universität innsbruck

Department of Computer Science

STI INNSBRUCK

Building a Knowledge Graph from schema.org annotations

KGC 2020, Tutorial

Elias Kärle & Umutcan Simsek

STI2, University of Innsbruck, May the 4th (be with you), 2020

@eliaska
@umutsims

About Us

Elias Kärle
PhD Student
elias.kaerle@sti2.at
Twitter: @eliaska
https://elias.kaerle.com

Umutcan Simsek
PhD Student
umutcan.simsek@sti2.at
Twitter: @umutsims
http://umutcan.eu

Acknowledgements

This tutorial is based on the work being done in the MindLab, an industrial research project for building knowledge graphs to be consumed by conversational agents in domains like tourism. A version of this tutorial was given in SEMANTICS 2019 in Karlsruhe, Germany.

An extensive version of the content of this tutorial can be found in the book "*Knowledge Graphs - Methodology, Tools and Selected Use Cases*"

https://www.knowledgegraphbook.ai/

mindLab
https://mindlab.ai

feratel window to the world
ONLIM
universität innsbruck

Tutorial website:
https://stiinnsbruck.github.io/kgs/

About the Tutorial

The tutorial aims to introduce our take on the knowledge graph lifecycle

For Industry Practitioners

An entry point to Knowledge Graphs with concrete and practical examples

For Academics

A brief overview of the literature, introduction of several tools

https://mindlab.ai/en/publications/ - An extensive list of reading suggestions

Agenda

1. 09:00 - 10:30 Intro & Knowledge Creation

   10:30 - 11:00 Break

2. 11:00 - 12:00 Knowledge Hosting, Curation & Deployment
3. 12:00 - 13:00 free hands-on session

Hands-on and Discussion: #tutorial-building-a-kg-from-schema-dot-org (bring your own coffee)

Outline

1. What is a Knowledge Graph
2. Knowledge Creation
3. Knowledge Hosting
4. Knowledge Curation
5. Knowledge Deployment
6. Outlook

1. WHAT IS A KNOWLEDGE GRAPH?

1. What is a Knowledge Graph?

TL;DR:

very large semantic nets that integrate various and heterogeneous information sources

to represent knowledge about certain domains of discourse.

1. What is a Knowledge Graph?

Why are Knowledge Graphs something new and cool?

Name	Instances	Facts	Types	Relations
DBpedia (English)	4,806,150	176,043,129	735	2,813
YAGO	4,595,906	25,946,870	488,469	77
Freebase	49,947,845	3,041,722,635	26,507	37,781
Wikidata	15,602,060	65,993,797	23,157	1,673
NELL	2,006,896	432,845	285	425
OpenCyc	118,499	2,413,894	45,153	18,526
Google´s Knowledge Graph	570,000,000	18,000,000,000	1,500	35,000
Google´s Knowledge Vault	45,000,000	271,000,000	1,100	4,469
Yahoo! Knowledge Graph	3,443,743	1,391,054,990	250	800

Knowledge Graphs in the Wild [Paulheim, 2017]

1. What is a Knowledge Graph?

What makes Knowledge Graphs cool is also their curse…

Integration of data from heterogeneous sources can cause quality issues

The assessment of quality and its improvement is called **Knowledge Curation**

1. What is a Knowledge Graph?

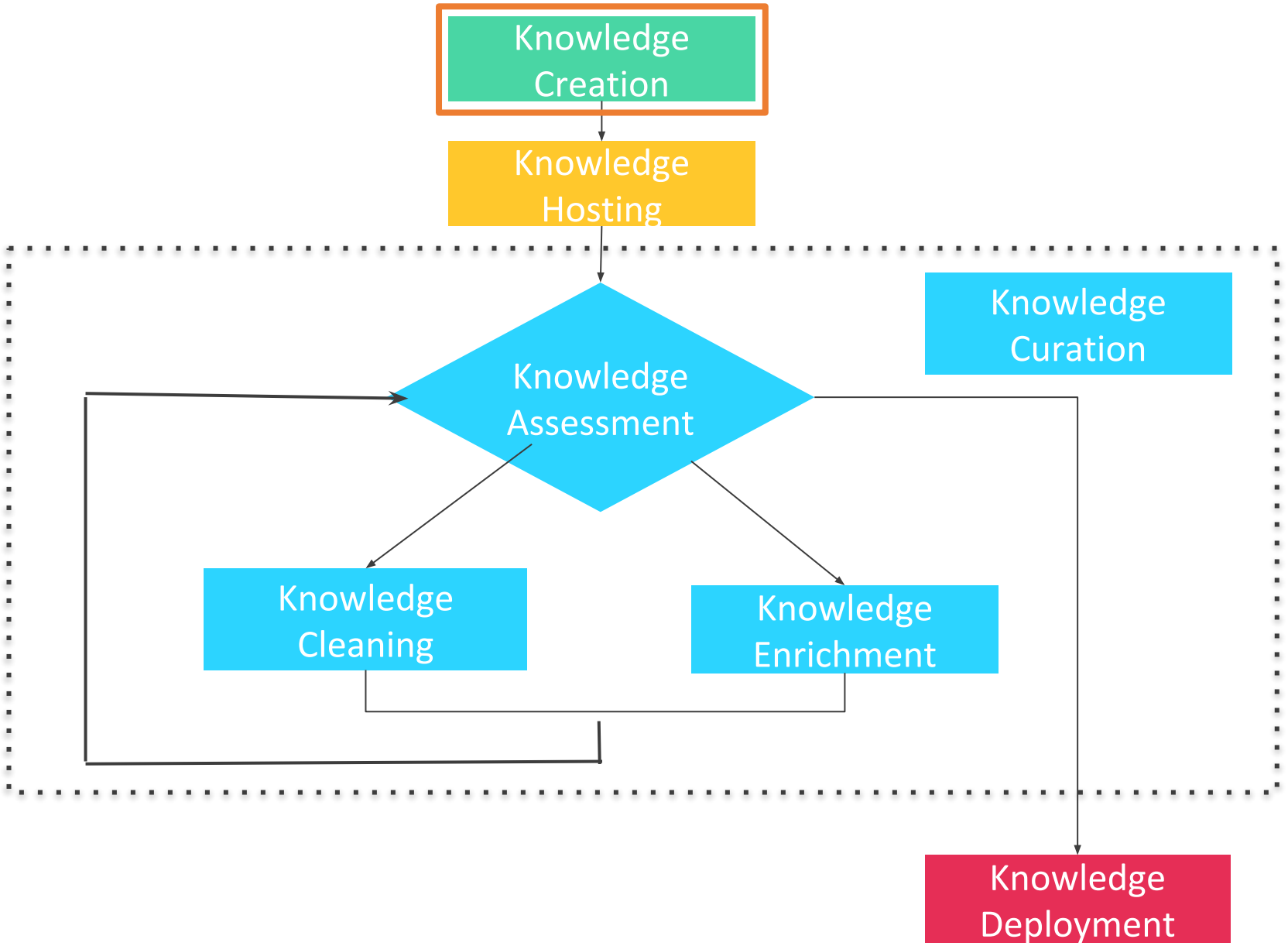
Two main entry points for improving the quality of knowledge graphs:

Fixing the vocabulary

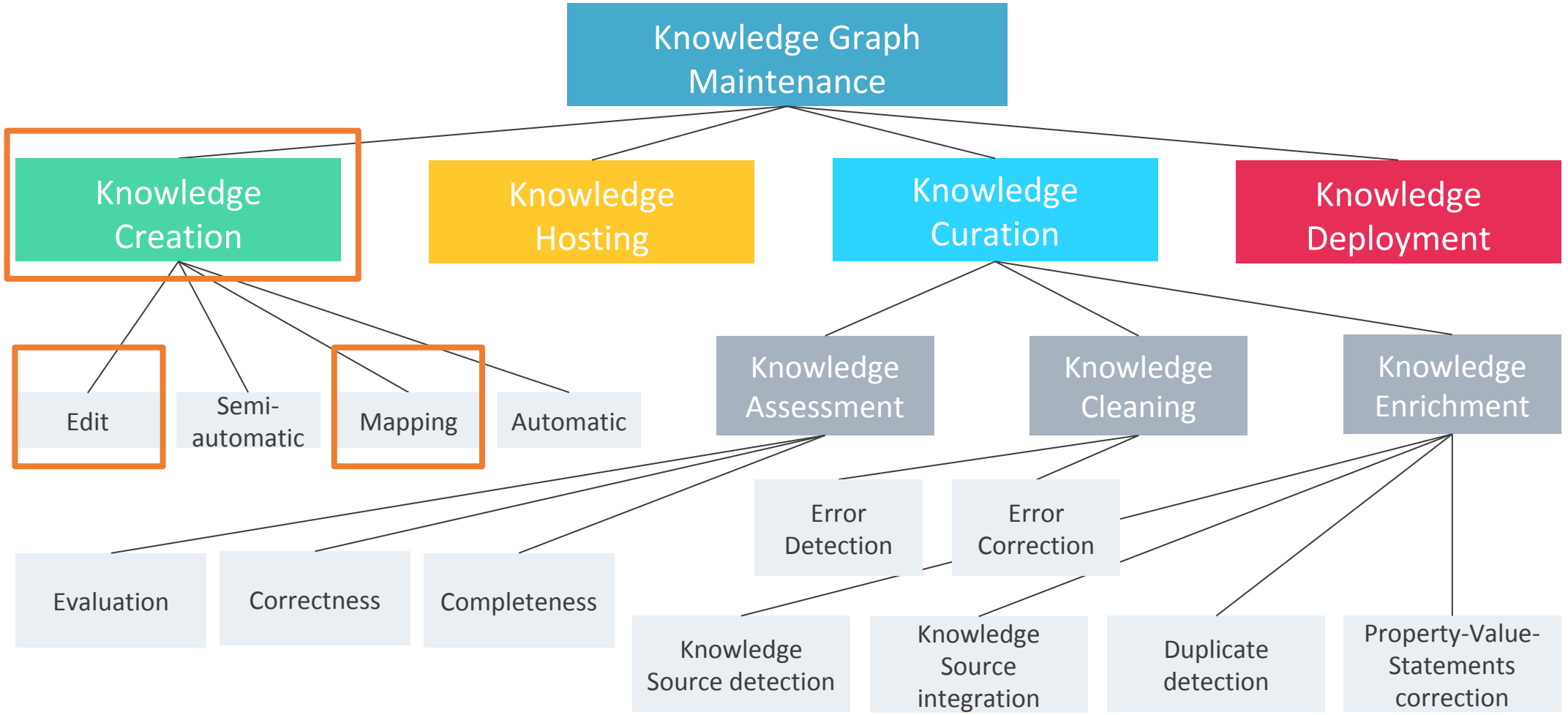
- We accept schema.org (and its extensions) as golden standard.

Fixing the facts

- This is where knowledge curation comes in.

