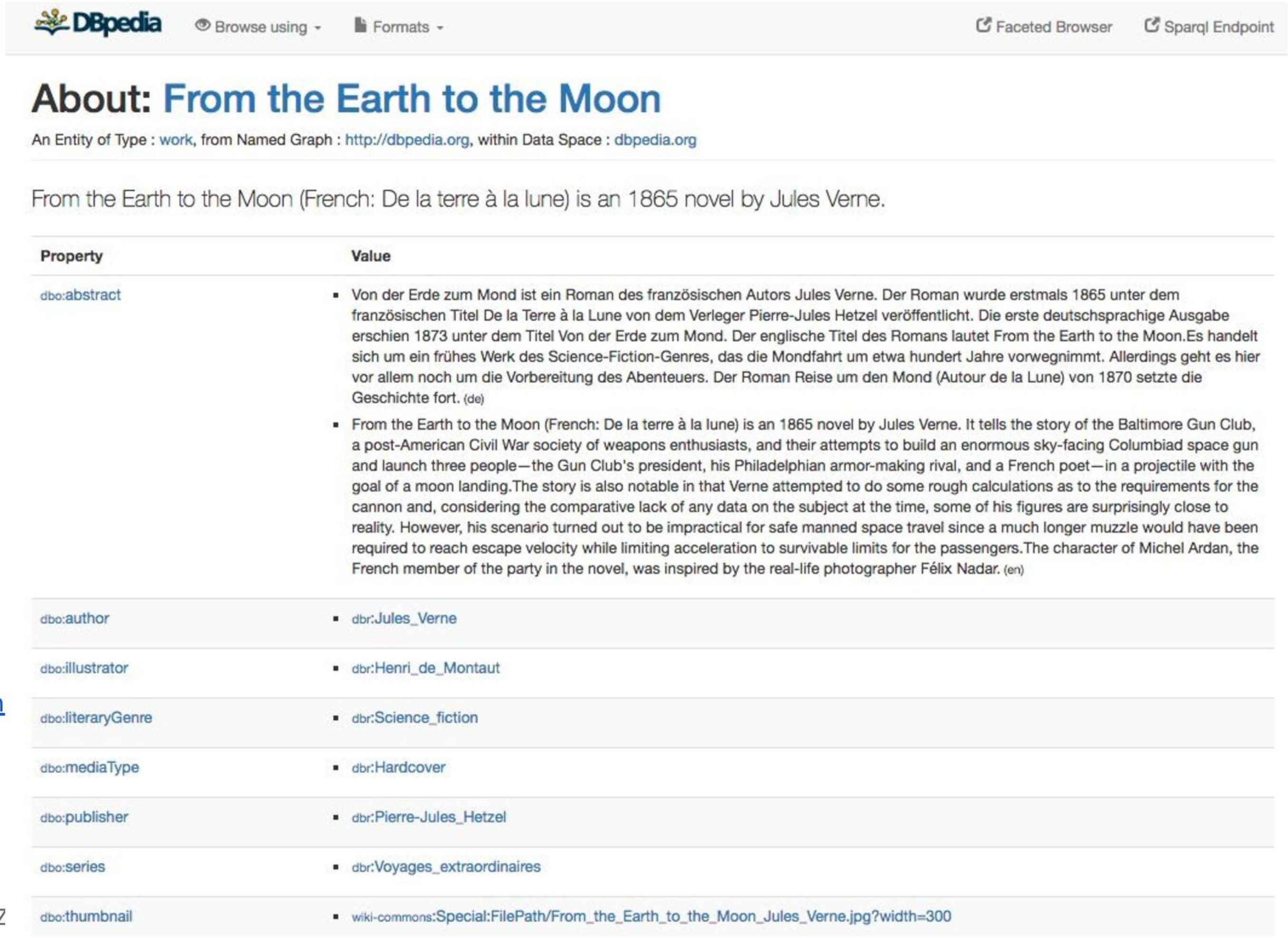
Exploratory Search and Recommendation



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

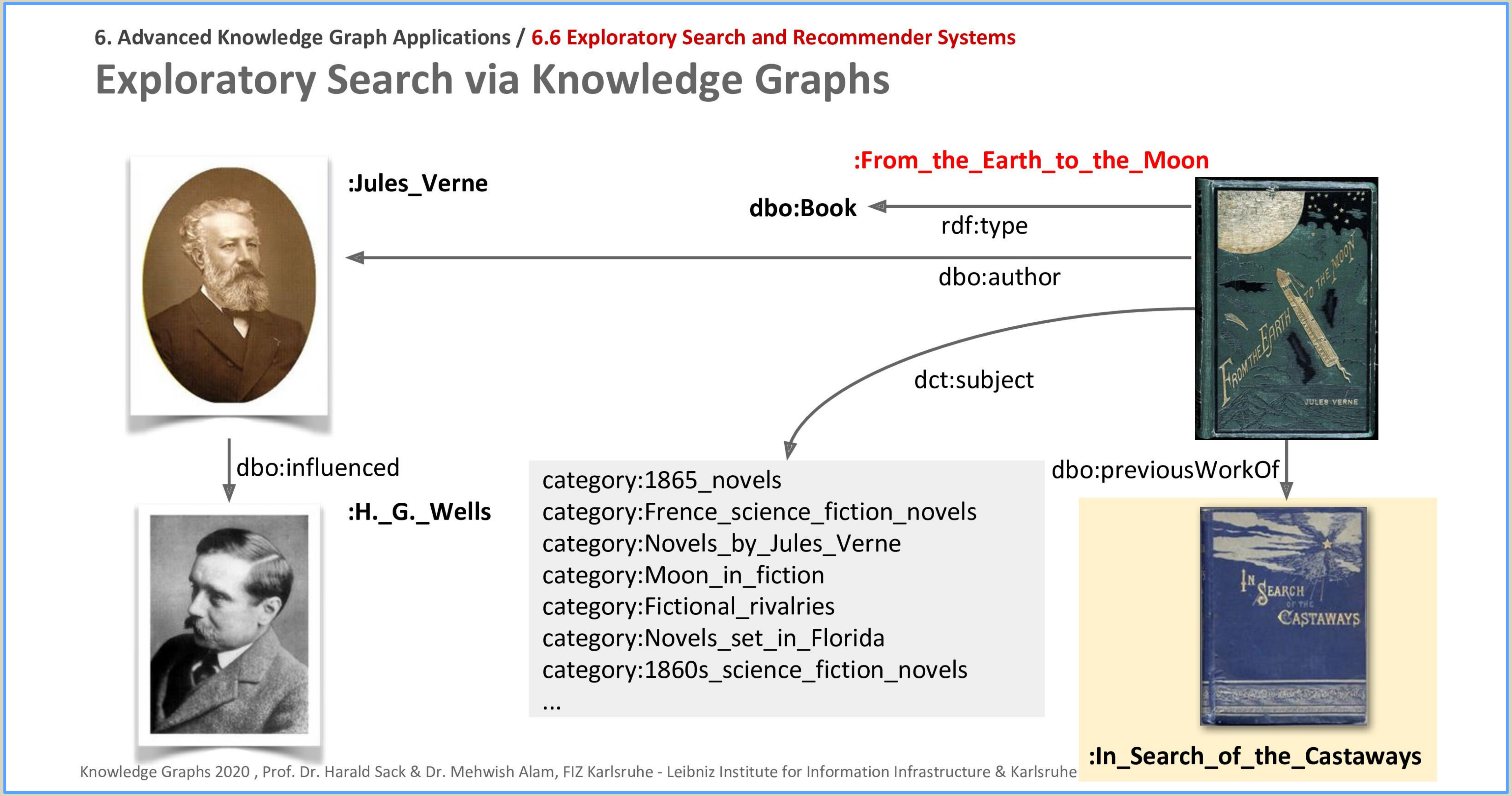
About: From the Earth to the Moon

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.

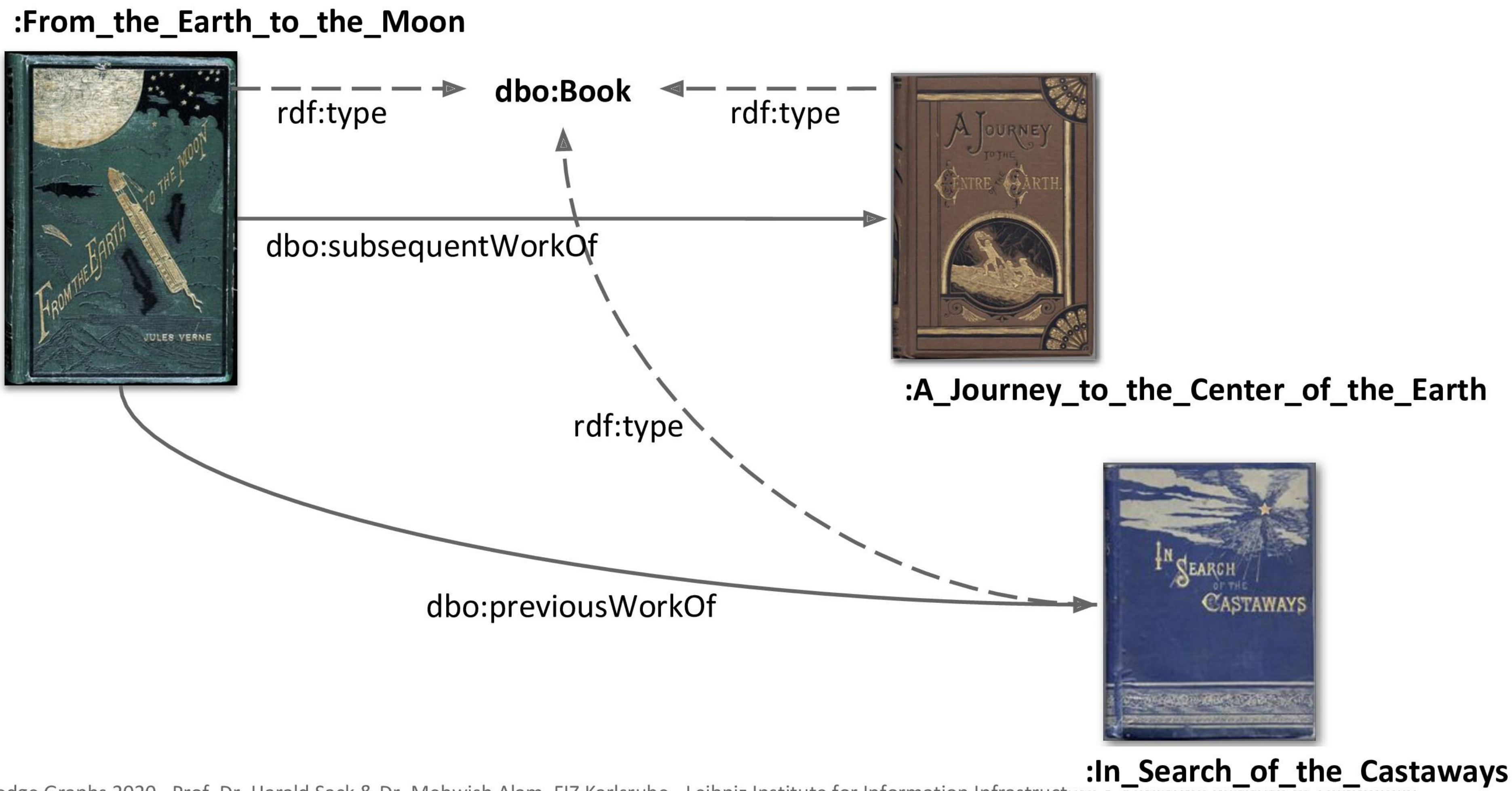
Property	Value
dbo:abstract	• Von der Erde zum Mond ist ein Roman des französischen Autors Jules Verne. Der Roman wurde erstmals 1865 unter dem französischen Titel De la Terre à la Lune von dem Verleger Pierre-Jules Hetzel veröffentlicht. Die erste deutschsprachige Ausgabe erschien 1873 unter dem Titel 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).



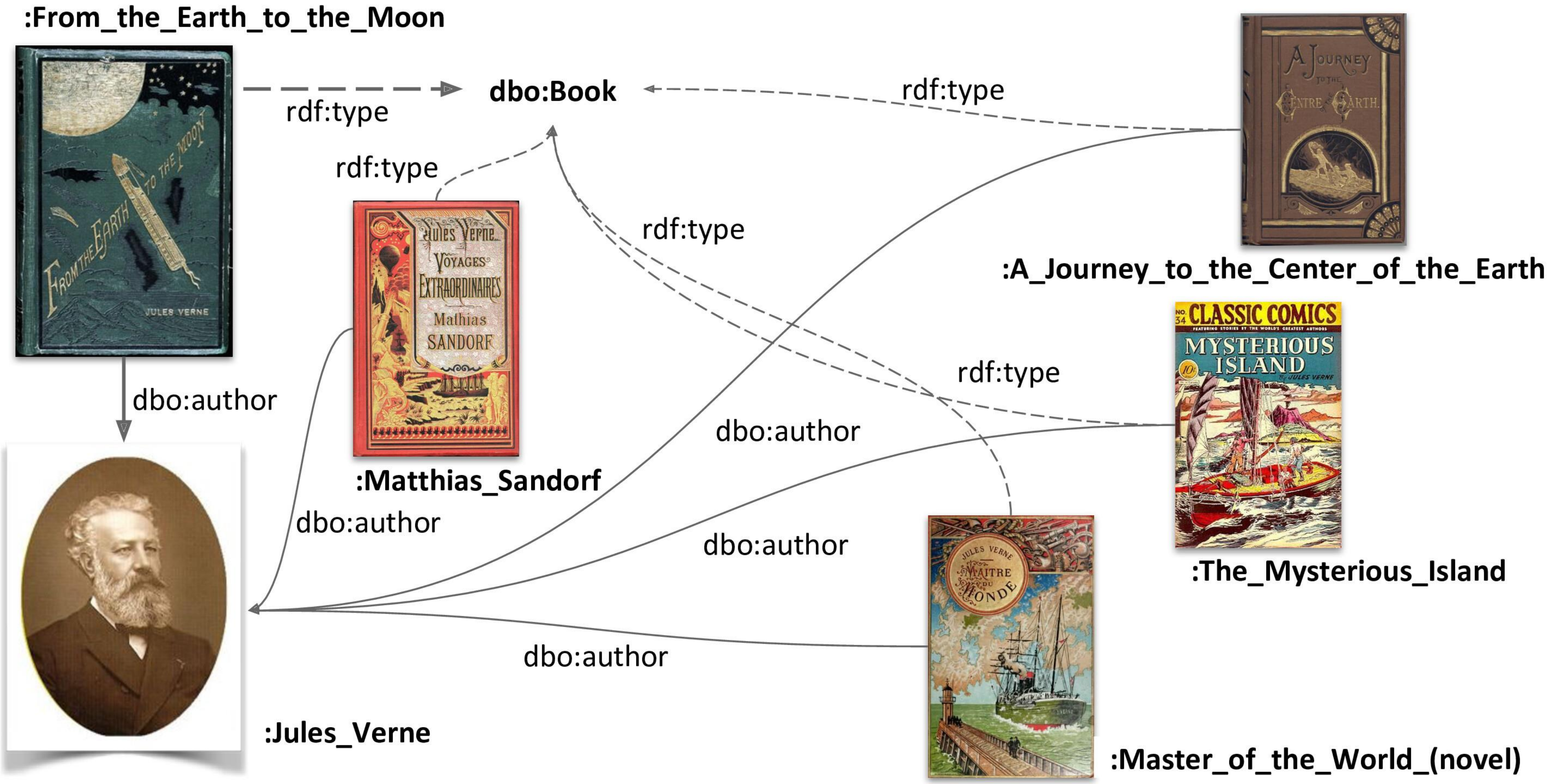
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Exploratory Search via Knowledge Graphs

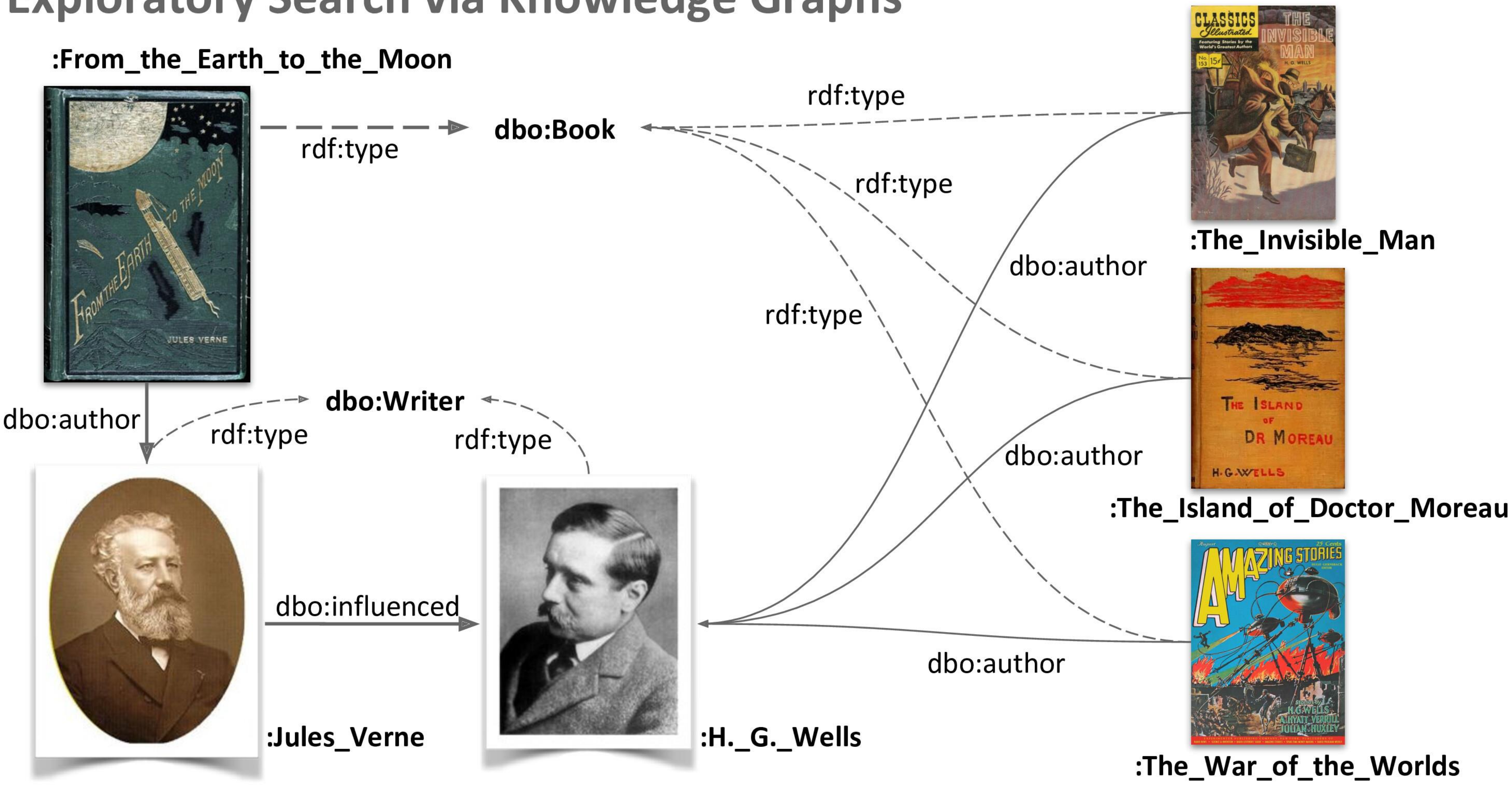


Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Exploratory Search via Knowledge Graphs



Exploratory Search via Knowledge Graphs



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

Exploratory Search via Knowledge Graphs

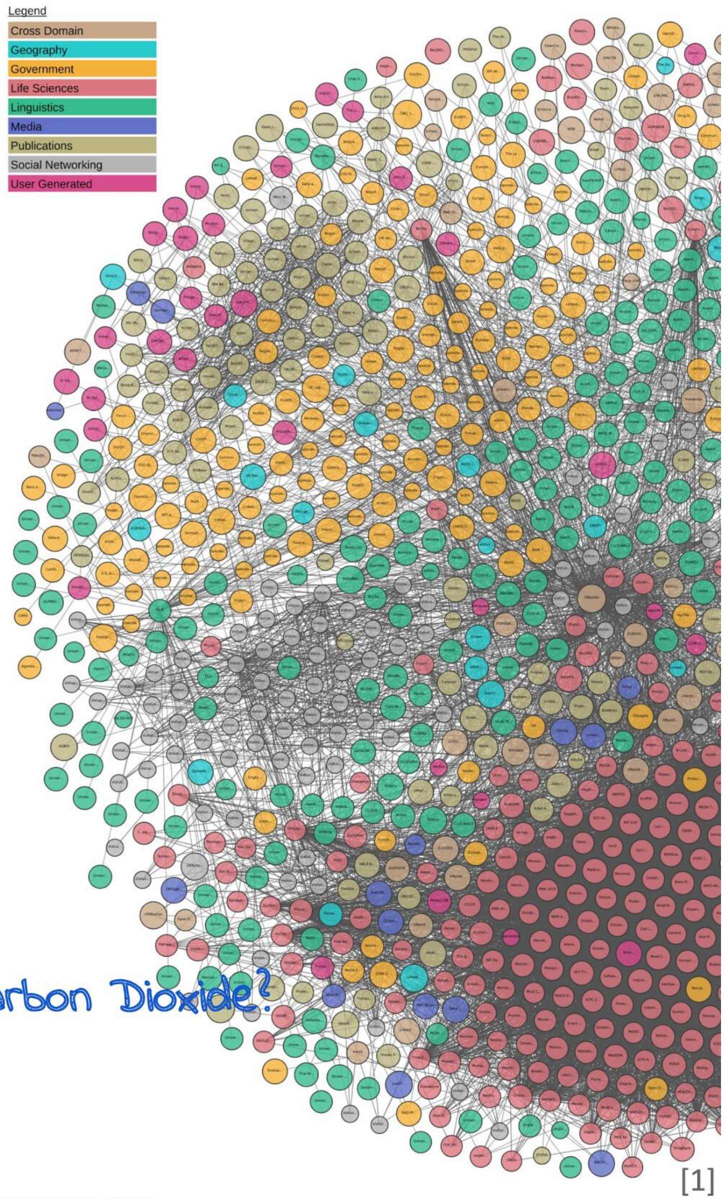
- **Exploratory Search** represents the activities carried out by searchers who are either:
    - **unfamiliar with the domain** of their goal (i.e. need to learn about the topic in order to understand how to achieve their goal),
    - **unsure about the ways** to achieve their goals (either the technology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