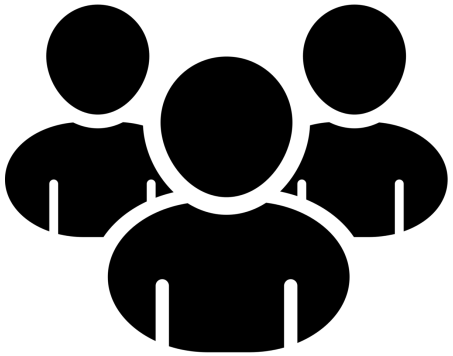

Knowledge Creation
Knowledge Hosting
Knowledge Curation
Knowledge Assessment
Knowledge Cleaning
Knowledge Enrichment
Knowledge Deployment

2. KNOWLEDGE CREATION


Knowledge Graph Maintenance
Knowledge Creation
Knowledge Hosting
Knowledge Curation
Knowledge Deployment
Edit
Semi-automatic
Mapping
Automatic
Knowledge Assessment
Knowledge Cleaning
Knowledge Enrichment
Evaluation
Correctness
Completeness
Error Detection
Error Correction
Knowledge Source detection
Knowledge Source integration
Duplicate detection
Property-Value-Statements correction

2. Knowledge Creation: Schema.org as Golden Standard

bing Google yahoo! Yandex



schema.org

2. Knowledge Creation: Schema.org as Golden Standard

schema.org

- Language to **describe** „Things“ on the Web
- wide distribution on the Web
- direct and “invisible” integration in websites
  - Microdata
  - RDFa
  - JSON-LD

```
<div vocab="http://schema.org/" typeof="Movie">
  <h1 property="name">Avatar</h1>
  <div property="director" typeof="Person">
  Director: <span property="name">James Cameron</span>
(born <time property="birthDate" datetime="1954-08-16">August 16, 1954</time>)
  </div>
  <span property="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html
</div>
```

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name">Avatar</h1>
  <div itemprop="director" itemscope itemtype="http://schema.org/Person">
  Director: <span itemprop="name">James Cameron</span>
(born <time itemprop="birthDate" datetime="1954-08-16">August 16, 1954</ti
  </div>
  <span itemprop="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html" itemprop="trailer">Tr
</div>
```