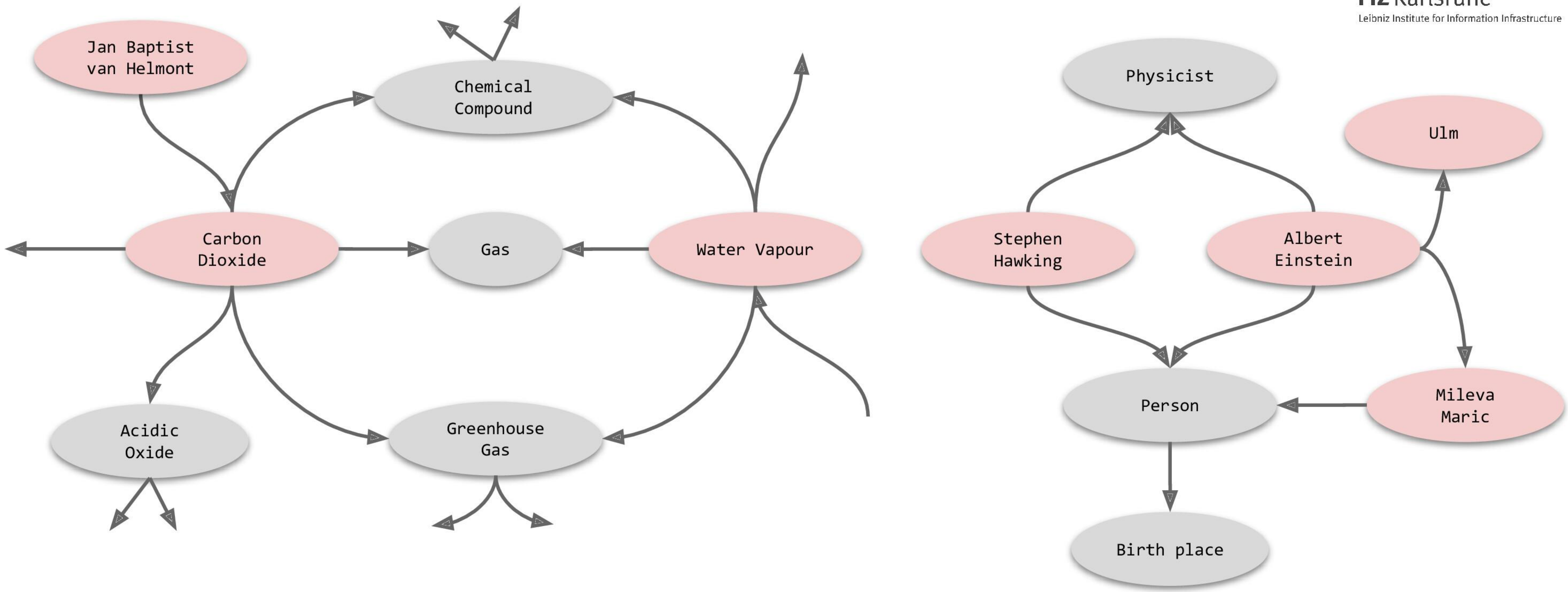
- For search and retrieval systems, **semantic similarity of entities** is an important feature, as e.g.
  - Given an entity find the most similar entities
  - Given an entity find the most similar documents
  - Given a document find the most similar documents, etc.
- **When are two entities (semantically) 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?**

The Linked Open Data Cloud from lod-cloud.net

[1]

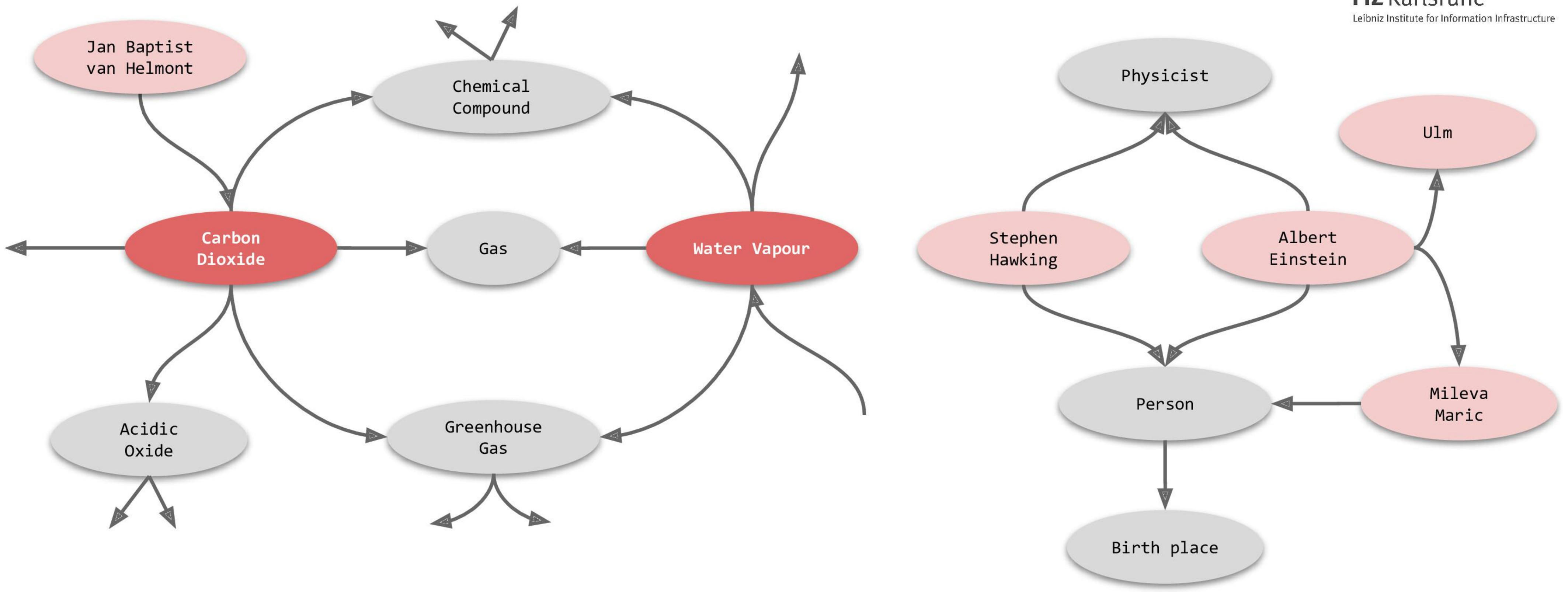
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Semantic Similarity



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

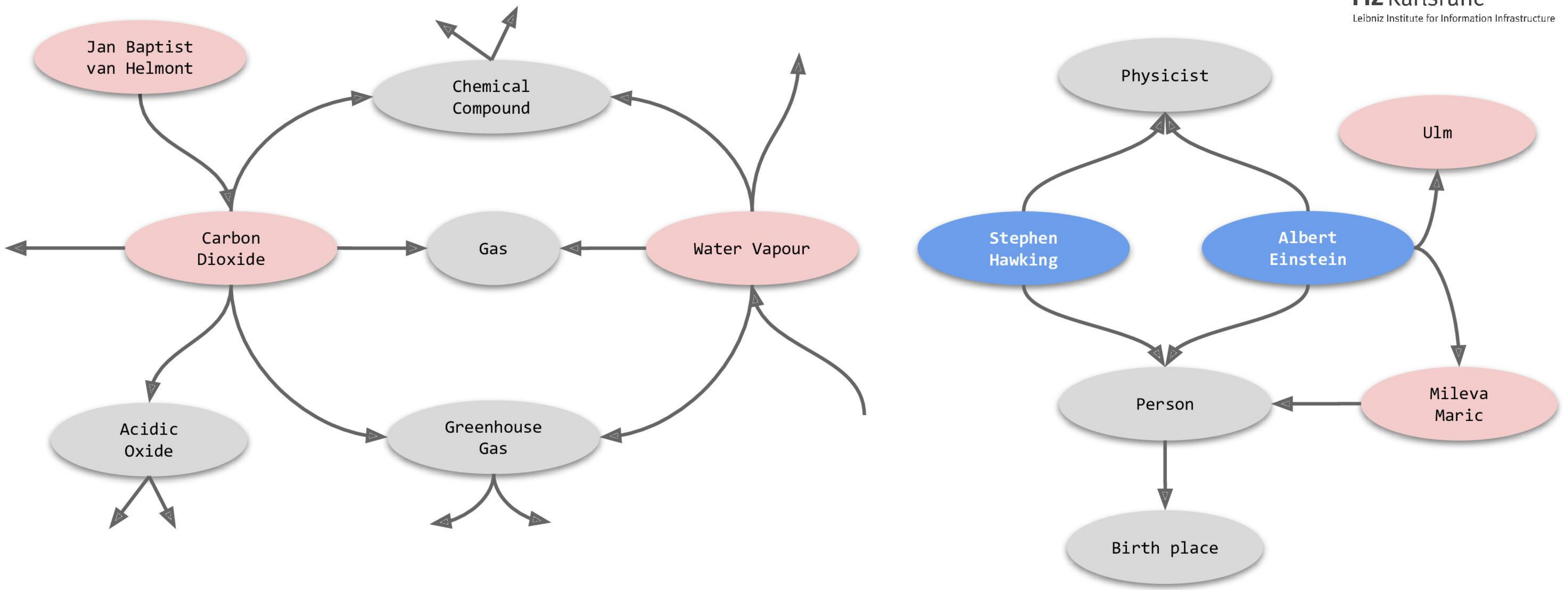
Semantic Similarity



- 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).