```
<script type="application/ld+json">
{
  "@context": "http://schema.org/",
  "@type": "Movie",
  "name": "Avatar",
  "director":
    {
       "@type": "Person",
       "name": "James Cameron",
       "birthDate": "1954-08-16"
    },
  "genre": "Science fiction",
  "trailer": "../movies/avatar-theatrical-trailer.html"
}
</script>
```

2. Knowledge Creation: Schema.org

schema.org

LandmarksOrHistoricalBuildings

Thing > Place > LandmarksOrHistoricalBuildings

An historical landmark or building.

[more...]

TouristAttraction

Thing > Place > TouristAttraction

A tourist attraction. In principle any Thing can be a TouristAttraction, from a Mountain and LandmarksOrHistoricalBuildings to a LocalBusiness. This Type can be used on its own to describe a general TouristAttraction, or be used as an additionalType to add tourist attraction properties to any other type. (See examples below)

[more...]

Event

Thing > Event

An event happening at a certain time and location, such as a concert, lecture, or festival. Ticketing information may be added via the offers property. Repeated events may be structured as separate Event objects.

[more...]

schema.org

Custom Search

Home | Schemas | Documentation

Hotel

Thing > Organization > LocalBusiness > LodgingBusiness > Hotel
Thing > Place > LocalBusiness > LodgingBusiness > Hotel

A hotel is an establishment that provides lodging paid on a short-term basis (Source: Wikipedia, the free encyclopedia, see http://en.wikipedia.org/wiki/Hotel).

See also the dedicated document on the use of schema.org for marking up hotels and other forms of accommodations.

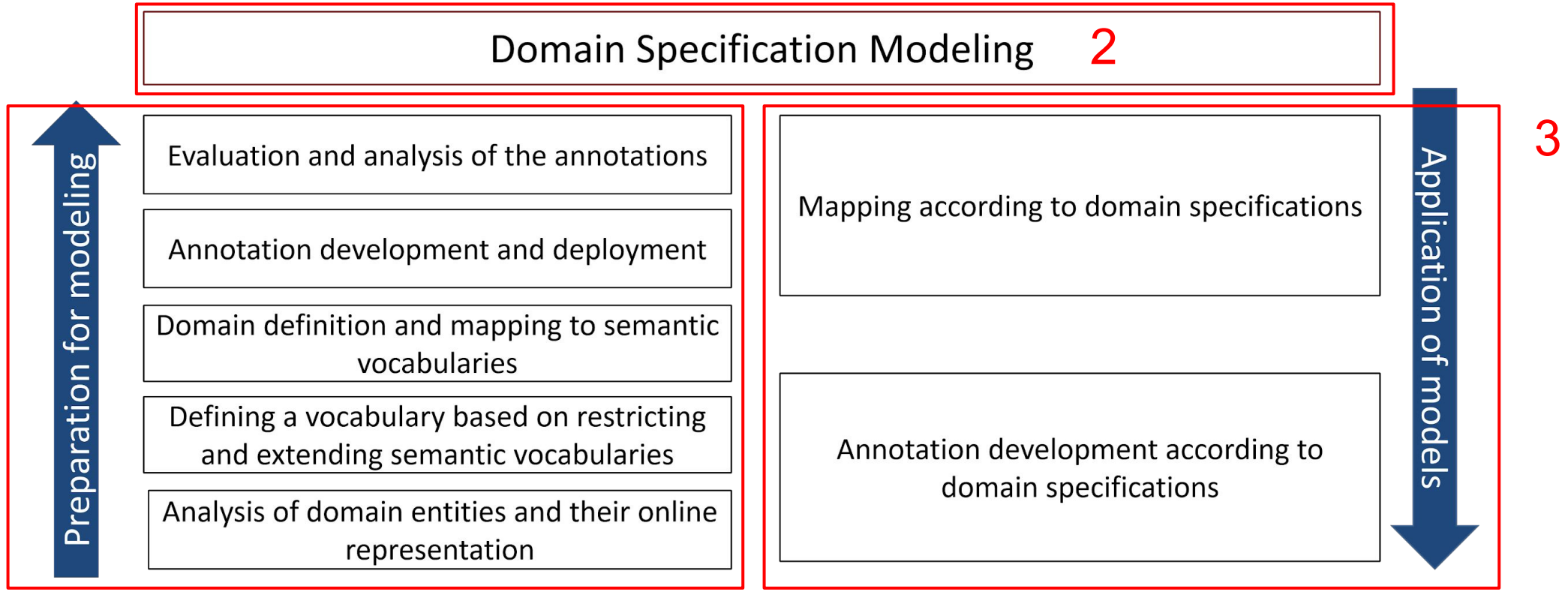
[more...]

Property	Expected Type	Description
Properties from LodgingBusiness		
amenityFeature	LocationFeatureSpecification	An amenity feature (e.g. a characteristic or service) of the Accommodation. This generic property does not make a statement about whether the feature is included in an offer for the main accommodation or available at extra costs.
audience	Audience	An intended audience, i.e. a group for whom something was created. Supersedes serviceAudience.
availableLanguage	Language or Text	A language someone may use with or at the item, service or place. Please use one of the language codes from the IETF BCP 47 standard. See also inLanguage
checkinTime	DateTime or Time	The earliest someone may check into a lodging establishment.
checkoutTime	DateTime or Time	The latest someone may check out of a lodging establishment.
numberOfRooms	Number or QuantitativeValue	The number of rooms (excluding bathrooms and closets) of the accommodation or lodging business. Typical unit code(s): ROM for room or C62 for no unit. The type of room can be put in the unitText property of the QuantitativeValue.
petsAllowed	Boolean or Text	Indicates whether pets are allowed to enter the accommodation or lodging business. More

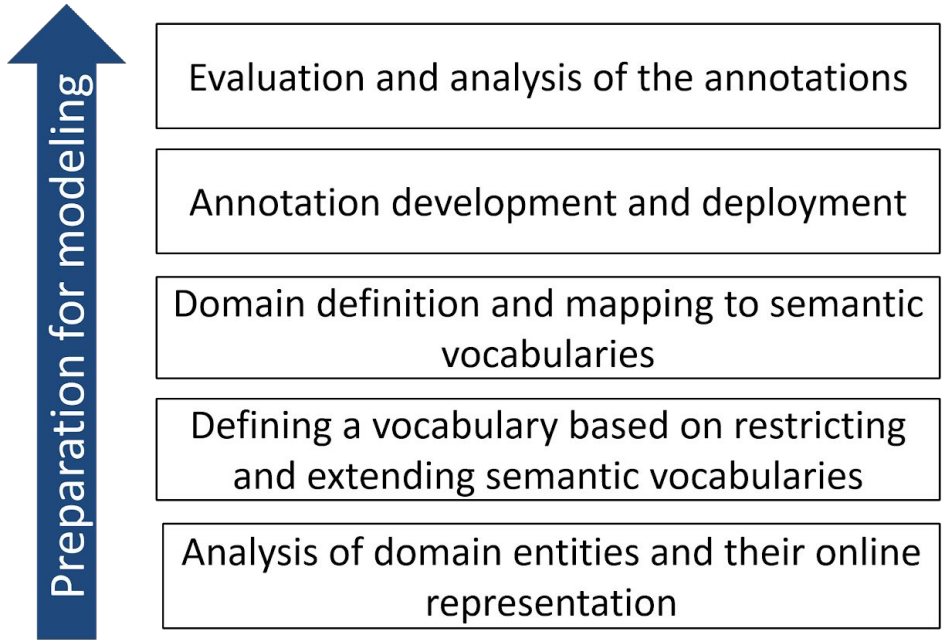
```
{
  "@context": "http://schema.org",
  "@type": "LocalBusiness",
  "name": "Imbiss-Stand \"Wurscht & Durscht\"",
  "geo": {
    "@type": "GeoCoordinates",
    "latitude": "47.3006092921797",
    "longitude": "10.9136698539673"
  },
  "address": {
    "@type": "PostalAddress",
    "streetAddress": "Unterer Mooswaldweg 2",
    "addressLocality": "Obsteig",
    "postalCode": "6416",
    "addressCountry": "AT",
    "telephone": "+43 664 / 26 32 319",
    "faxNumber": "",
    "email": "info@wudu-imbiss.at",
    "url": "www.wudu-imbiss.at"
  },
  "description": "Der Imbisstand direkt an der Bundesstraße B 189 in Obsteig verwöhnt die Gäste mit qualitativ hochwertigen \"Würschtln\" (Wurst) aller Art.",
}
```

2. Knowledge Creation - Methodology

a.k.a Knowledge Acquisition: "...describes the process of extracting information from different sources, structuring it, and managing established knowledge" - Schreiber et al.



2. Knowledge Creation - Methodology



1) bottom-up: describes a first annotation process

a) analysis of a domain's entities and their (online) representation
b) defining a vocabulary (potentially by restricting and/or extending an already existing voc.)
c) "domain definition", mapping to semantic vocabularies
d) annotation
e) evaluation and analysis of annotations

2. Knowledge Creation - Methodology

Domain Specification Modeling

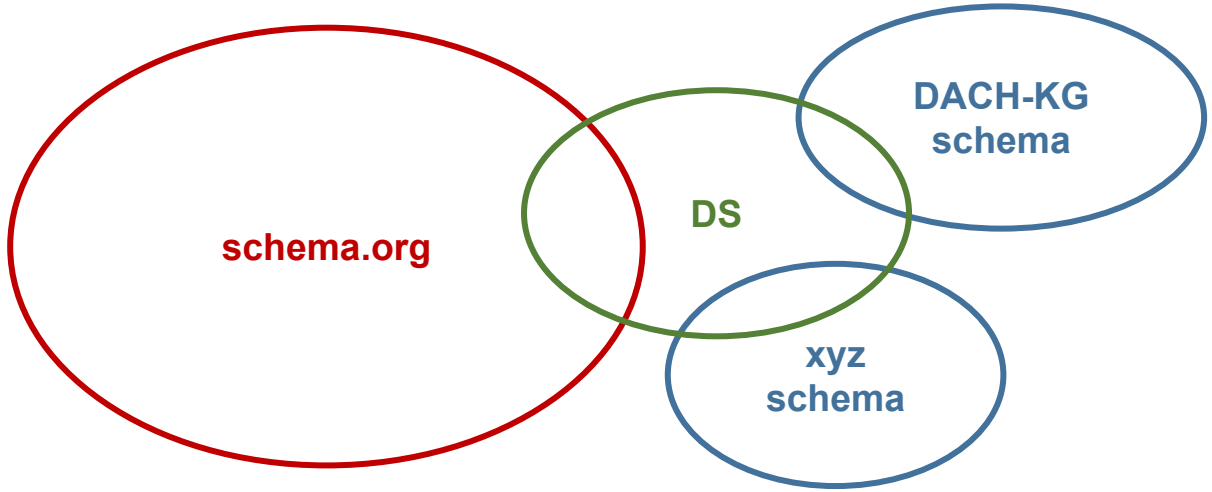
2) domain specification modeling: reflects the results of step 1)

formalize the findings of step 1) in a

- unified
- exchangeable
- machine-read and understandable way

⇒ **Domain Specifications**

2. Knowledge Creation - DS



What are Domain Specifications (DS)?

Conceptually:

“Templates for important schema.org terms”

“Extended subset schema.org”

“The union of a subset of schema.org and a subset of a schema.org extension”

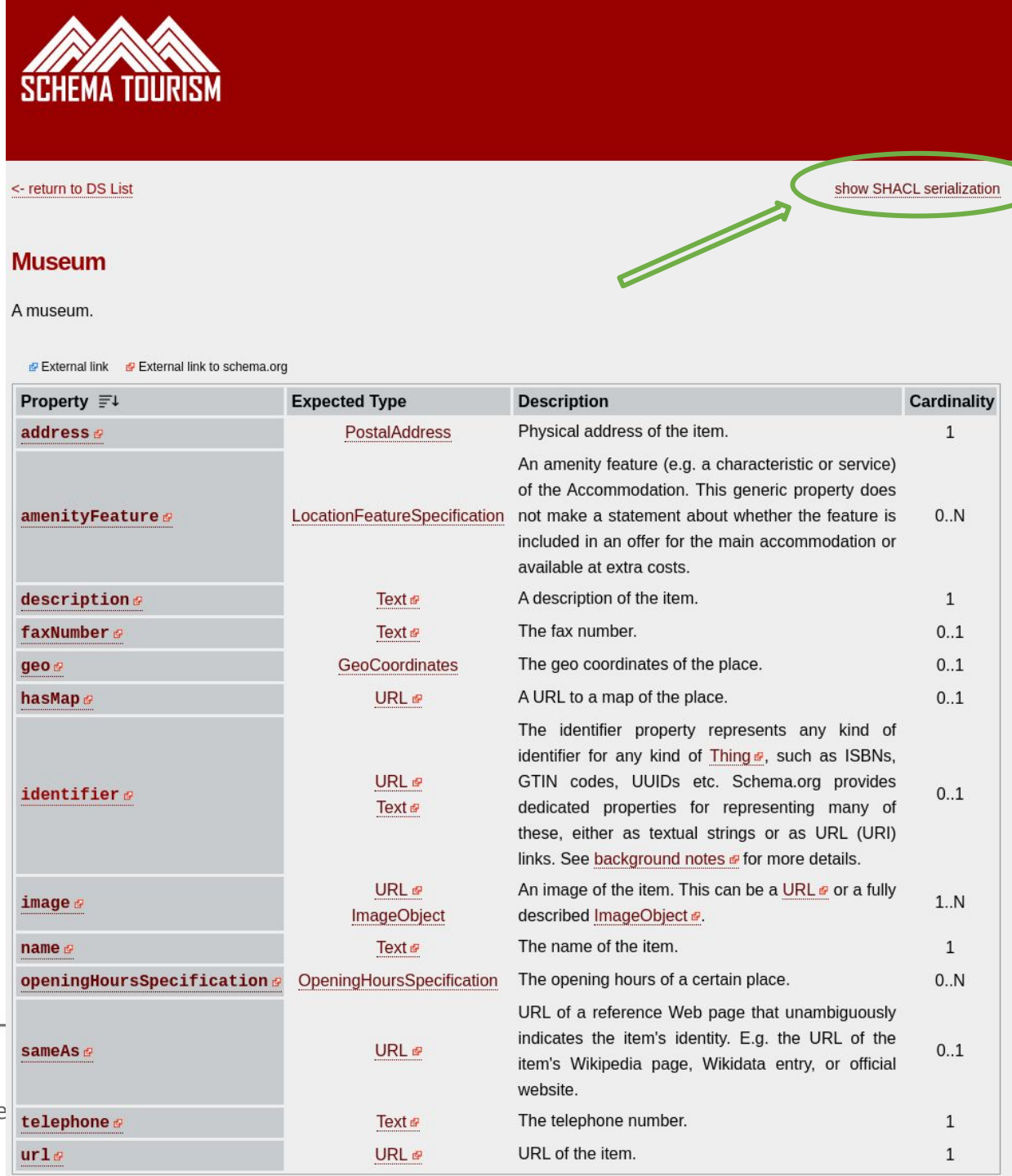
2. Knowledge Creation - DS

Example: Museum

DS defines:

- Main class: e.g.: schema.org/Museum
- important properties
  - address
  - amenityFeature
  - description
- the properties' ranges
  - address ⟶ schema.org/PostalAddress
  - description ⟶ Text
- cardinality

  0, 1, 0..1, 0..N, 1..N



Property	Expected Type	Description	Cardinality
address	PostalAddress	Physical address of the item.	1