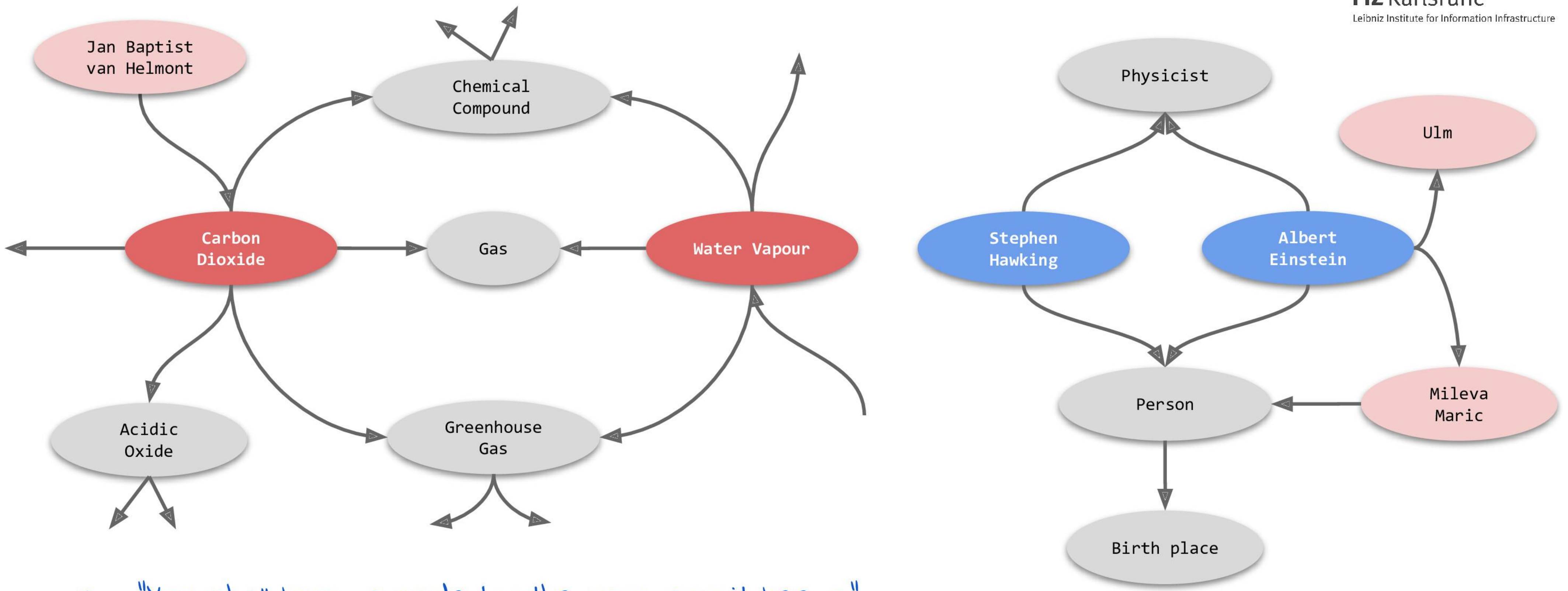
Semantic Similarity



- 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

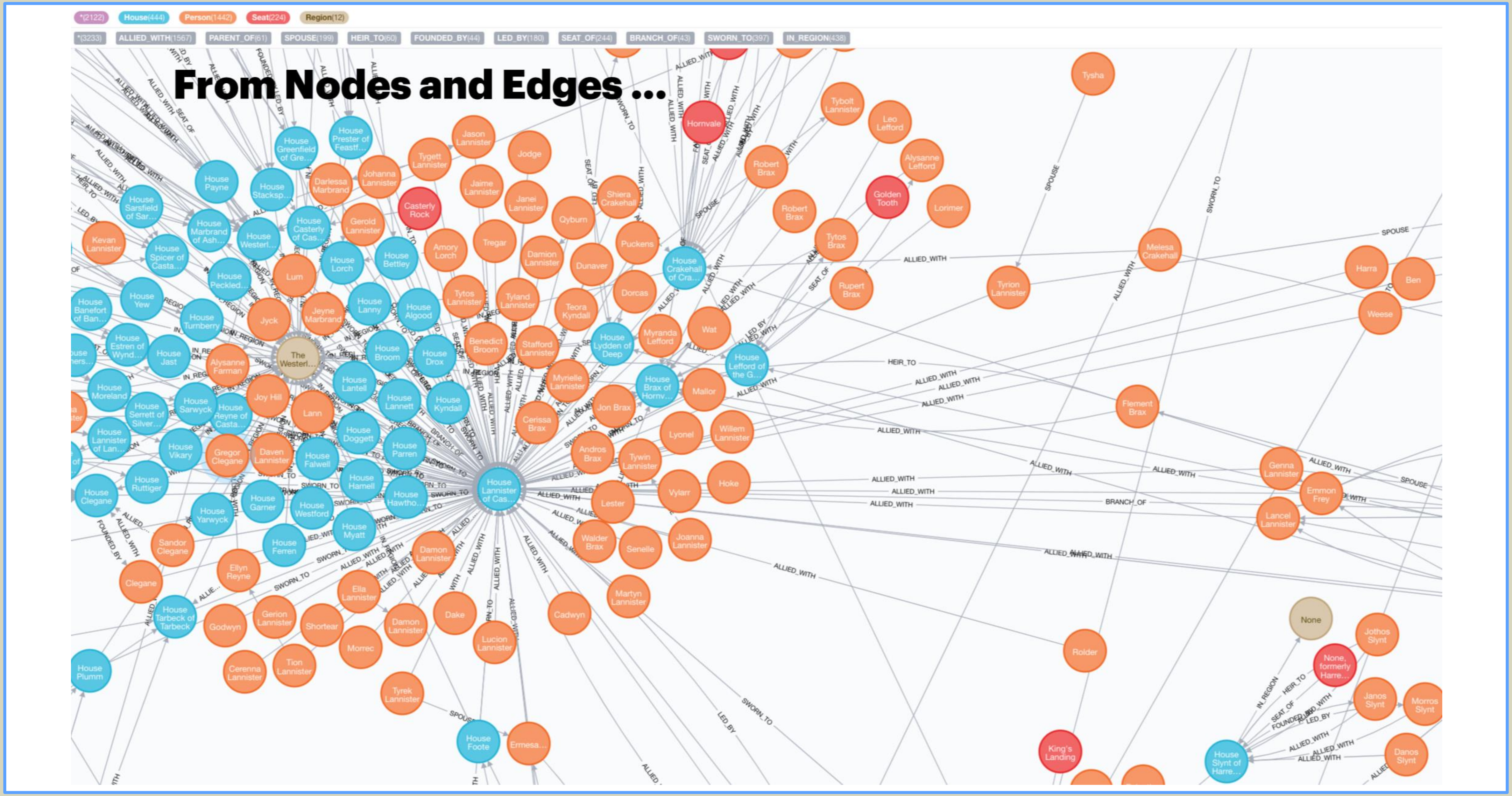
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

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**

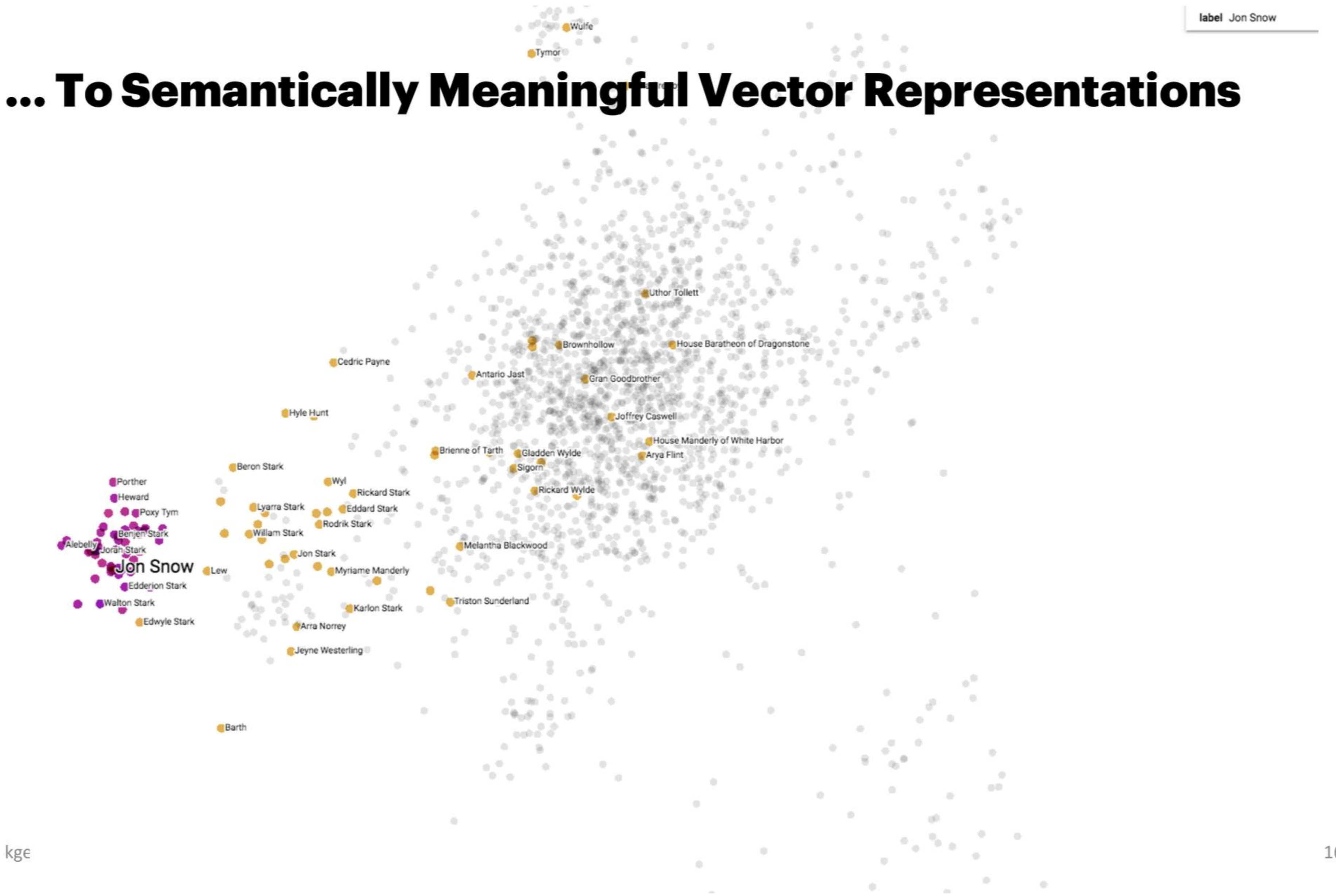
- **Problem:**
  - Algorithms to determine semantic similarity in graphs are of high complexity, i.e. with large KGs, as e.g. Wikidata, they don't work efficiently.

- **Idea**:
  - Approximate the problem by transferring it from graph structures to vector spaces That are easier to handle.



Slide from [Knowledge 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).

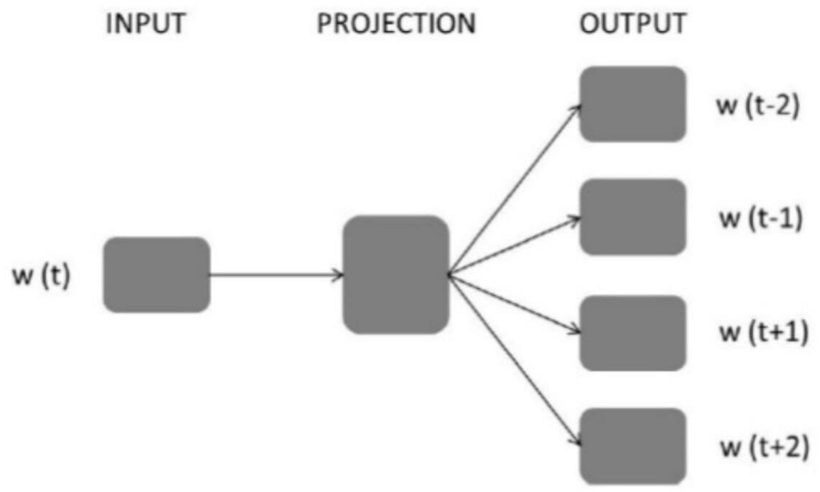
Excursion: Word Embeddings

- **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

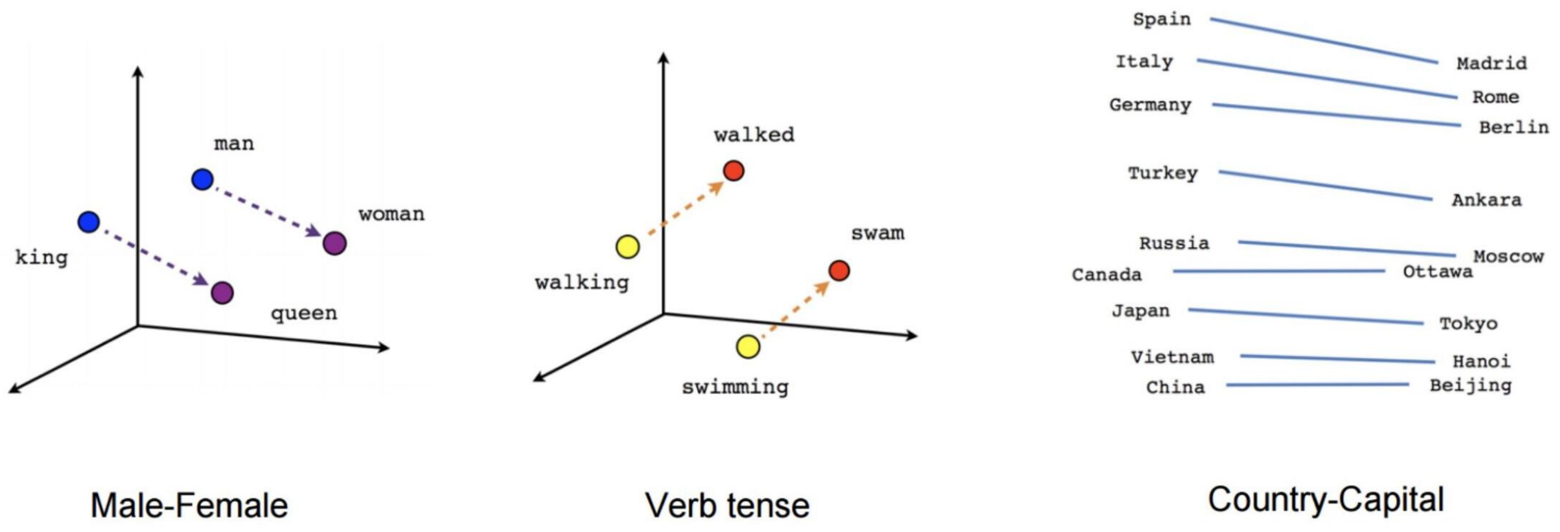
Excursion: Word Embeddings

- **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

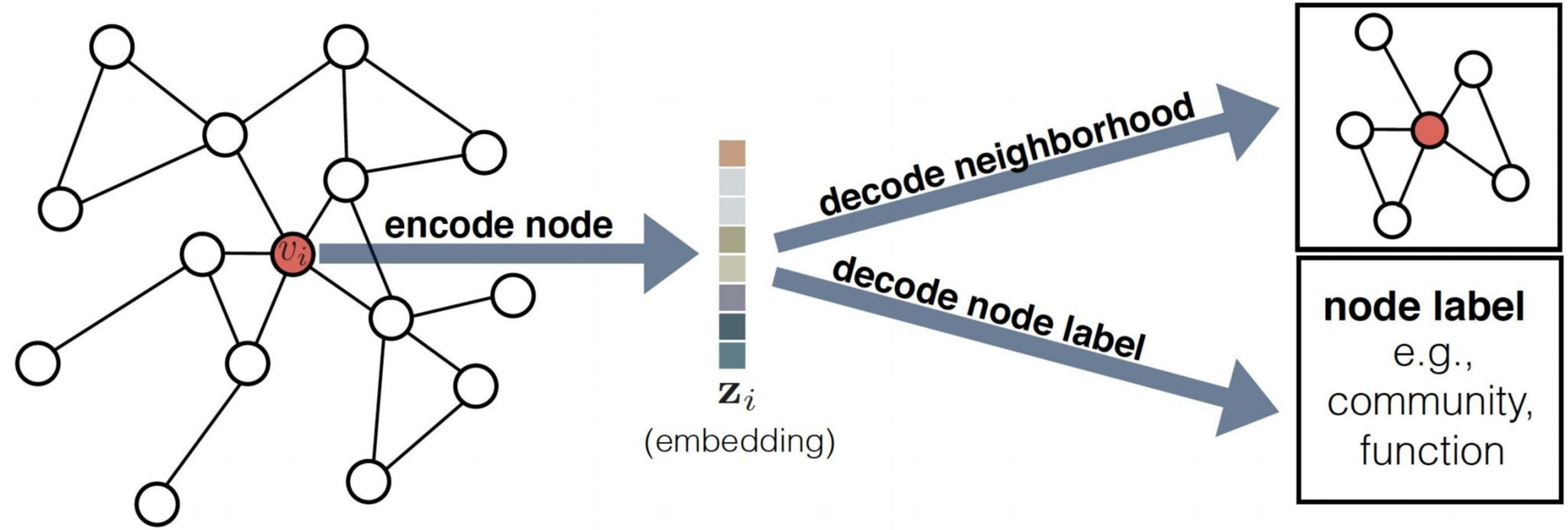
Word Embeddings



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

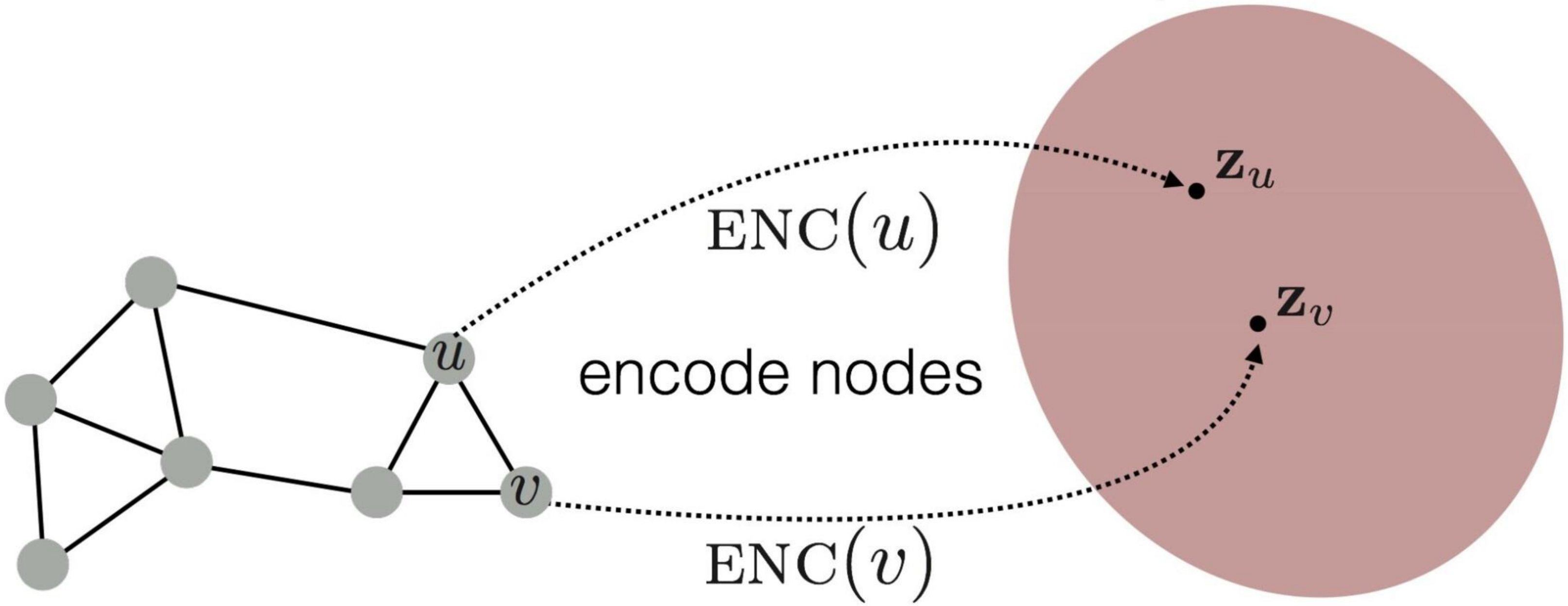
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

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

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

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

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).

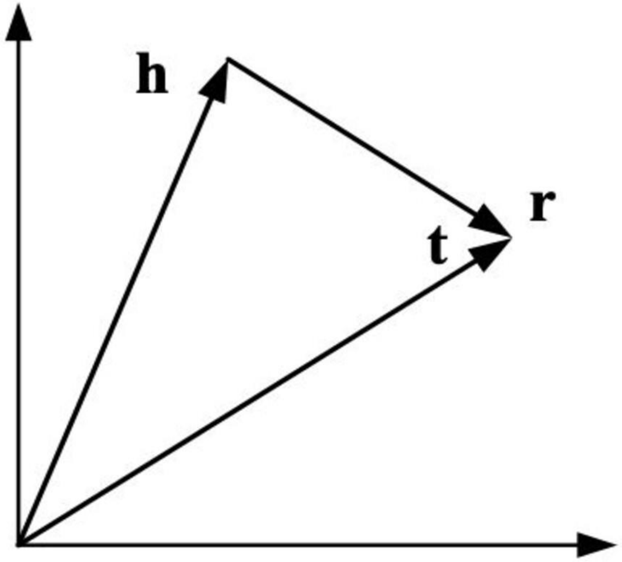
Translational Distance Models

- 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

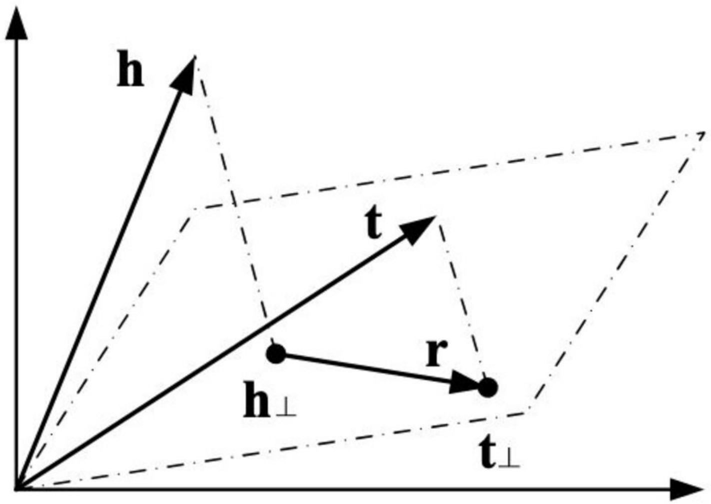


Bordes et al, Translating Embeddings for Modeling Multi-relational Data, NIPS 2013

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.



Entity and Relation Space

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).