amenityFeature	LocationFeatureSpecification	An amenity feature (e.g. a characteristic or service) of the Accommodation. This generic property does not make a statement about whether the feature is included in an offer for the main accommodation or available at extra costs.	0..N
description	Text	A description of the item.	1
faxNumber	Text	The fax number.	0..1
geo	GeoCoordinates	The geo coordinates of the place.	0..1
hasMap	URL	A URL to a map of the place.	0..1
identifier	URL Text	The identifier property represents any kind of identifier for any kind of Thing, such as ISBNs, GTIN codes, UUIDs etc. Schema.org provides dedicated properties for representing many of these, either as textual strings or as URL (URI) links. See background notes for more details.	0..1
image	URL ImageObject	An image of the item. This can be a URL or a fully described ImageObject.	1..N
name	Text	The name of the item.	1
openingHoursSpecification	OpeningHoursSpecification	The opening hours of a certain place.	0..N
sameAs	URL	URL of a reference Web page that unambiguously indicates the item's identity. E.g. the URL of the item's Wikipedia page, Wikidata entry, or official website.	0..1
telephone	Text	The telephone number.	1
url	URL	URL of the item.	1

2. Knowledge Creation - DS

What are Domain Specifications (DS)?

Technically:

- JSON files
- SHACL syntax
- “Shapes” drawn around the schema.org-vocabulary tree
- every DS corresponds to a SHACL file
- SHACL is a W3C Standard

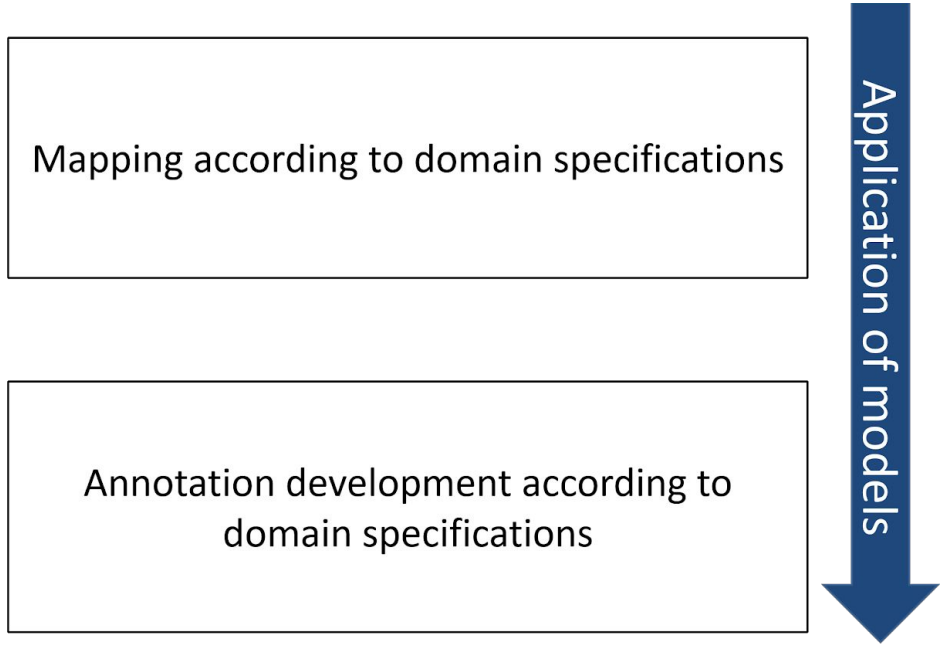
```
95      {
96        "@type": "sh:PropertyShape",
97        "sh:maxCount": 1,
98        "sh:minCount": 1,
99        "sh:order": 2,
100       "sh:path": "schema:address",
101       "sh:class": "schema:PostalAddress",
102       "sh:node": {
103         "@type": "sh:NodeShape",
104         "sh:property": [
105           {
106             "@type": "sh:PropertyShape",
107             "sh:maxCount": 1,
108             "sh:minCount": 1,
109             "sh:order": 0,
110             "sh:path": "schema:addressCountry",
111             "sh:class": "schema:Country",
112             "sh:node": {
113               "@type": "sh:NodeShape",
114               "sh:property": [
115                 {
116                   "@type": "sh:PropertyShape",
117                   "sh:maxCount": 1,
118                   "sh:minCount": 1,
119                   "sh:order": 0,
120                   "sh:path": "schema:name",
121                   "sh:datatype": "xsd:string"
122                 }
123               ]
124             }
125           },
126           {
127             "@type": "sh:PropertyShape",
128             "sh:maxCount": 1,
129             "sh:minCount": 1,
130             "sh:order": 1,
```

2. Knowledge Creation - Methodology

3) top-down: applies models for further knowledge acquisition

a) mapping according to domain specifications

b) annotation development according to domain specifications

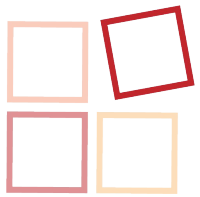


2. Knowledge Creation - tools - semantify.it

In the “early days” of our KG building efforts: three core questions (by our show-case users*) arose

* our efforts were always driven by educating people (real users, outside of academia, mostly from the industry/tourism) to create their own semantically rich content

1) which vocabulary to use
2) how to create JSON-LD files
3) how to publish those annotations (schema.org in JSON-LD files)



semantify.it
bring meaning to the web

Tool, developed as a research project, grown to a full-stack annotation creation, validation and publication framework!

2. Knowledge Creation - tools

1) Which vocabulary to choose? ⇒ schema.org

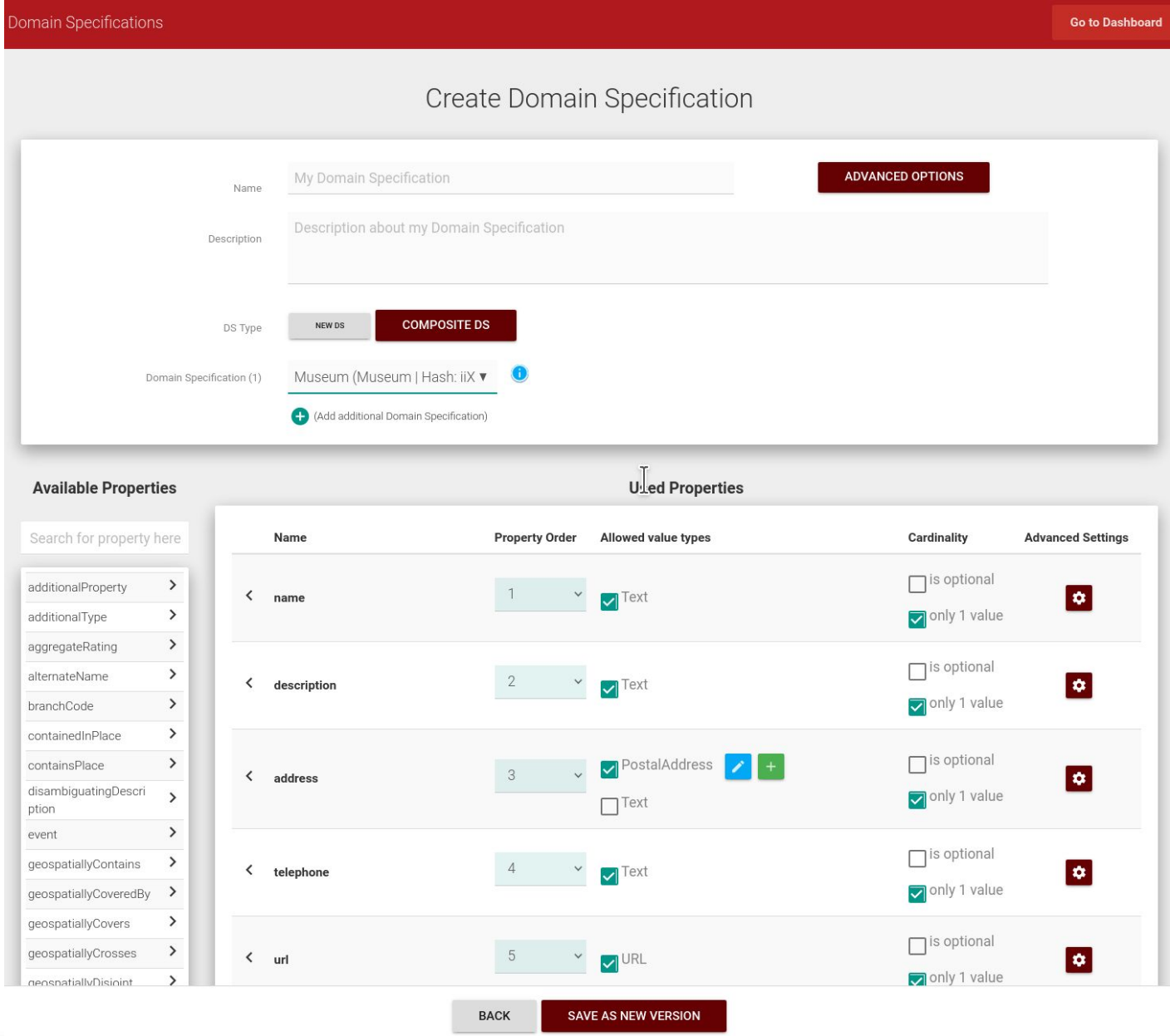
Still hundreds of classes and properties in schema.org

Domain Specifications

- Domain expert builds DS files as templates for editor
- Easy to use DS editor

Domain Specifications

1. from scratch
2. replicate (and change) exiting DS
3. combine existing DS (and extend)



2. Knowledge Creation - DS - Demo

Domain	Property	Range
s:LandmarksOrHistoricalBuildings	s:address	s:PostalAddress
	s:containedInPlace	s:Place
s:PostalAddress	s:streetAddress	s:Text
	s:addressLocality	s:Text
	s:addressCountry	s:Country
	s:postalCode	s:Text
s:TouristAttraction	s:availableLanguage	s:Text

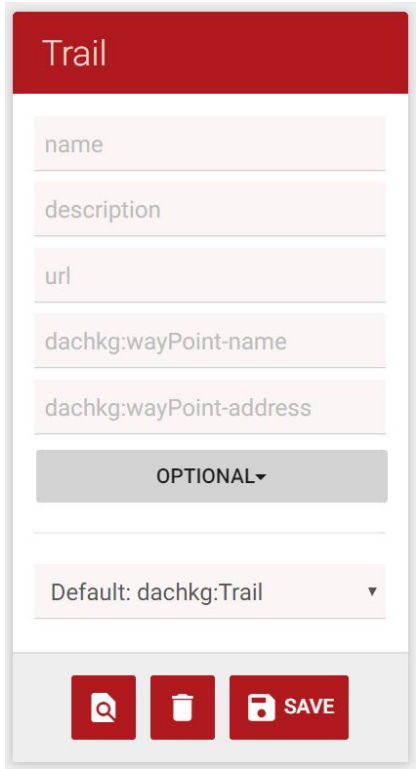
Demo: sight-seeing DS
https://semantify.it/domainSpecifications

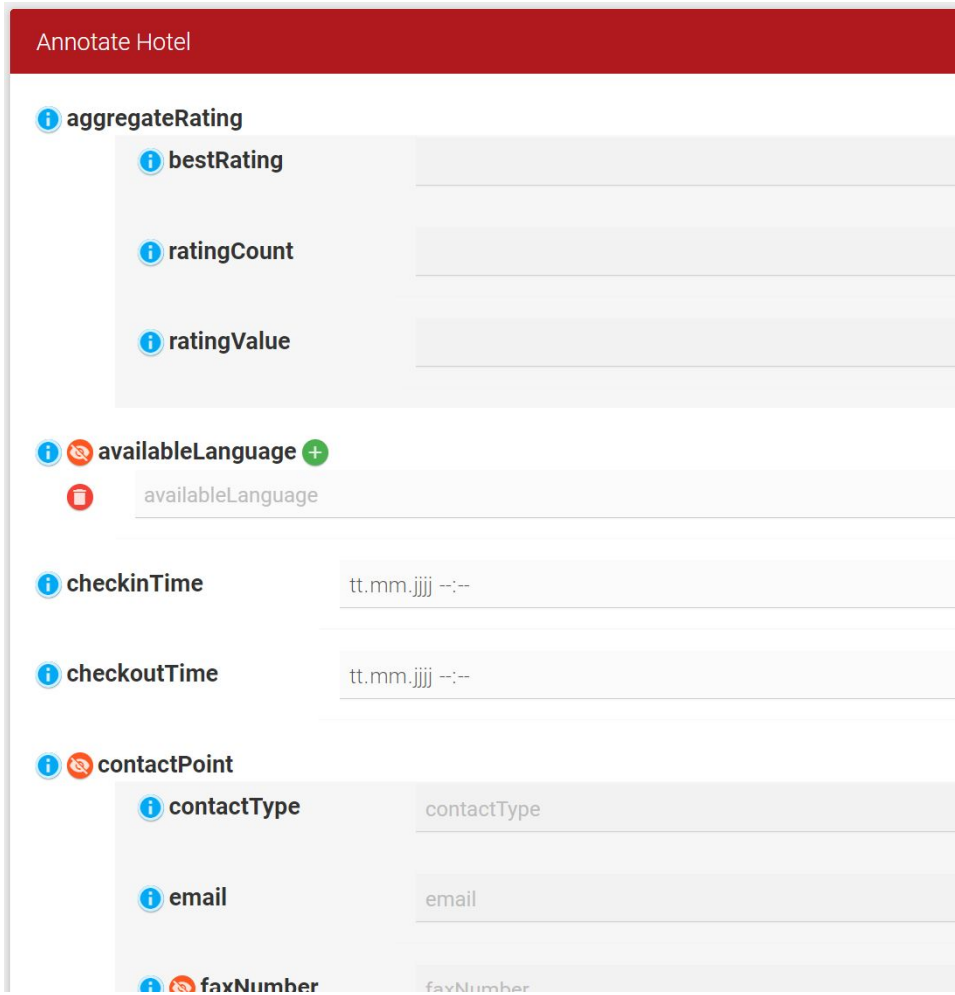
2. Knowledge Creation - tools - semantify.it

2) How to create those JSON-LD files?

- semantify.it editor & instant annotations
  - based on DS
  - Inside platform (big DS files)
  - or Instant Annotations (IA)
    portable to every website (based on JS)
- wrapper framework
- semi-automatic
- mappers (RocketRML)

RocketRML ⇒

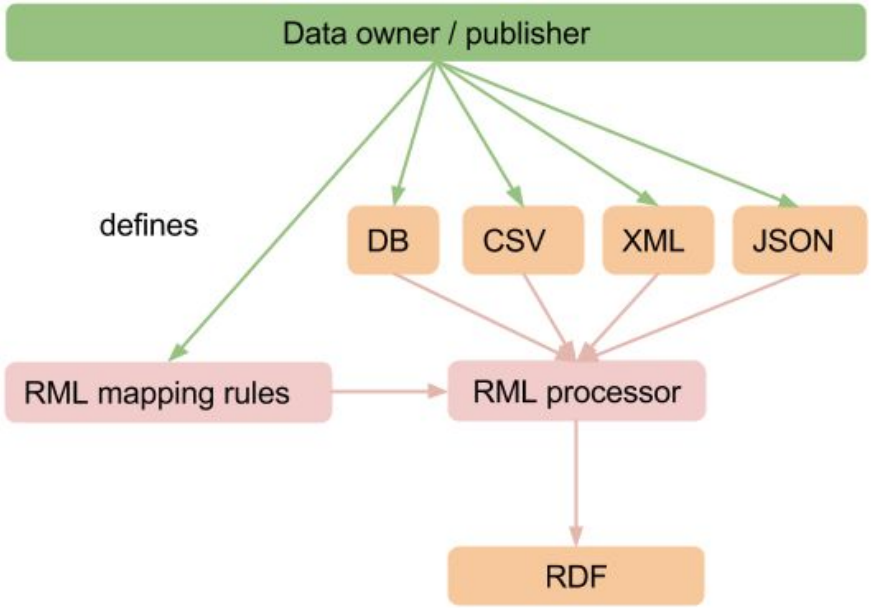




2. RocketRML - A Scalable RML Mapper [Simsek et al., 2019]

Based on RML [Dimou et al., 2014]:

- Easier to learn RML than a programming language
- Easy sharing
- Mapping can be visualized
- Mapfiles can be faster to write than code
- Easily change mappings

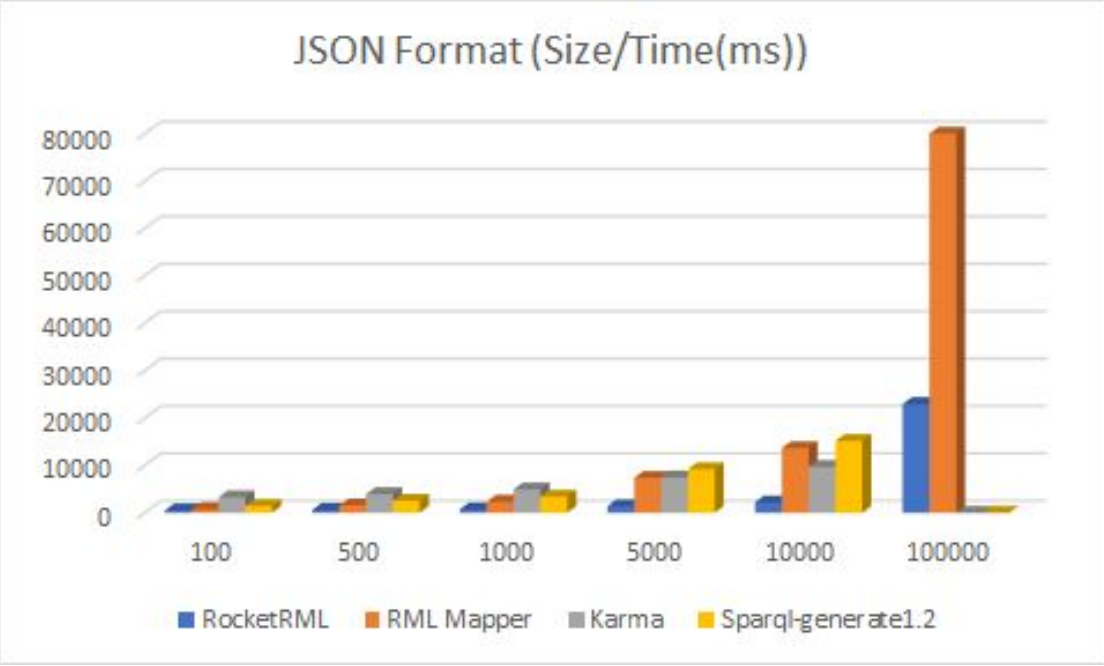


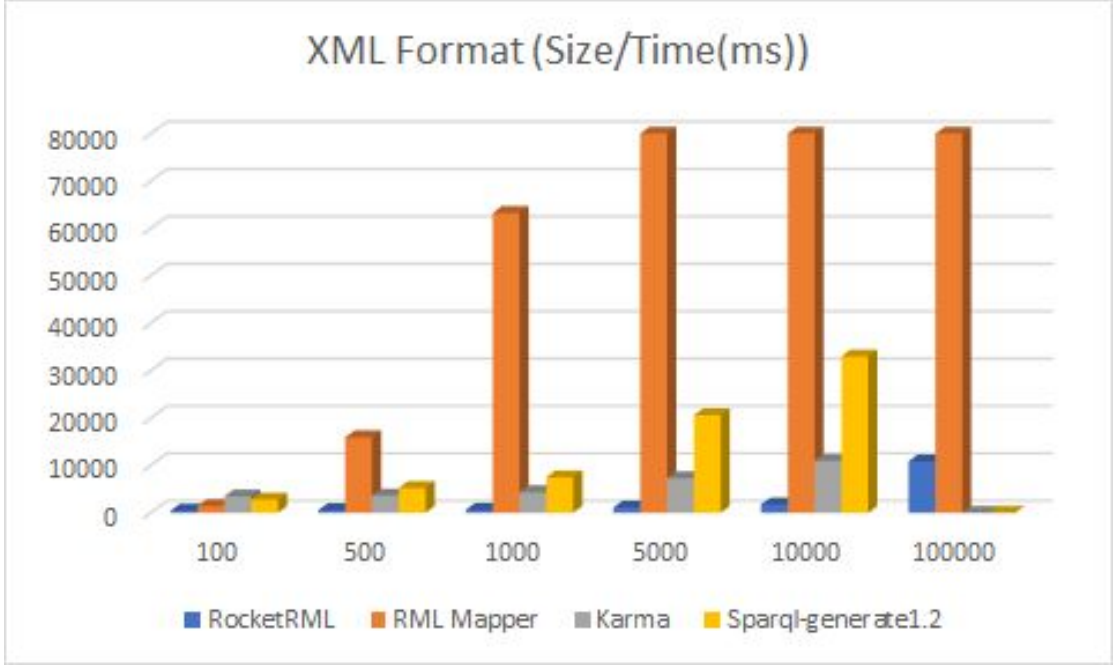
RML

YARRRML

Matey

2. RocketRML - Performance





2. RocketRML - Source Code

GitHub https://github.com/semantifyit/RocketRML

npm https://www.npmjs.com/package/rocketrml

docker https://semantifyit.github.io/RocketRML/

2. RocketRML - A Quite Scalable RML Mapper

- Quick demo (https://semantifyit.github.io/rml):

Raw data set (JSON):

```json
{
  "persons": [
      {
          "firstname": "Elias",
          "lastname": "Kärle",
          "speaks": [
              "de",
              "en",
              "it",
              "fr",
              "Tyrolean"
          ]
      },
      {
          "firstname": "Umutcan",
          "lastname": "Simsek",
          "speaks": [
              "tr",
              "en",
              "de",
              "Hessisch"
          ]
      }
  ]
}
```

Mapping file (YARRRML*):

```yaml
prefixes:
  schema: "http://schema.org/"
  myfunc: "http://myfunc.com/"
mappings:
  person:
    sources:
      - ['input~jsonpath', '$.persons[*]']
    s: http://example.com/$(firstname)
    po:
      - [a, schema:Person]
      - [schema:name, $(firstname)]
      - [schema:language, $(speaks.*)]
```

* YARRRML is the yaml-based, human readable, translation of the actual turtle-based RML syntax. (https://semantifyit.github.io/rml/)

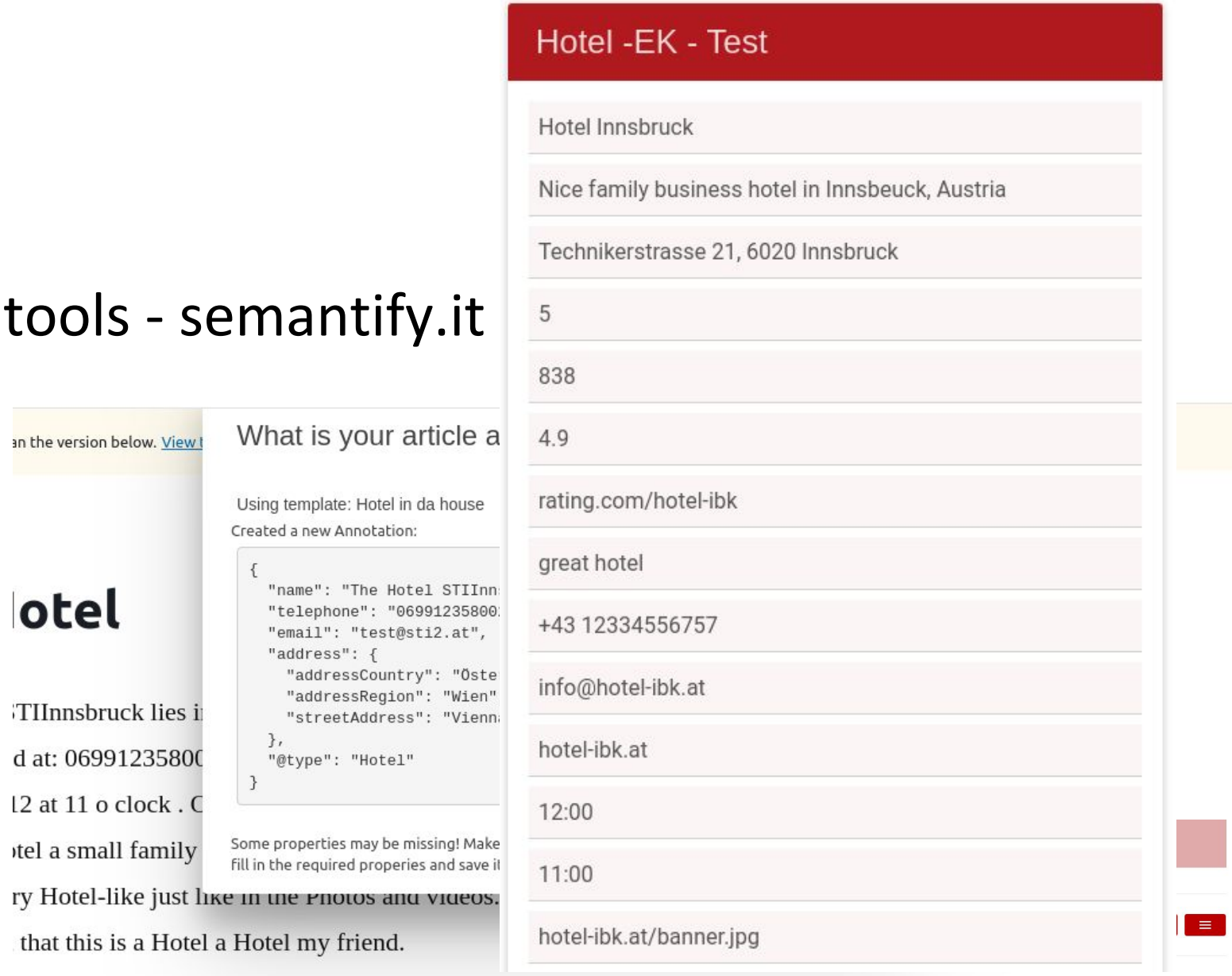
Mapping resu