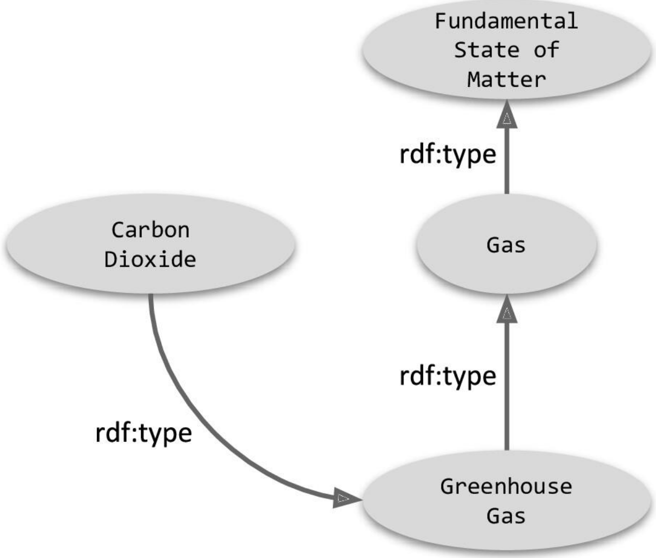
Graph Convolutional Network

- **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

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.
```

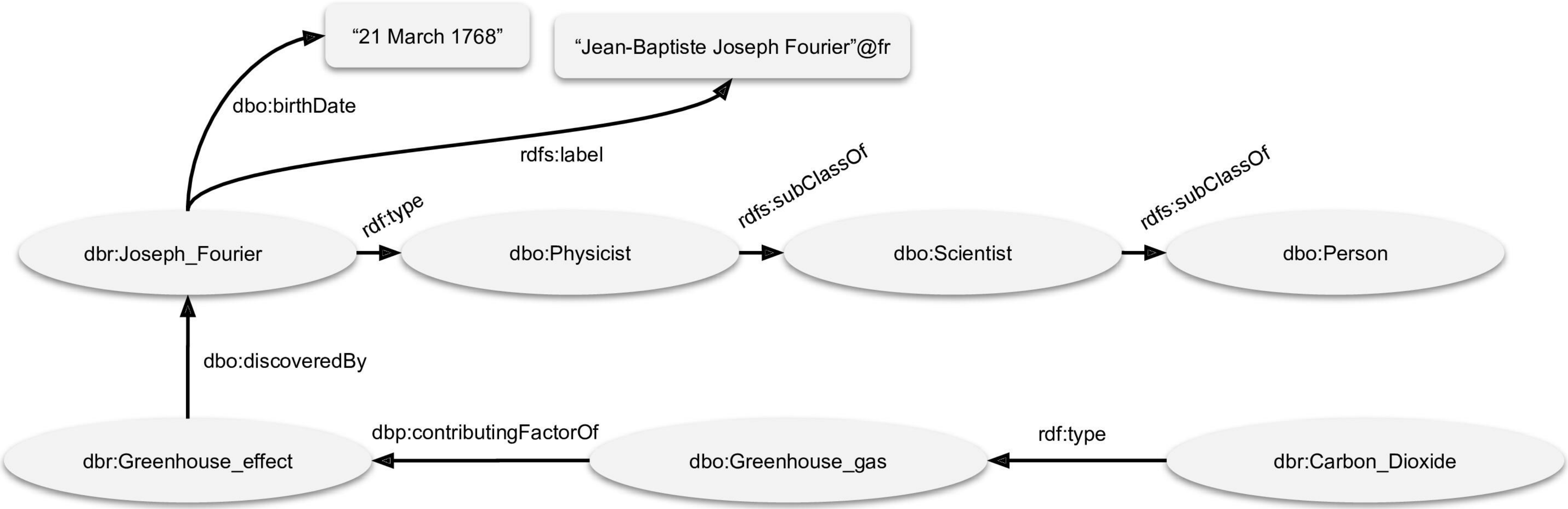


  - Selection strategies:
    - Depth first search
    - Breadth first search
    - Random walk
    - RDF Graph Kernels

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

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

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/

KNOWLEDGE GRAPH COMPLETION

How to guess the missing triples?

Knowledge Graph Refinement

- 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>

Knowledge Graph Completion

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

- 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.

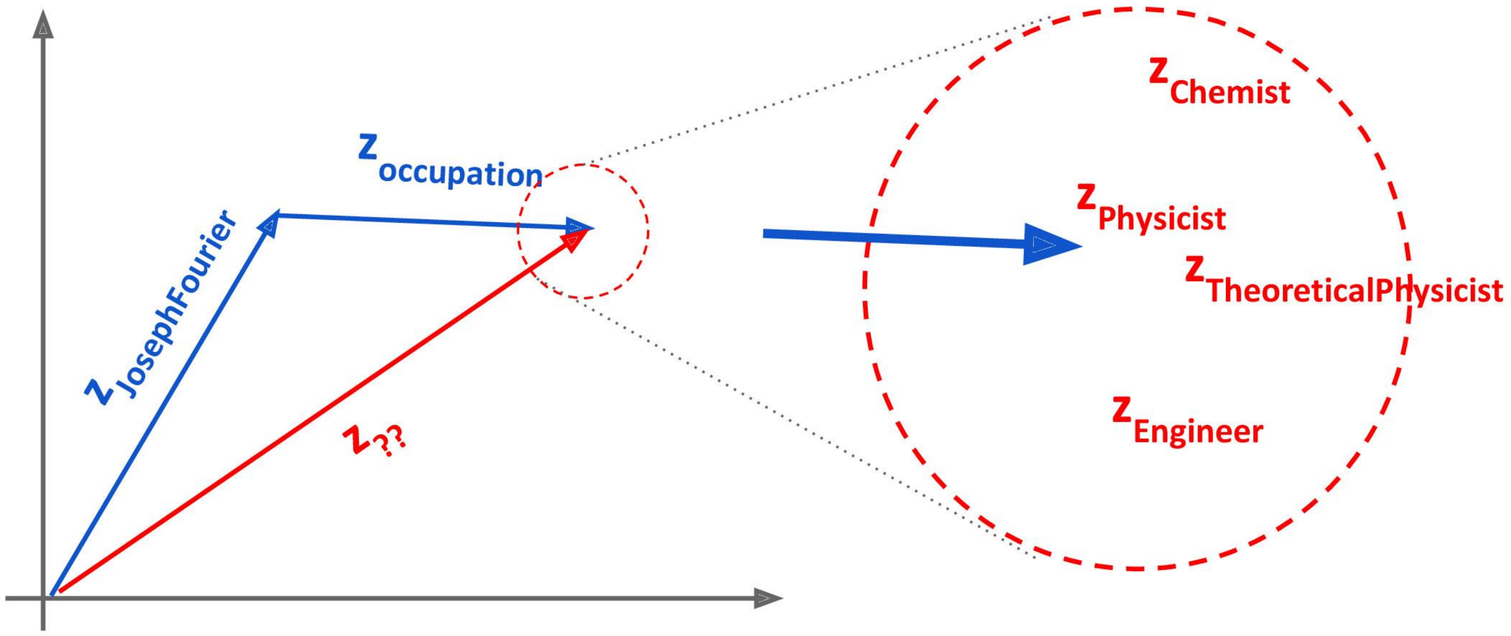
Type Prediction

- **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

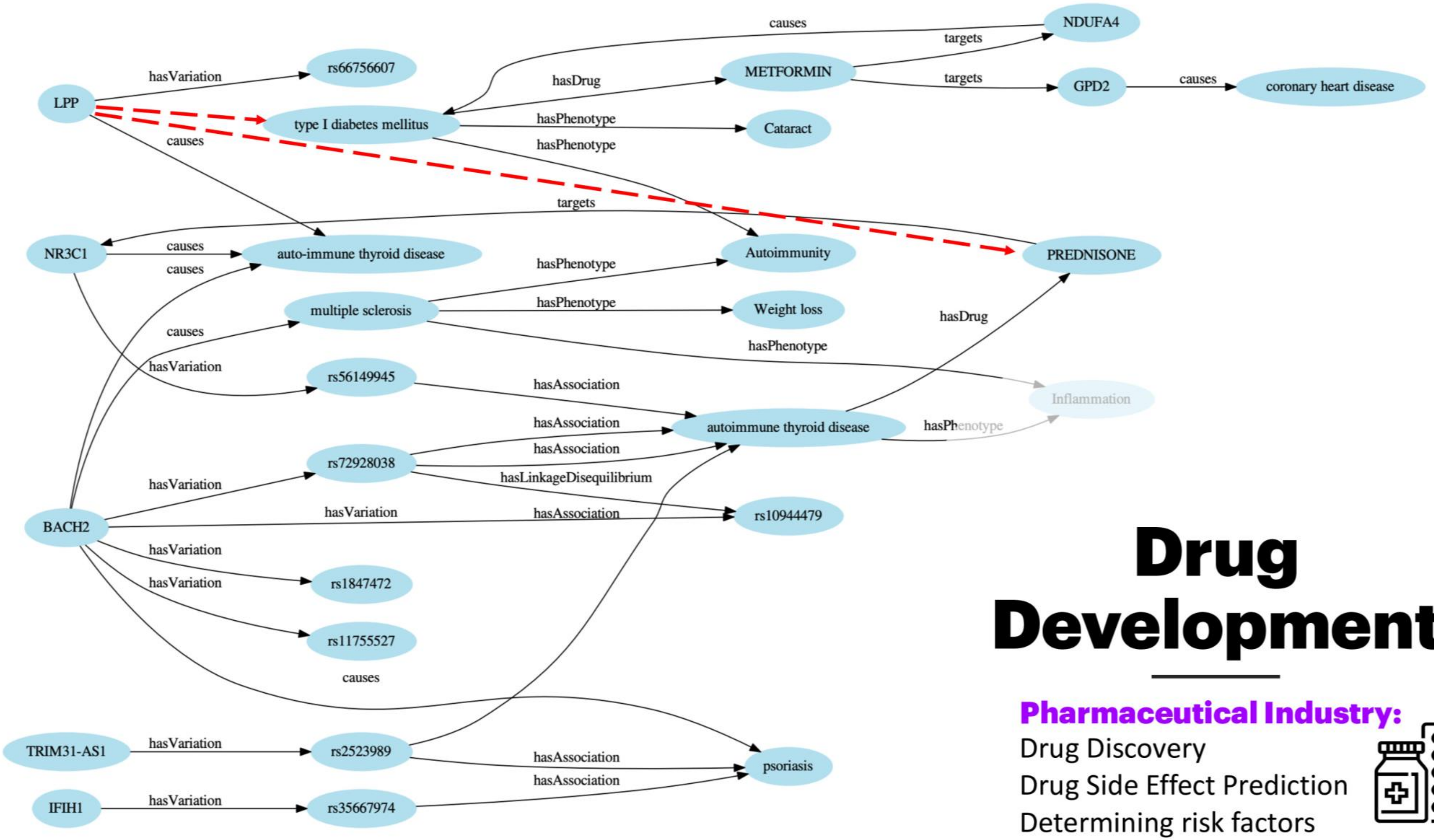
Slide from [Knowledge Graph Embeddings Tutorial] by L. Costabello et al. (ECAI 2020).

Drug Development

- Drug Development is a time consuming and expensive process which ranges from gene identification, identifying a compound to target the gene, and finally experimentation on animals and humans.

- The initial step of identification of gene/drug takes several years and if not identified correctly may result in loss of time and money.