```json
[
  {
    "@id": "http://example.com/Elias",
    "@type": "Person",
    "language": [
      "de",
      "en",
      "it",
      "fr",
      "Tyrolean"
    ],
    "name": "Elias",
    "@context": {
      "@vocab": "http://schema.org/"
    }
  },
  {
    "@id": "http://example.com/Umutcan",
    "@type": "Person",
    "language": [
      "tr",
      "en",
      "de",
      "Hessisch"
    ],
    "name": "Umutcan",
    "@context": {
      "@vocab": "http://schema.org/"
    }
  }
]
```

2. Knowledge Creation - tools - semantify.it

2) How to create those JSON-LD files?

- semi automatic generation
  - WordPress plugin
  - “guess” the entities of the web page through machine learning
  - model trained on entities in our knowledge graph



2. Knowledge Creation - tools - semantify.it

3) How to publish annotations (schema.org in JSON-LD files)?

- copy-paste?
  ⟶ pasting content to website is no option for inexperienced users and does not scale

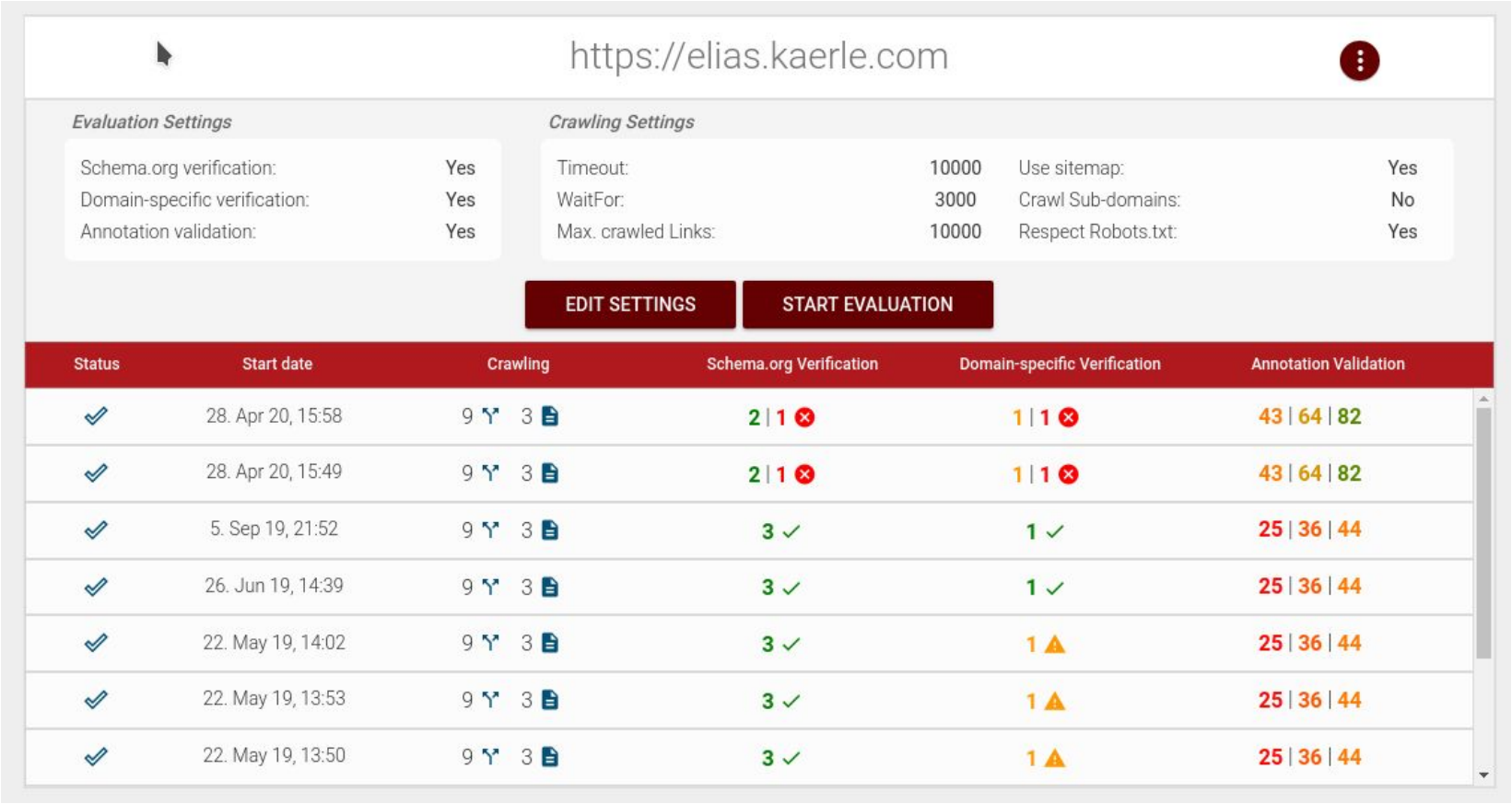
- semantify.it **stores** all created annotations and **provides** them over an **API**

2. Knowledge Creation

Evaluator

validation & verification

- **verification** against schema.org
- **verification** against DS
- **validation** against website →



https://elias.kaerle.com

Statistics

RETRIEVAL | SDO VERIFICATION | DS VERIFICATION | VALIDATION

Crawling statistics

Duration

8 seconds

Start Date:	28. Apr 20, 15:58
End Date:	28. Apr 20, 15:58
Robots.txt found:	Yes
Sitemap Found:	Yes

Links found

9

Retrieval success:	9
Retrieval errors:	0
Extraction errors:	0

Annotation statistics

Annotations found

3

RDFa:	0
Microdata:	0
JSON-LD:	3
Invalid JSON-LD:	0

Person: 1 | Product: 1 | ScholarlyArticle: 1

Web page results

Order by URL Path Ascending

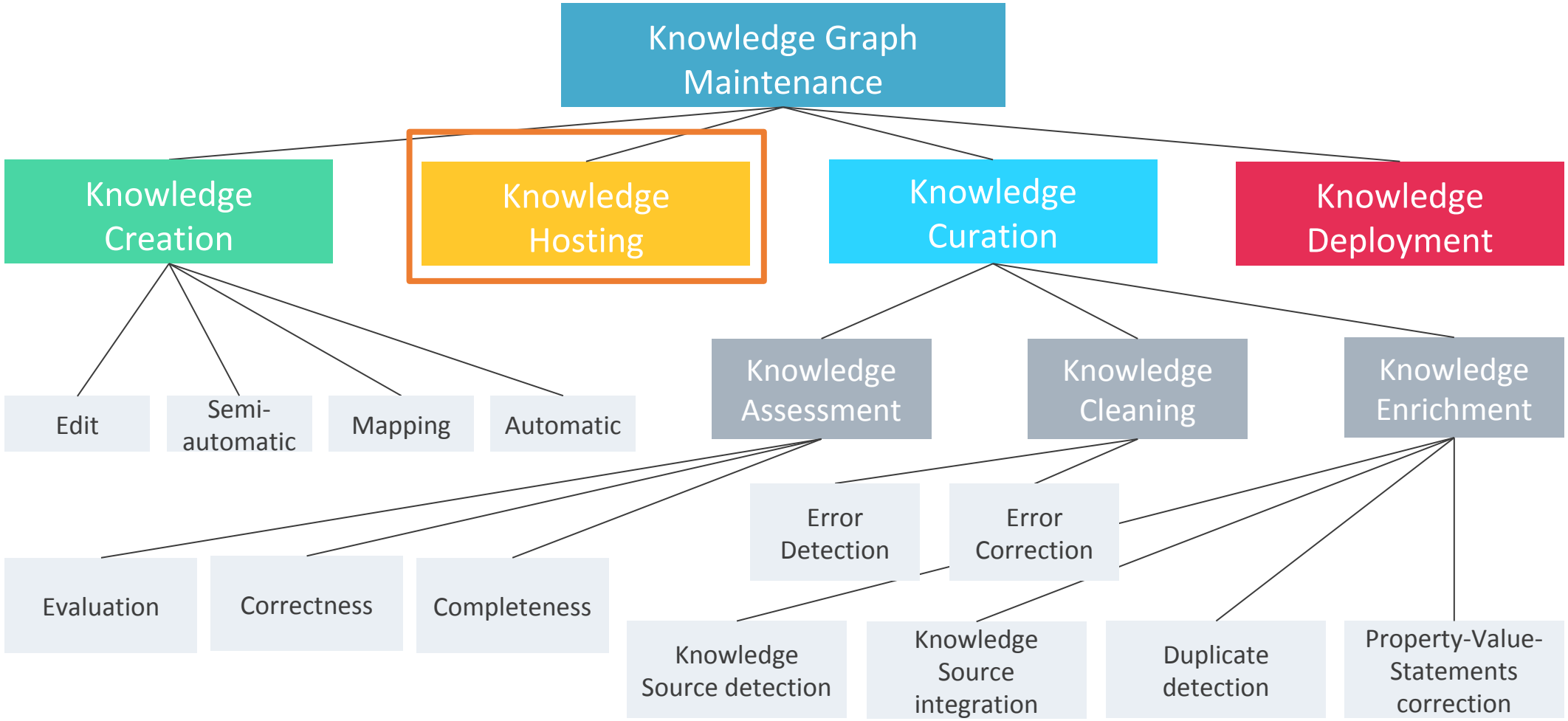
URL Path	Crawling	Schema.org Verification	Domain-specific Verification	Annotation Validation
/	1	1 ✓	1 ⚠	66
/annotating-a-hotel-offering-rooms-with-the-new-schema-org-version-3-1/	0	0	0	-
/annotating-ski-resorts-lifts-and-slopes-with-schema-org/	0	0	0	-
/becoming-an-entity-in-the-google-knowledge-graph/	0	0	0	-
/bibtex-style-for-rinton-press-journal-of-mobile-multimedia-jmm/	0	0	0	-
/blog/	0	0	0	-
/projects/	1	1 ✓	0	82
/publications/	1	1 ✗	1 ✗	43
/sitemap.html	0	0	0	-

2. Knowledge Creation - tools - semantify.it

Annotation Validation Result

Property	Value	Score	Confidence
author.name	Elias Kaerle	100	high
datePublished	2017-01-01	0	medium
headline	semantify. it, a Platform for Creation, Publication and Distribution of Semantic Annotations.	100	high
image	https://semantify.it/images/logo_text.png	0	medium
publisher.logo.url	http://www.thinkmind.org/images/top_left.gif	0	medium
publisher.name	http://www.thinkmind.org/index.php?view=article&articleid=semapro_2017_2_10_30007	100	high