- “Drug Developers” identify the genes/drugs by reading the latest research before proceeding with experimentation. But it is highly dependent on the experience of the person.

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



Drug Development

Pharmaceutical Industry:

Drug Discovery

Drug Side Effect Prediction

Determining risk factors

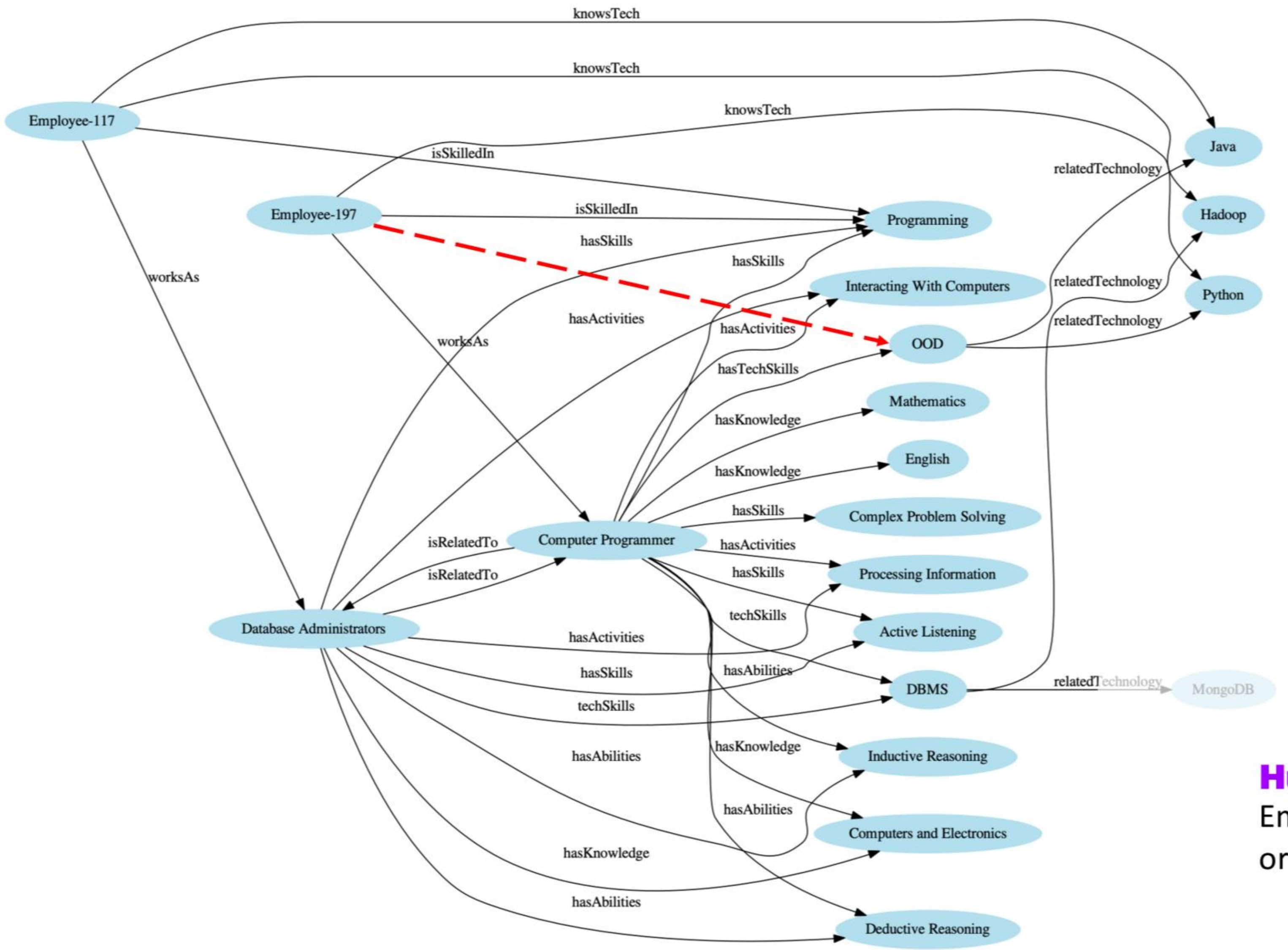
Slide from Knowledge Graph Embeddings Tutorial by L. Costabello et al. (ECAI 2020).

Human Resource

- Technology is evolving at an extremely fast pace. People need to learn new skills to be relevant in the market.
- Due to automation, a lot of roles are becoming obsolete and companies are forced to lay off people.

KGEs can be used for following tasks:

- Suggest new technology/tasks for career progression.
- Recommend similar roles within the organization when existing role becomes obsolete.



Human Resource

Human Resources:
Employee Career Progression or Transition

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

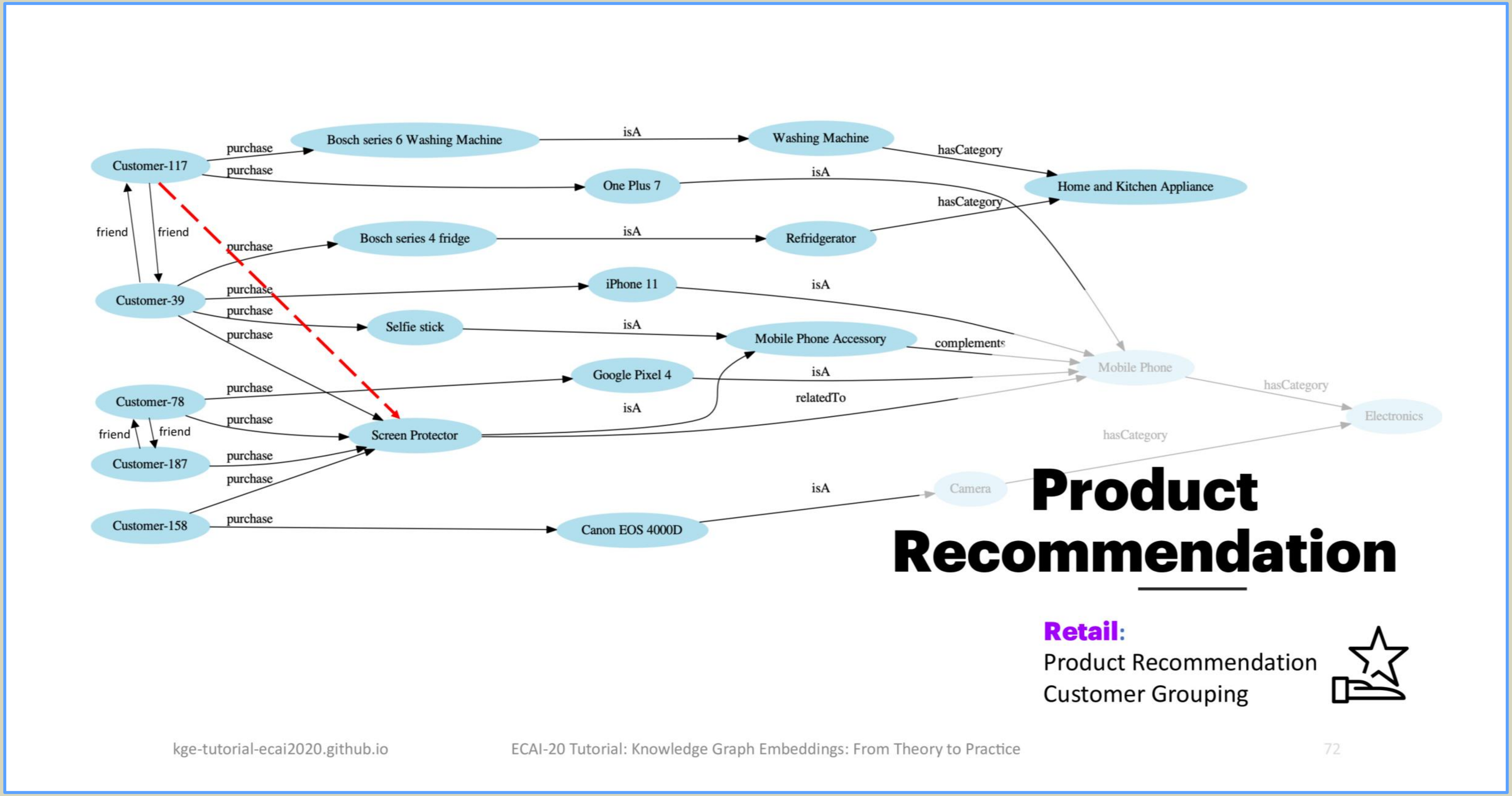
Product Recommendation

KGEs can leverage relation between customers and products.

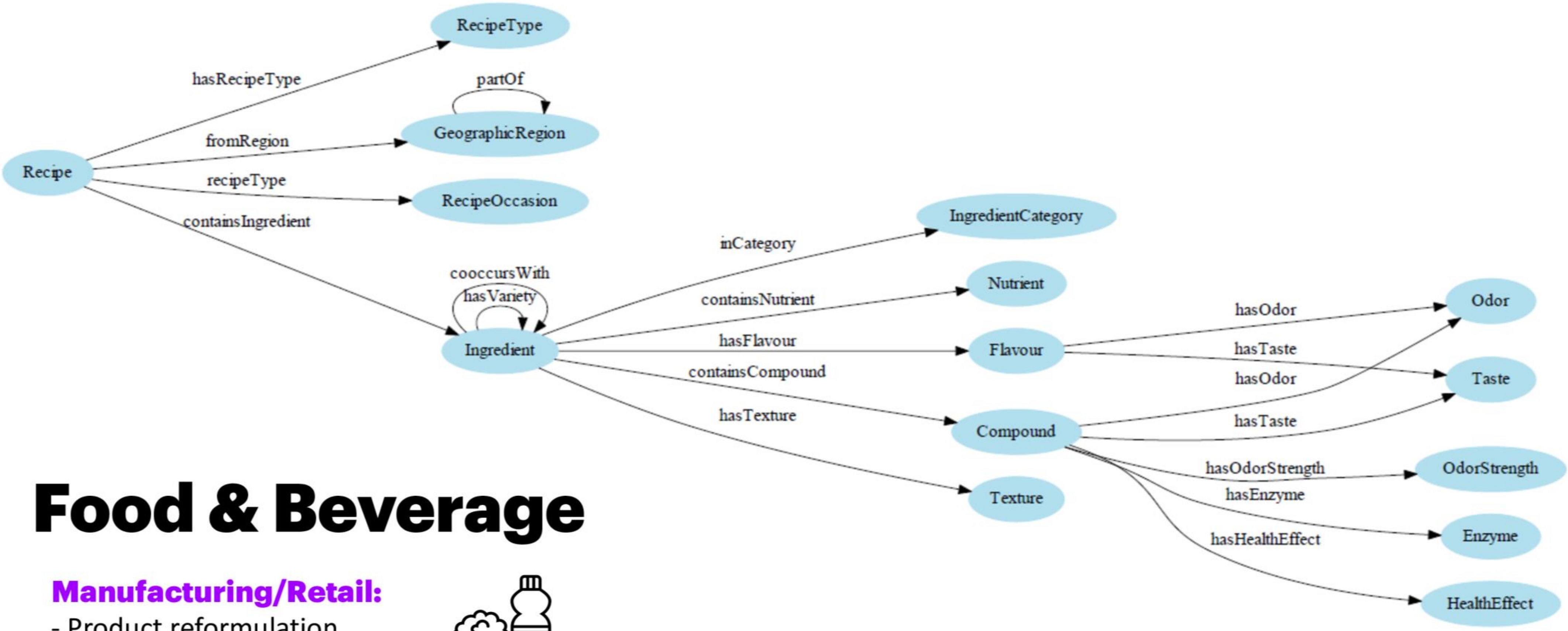
KGEs can be used for following tasks:

- Recommend new products to customers
- Group customers based on their purchase history

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



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

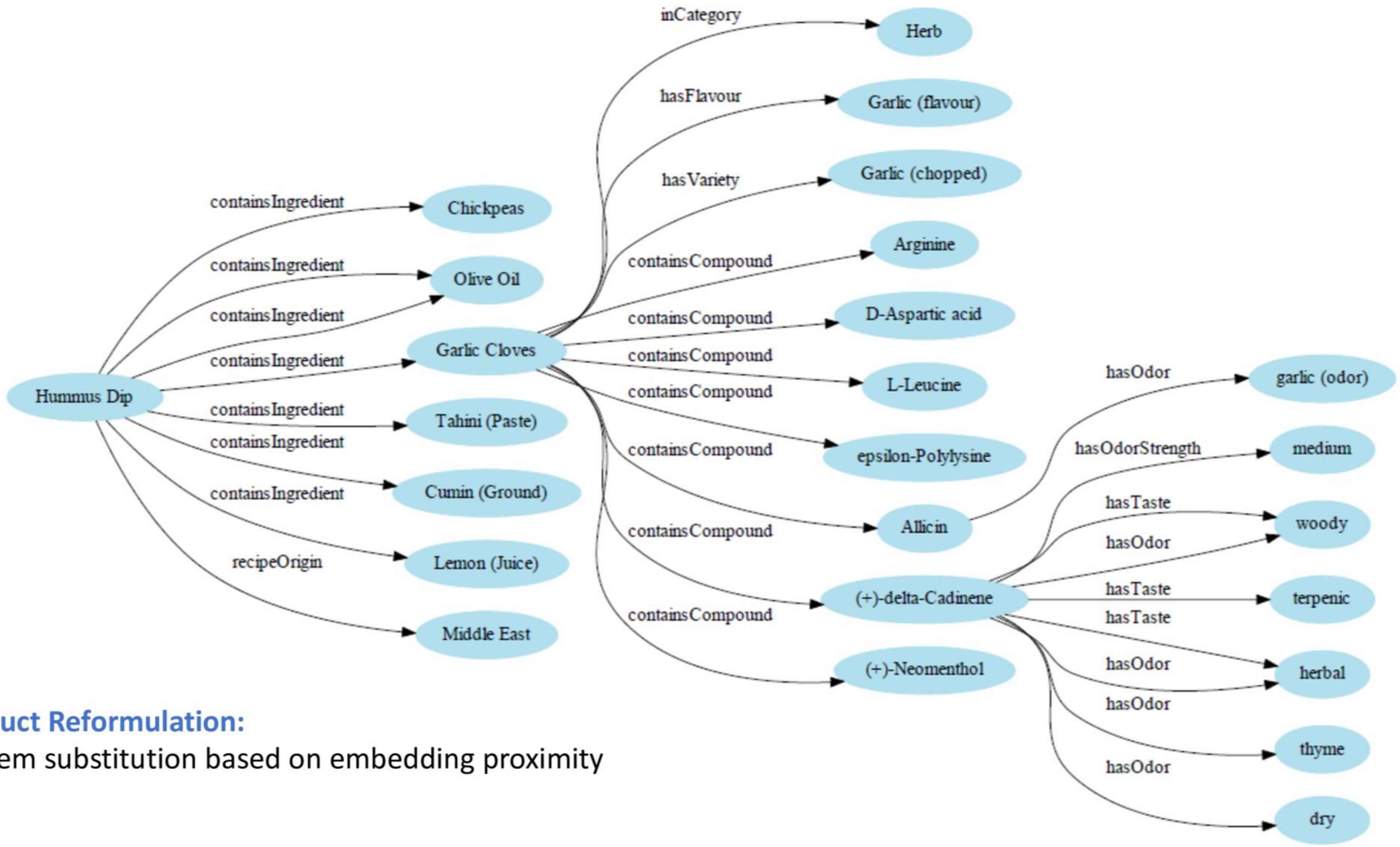


Food & Beverage

Manufacturing/Retail:

- Product reformulation
- Adapting to consumer trends

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



Product Reformulation:

- Item substitution based on embedding proximity

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

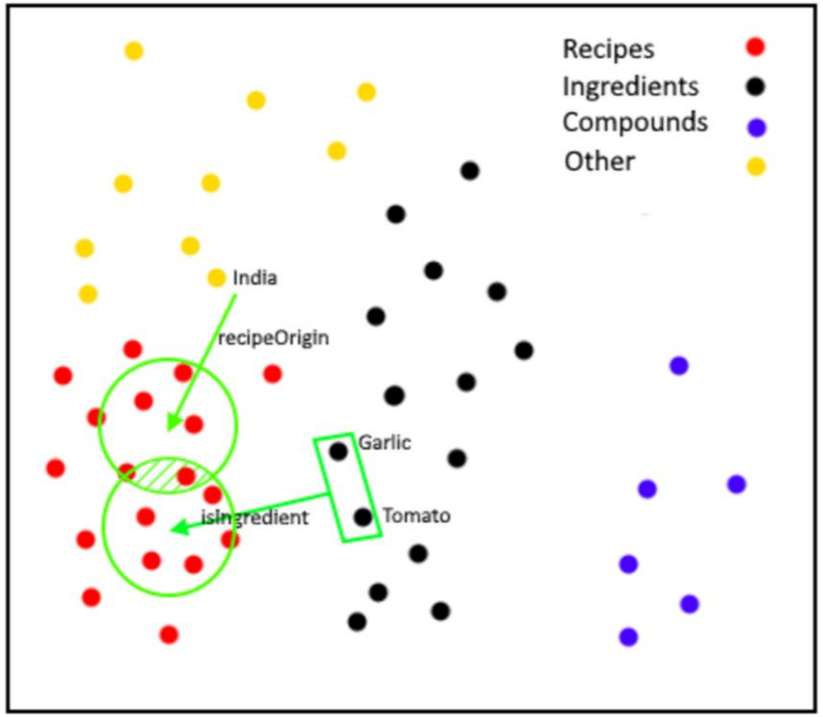
Item Recommendation

- Use vector algebra to find latent region that satisfy input criteria

- Example:
  - *"I want Indian recipes that contain garlic and tomato"*
  - $nearest(\ avg\ (avg(GARLIC, TOMATO) - containsIngredient, India\ - recipeOrigin))$

- *Note: the above is pseudo-code, actual solutions will depend on model, data, fine-tuning, etc*

- Alternatively use Bayesian optimization ..



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

CHATGPT IS A BULLSHIT

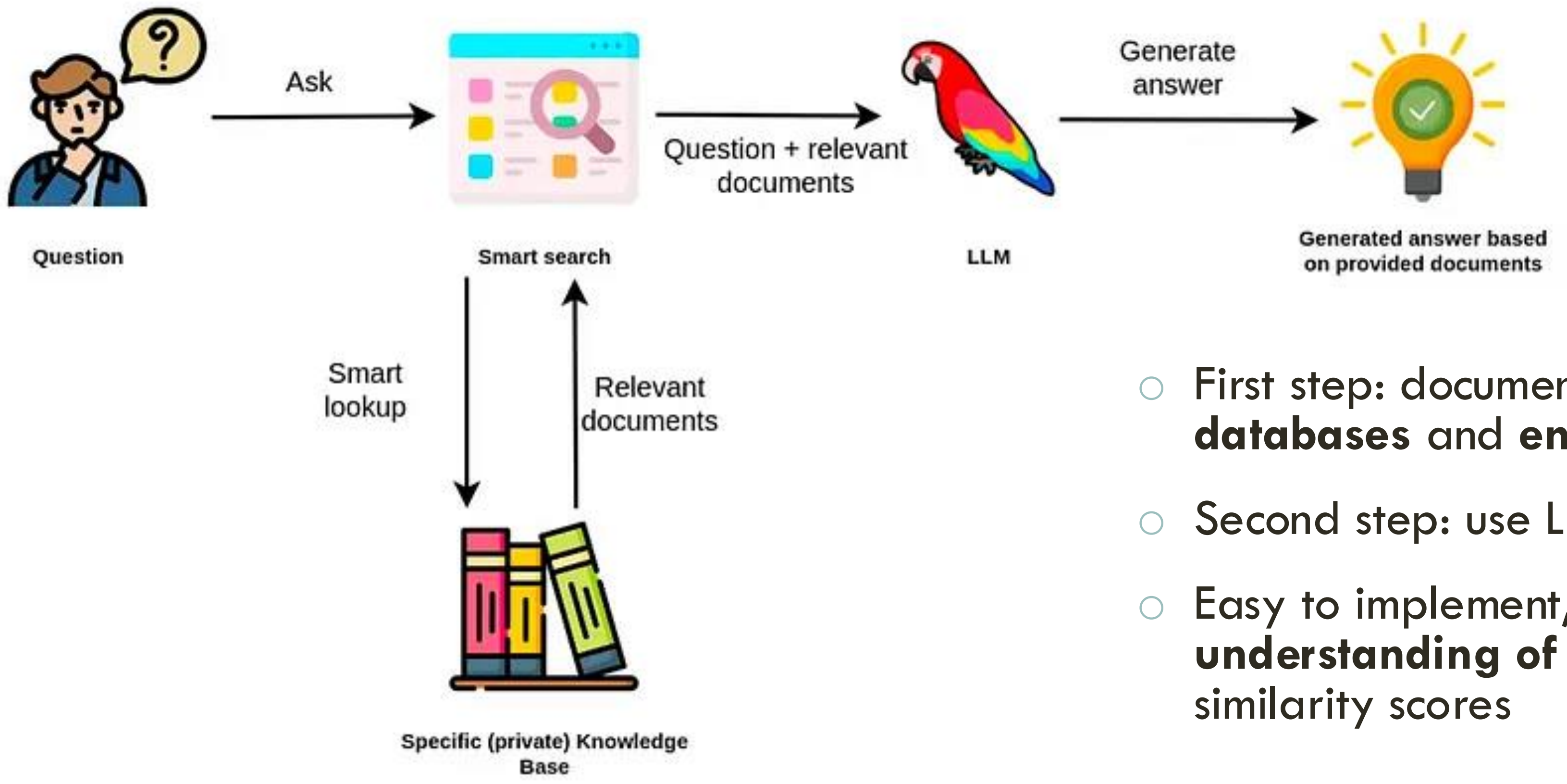
How can we fix it?

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.

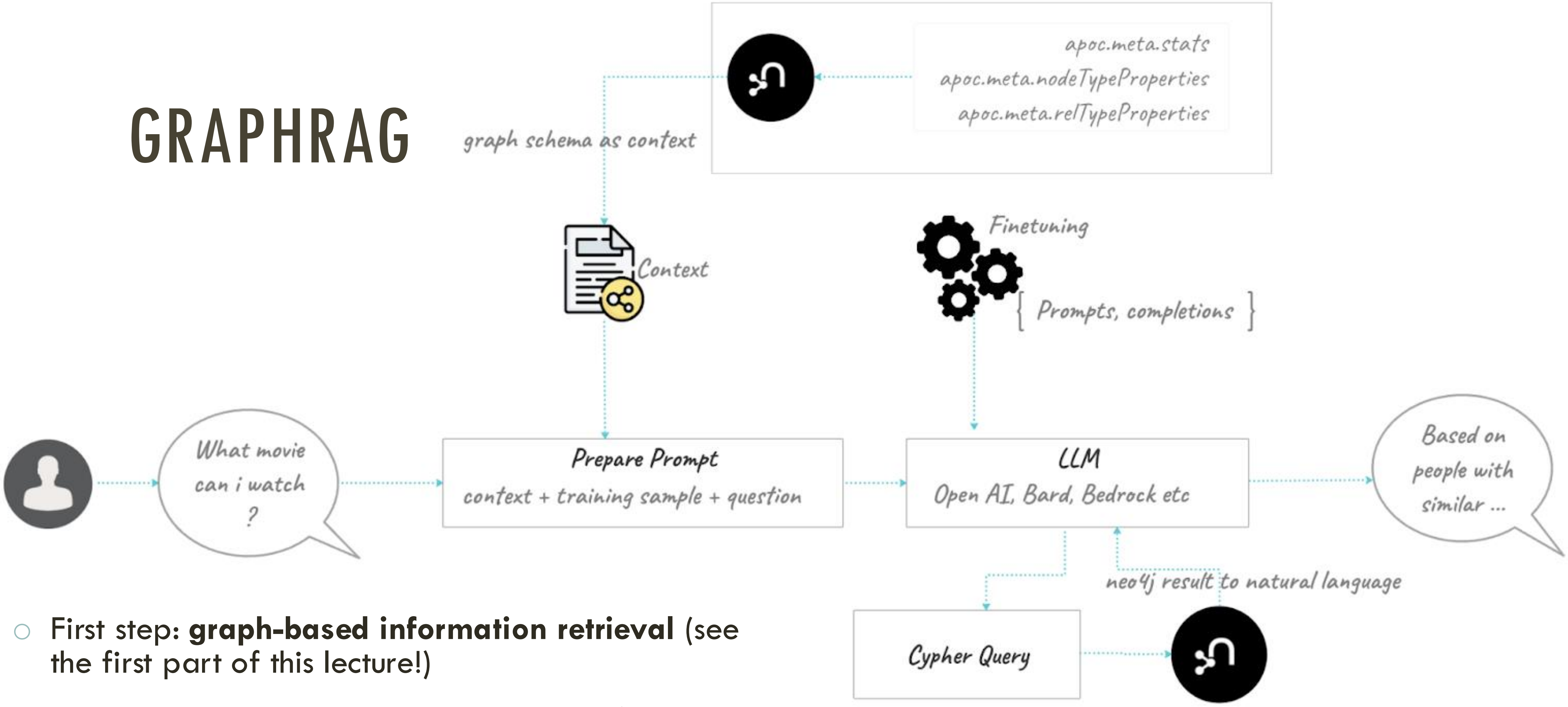
RETRIEVAL-AUGMENTED GENERATION (RAG)



- 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]

OUTLINE

1. Knowledge graphs
2. Towards automated KG management
3. Semantic search and recommendations
4. Knowledge graph embeddings
5. Knowledge graph completion
6. ChatGPT is a bullshit. How can we fix it?

THANK YOU FOR YOUR ATTENTION!

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

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