dateModified	2020-04-28T15:56	0	medium

3. KNOWLEDGE HOSTING

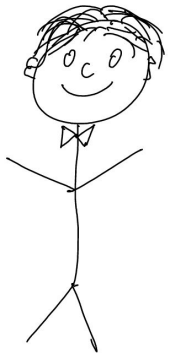

Knowledge Graph Maintenance
Knowledge Creation
Knowledge Hosting
Knowledge Curation
Knowledge Deployment
Edit
Semi-automatic
Mapping
Automatic
Knowledge Assessment
Knowledge Cleaning
Knowledge Enrichment
Evaluation
Correctness
Completeness
Error Detection
Error Correction
Knowledge Source detection
Knowledge Source integration
Duplicate detection
Property-Value-Statements correction

3. Knowledge Hosting

In our context:

"Knowledge is represented in the form of **semantically enriched** data"

→ **metadata** is added to **describe** the data
→ by using a (de-facto) **standard vocabulary** (schema.org in our case)
→ according to the principles of **RDF**
→ also called **annotated** data



Max	schema:name = "Max"
30 years	schema:birthDate = "1990"
from Innsbruck	schema:homeLocation = "Innsbruck"
researcher	schema:hasOccupation = "researcher"

3. Knowledge Hosting

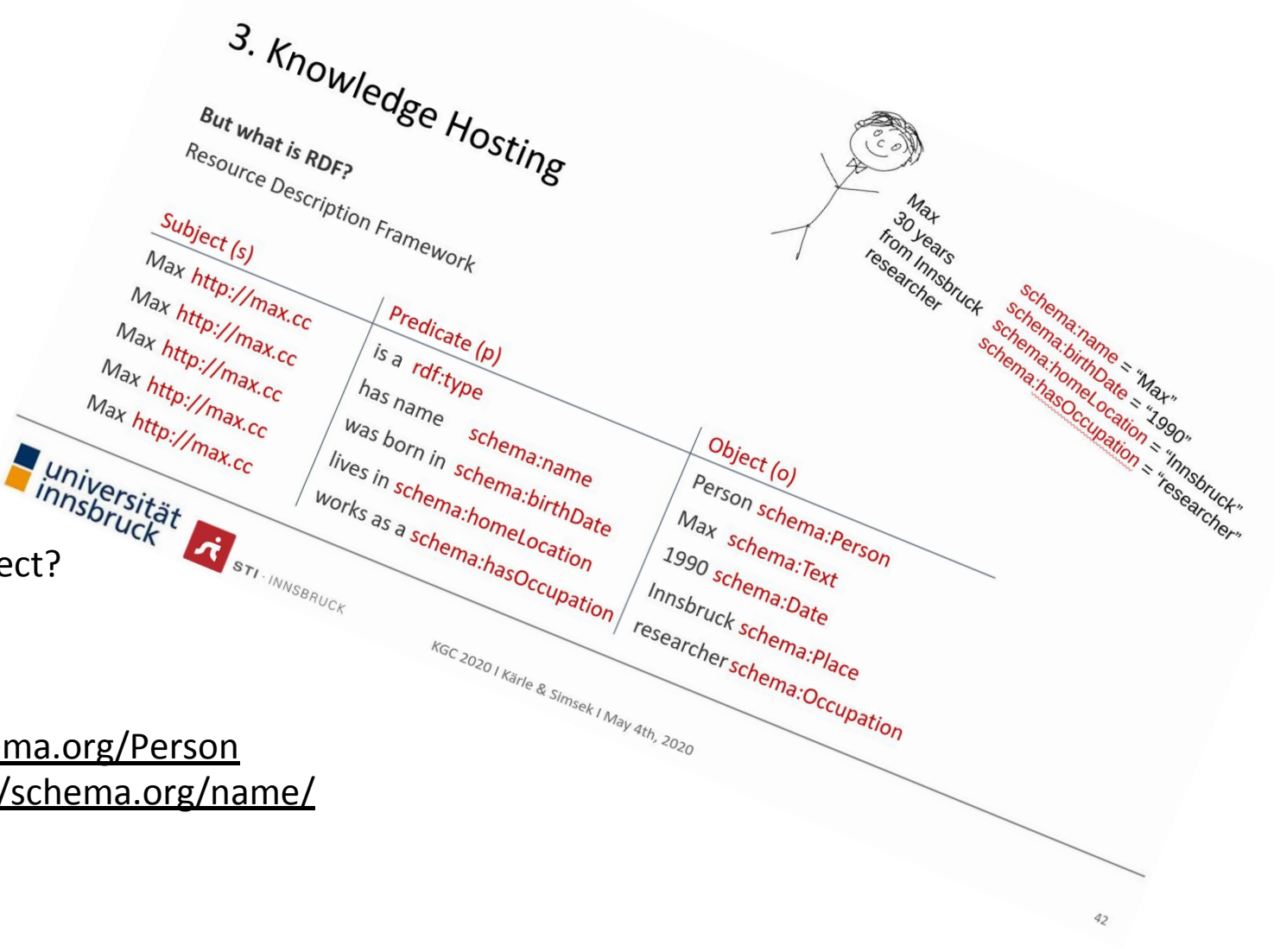
schema:name = “Max”
schema:birthDate = “1990”
schema:homeLocation = “Innsbruck”
schema:hasOccupation = “researcher”

But what is RDF?

Resource Description Framework

Subject (s)	Predicate (p)	Object (o)
Max http://max.cc	is a rdf:type	Person schema:Person
Max http://max.cc	has name schema:name	Max schema:Text
Max http://max.cc	was born in schema:birthDate	1990 schema:Date
Max http://max.cc	lives in schema:homeLocation	Innsbruck schema:Place
Max http://max.cc	works as a schema:hasOccupation	researcher schema:Occupation

3. Knowledge Hosting



But what is RDF?

what actually are the Subject, Predicate and Object?

Either a **URL**

- to identify **resources** http://max.cc
- to refer to **types** of an ontology http://schema.org/Person
- to refer to **properties** of an ontology http://schema.org/name/

or a **literal**

- String: “Max”
- Date: “1990”
- Number: 42

3. Knowledge Hosting

» 2 ways of hosting (at least):

1. **JSON-LD (for websites)**

{“@context”:”http://schema.org”

“@type”:”Person”

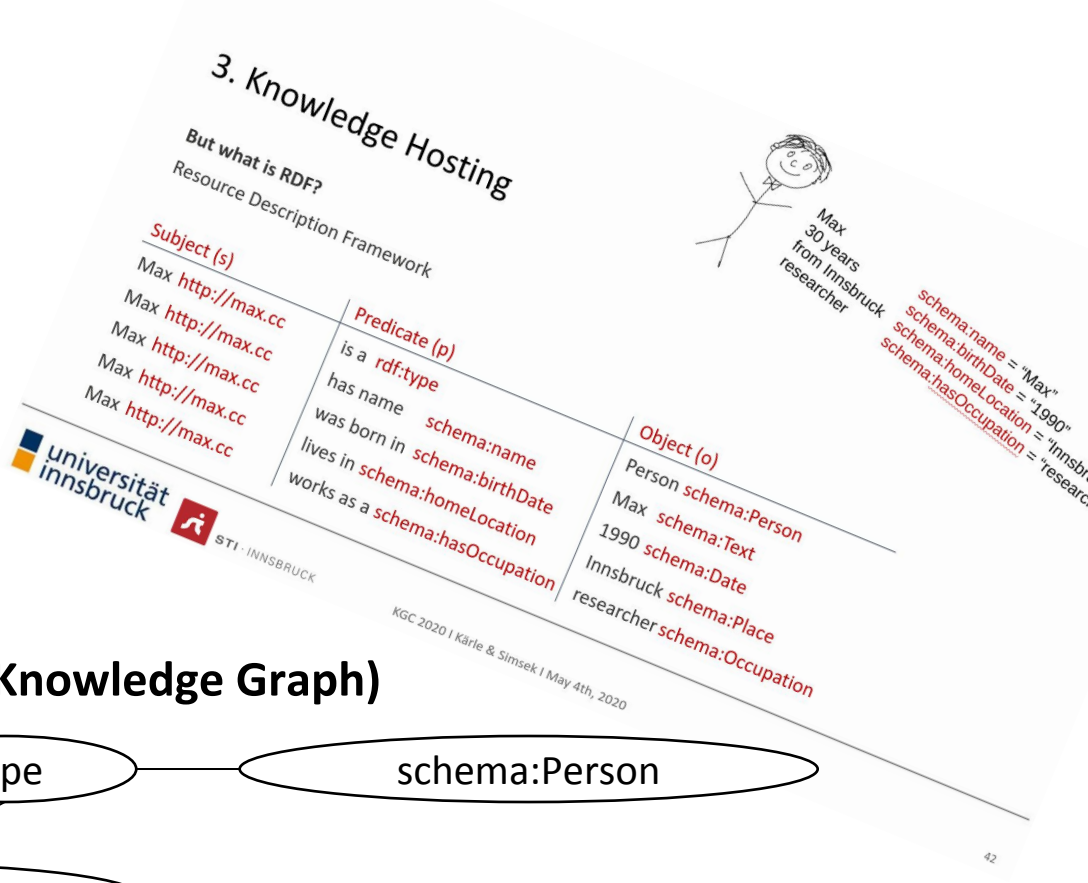
“@id”: “http://max.cc”,

“name”:”Max”,

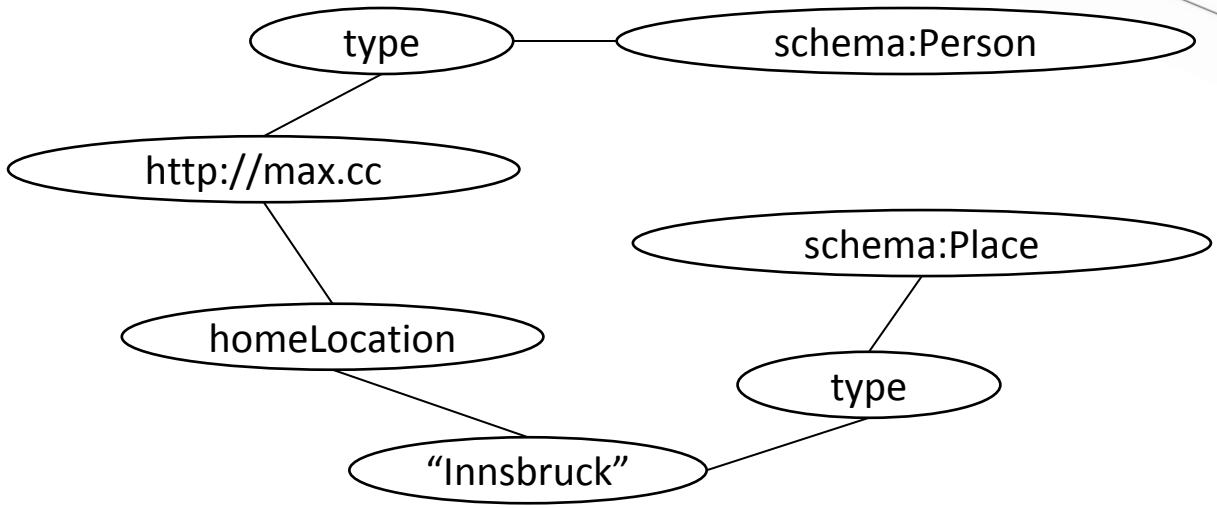
“homeLocation”:”Innsbruck”

“birthDate”:”1990”

“hasOccupation”:”researcher”}



2. Graph Database (Knowledge Graph)



3. Knowledge Hosting

Hosting as Knowledge Graph:

Use-case: storing semantically annotated data as a full-fledged Knowledge Graph

⟶ Linked Open Data repositories

⟶ enterprise Knowledge Graphs

⟶ advanced reasoning needs

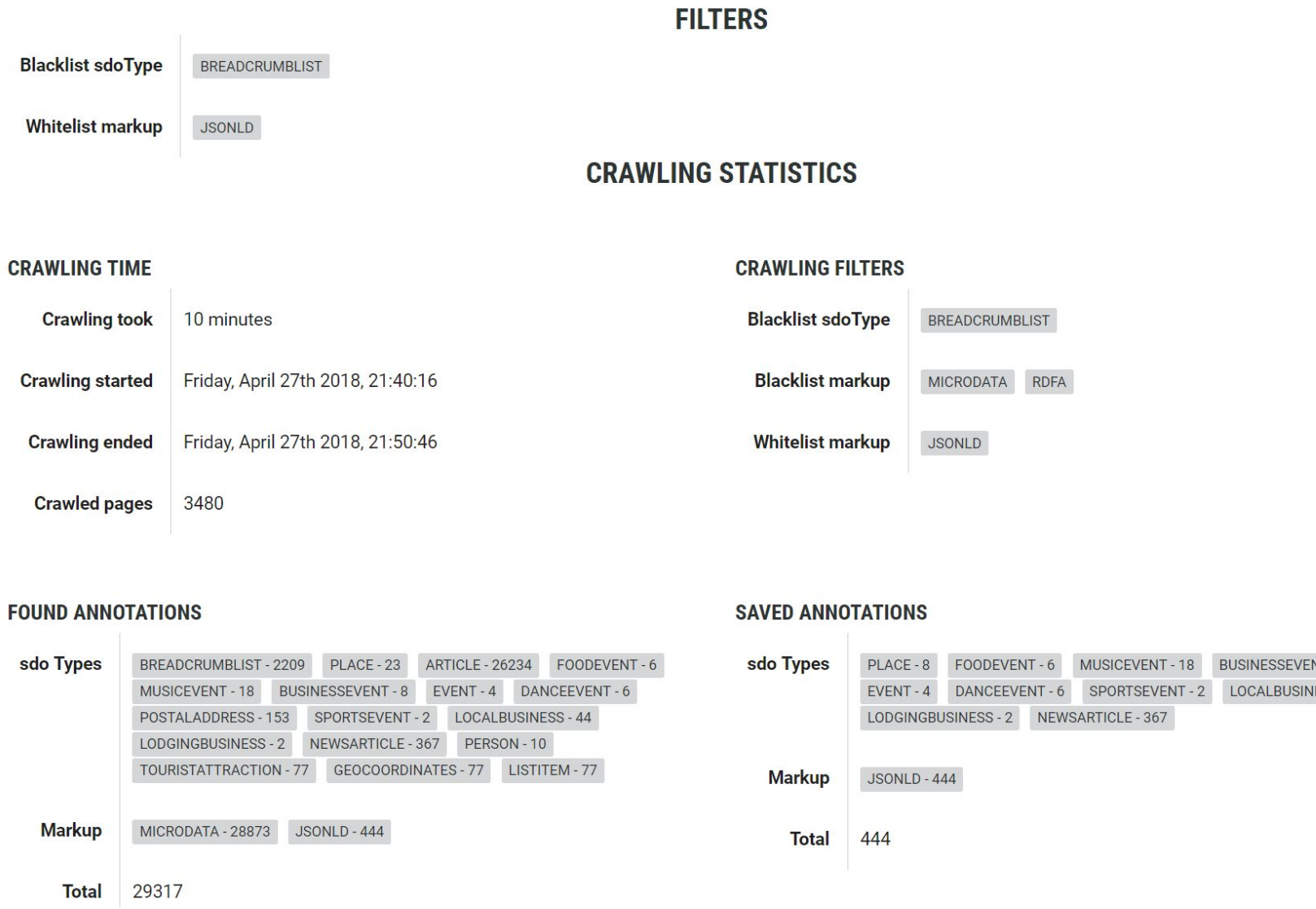
⟶ ML, intelligent assistants

Collection/creation: due to potentially millions of annotation files: mapping framework or also crawling of annotated web-sites ⟶ **semantify.it-broker**

3. Knowledge Hosting

semantify.it-broker:

- crawling platform to collect annotated data in JSON-LD, Microdata, RDFa
- storage in graph database
- provision of SPARQL UI



FILTERS

Blacklist sdoType	BREADCRUMBLIST
Whitelist markup	JSONLD

CRAWLING STATISTICS

CRAWLING TIME

Crawling took	10 minutes
Crawling started	Friday, April 27th 2018, 21:40:16
Crawling ended	Friday, April 27th 2018, 21:50:46
Crawled pages	3480

CRAWLING FILTERS

Blacklist sdoType	BREADCRUMBLIST
Blacklist markup	MICRODATA RDFA
Whitelist markup	JSONLD

FOUND ANNOTATIONS

sdo Types	BREADCRUMBLIST - 2209, PLACE - 23, ARTICLE - 26234, FOODEVENT - 6, MUSICEVENT - 18, BUSINESSEVENT - 8, EVENT - 4, DANCEEVENT - 6, POSTALADDRESS - 153, SPORTSEVENT - 2, LOCALBUSINESS - 44, LODGINGBUSINESS - 2, NEWSARTICLE - 367, PERSON - 10, TOURISTATTRACTION - 77, GEOCOORDINATES - 77, LISTITEM - 77
Markup	MICRODATA - 28873, JSONLD - 444
Total	29317

SAVED ANNOTATIONS

sdo Types	PLACE - 8, FOODEVENT - 6, MUSICEVENT - 18, BUSINESSEVEN, EVENT - 4, DANCEEVENT - 6, SPORTSEVENT - 2, LOCALBUSIN, LODGINGBUSINESS - 2, NEWSARTICLE - 367
Markup	JSONLD - 444
Total	444

3. Knowledge Hosting

Hosting as Knowledge Graph:

Storage: due to RDF-nature, storage in graph database

with respect to:

- provenance
- historical data
- data duplication

In our current setting:

- historical data is kept im named graphs
- ~12 Billion statements

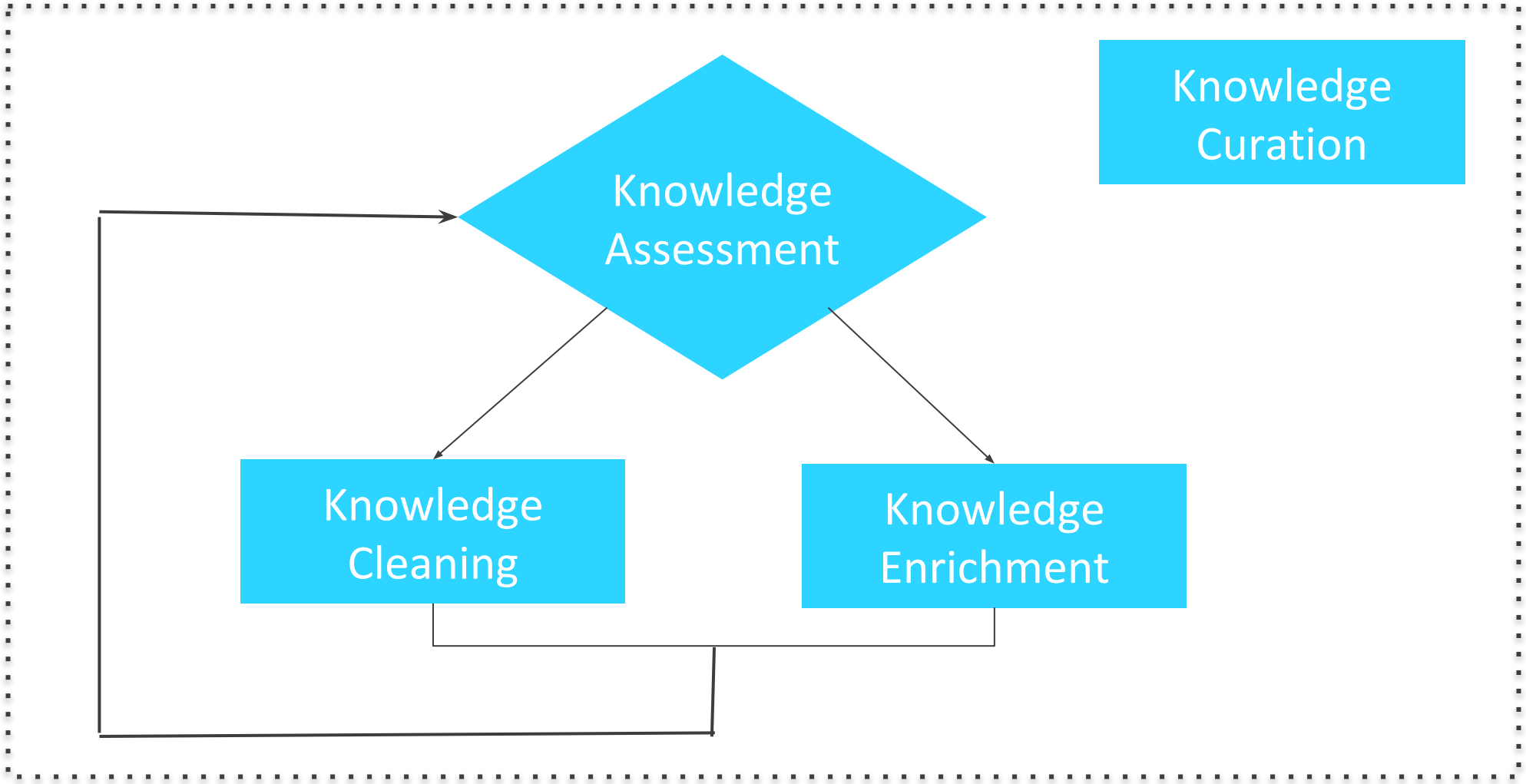
3. Knowledge Hosting

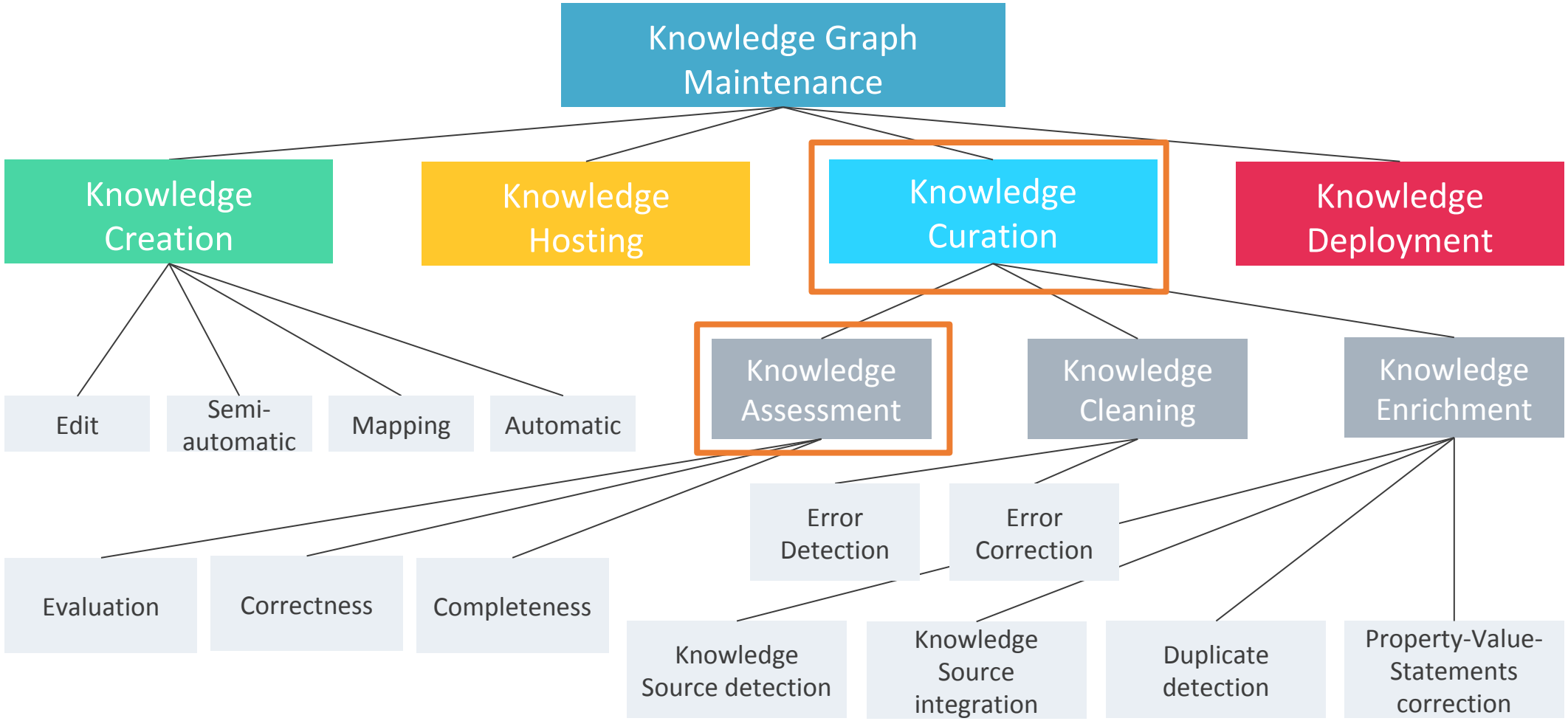
Hosting as Knowledge Graph:

Storage: popular triple stores
https://www.w3.org/wiki/LargeTripleStores

#	Name	# triples tested with
1	Oracle Spatial and Graph with Oracle Database 12c	1.08 T
2	AnzoGraph DB by Cambridge Semantics	1.065 T
3	AllegroGraph	1+ T
4	Stardog	50 B
5	OpenLink Virtuoso v7+	39.8 B
6	GraphDB™ by Ontotext	17 B

4. KNOWLEDGE CURATION


Knowledge Curation
Knowledge Assessment
Knowledge Cleaning
Knowledge Enrichment


Knowledge Graph Maintenance
Knowledge Creation
Knowledge Hosting
Knowledge Curation
Knowledge Deployment
Edit
Semi-automatic
Mapping
Automatic
Knowledge Assessment
Knowledge Cleaning
Knowledge Enrichment
Evaluation
Correctness
Completeness
Error Detection
Error Correction
Knowledge Source detection
Knowledge Source integration
Duplicate detection
Property-Value-Statements correction

4. Knowledge Assessment

- First step to improve the quality of a KG: Assess the situation
- Closely related to data quality literature
- Various dimensions for data quality assessment introduced [Batini & Scannapieco, 2006], [Färber et al., 2018], [Pipino et al., 2002], [Wang, 1998], [Wang & Strong, 1996], [Wang et al., 2001], [Zaveri et al., 2016])

4. Knowledge Assessment: Core Dimensions

1. accessibility
2. **accuracy (veracity)**
3. **completeness**
4. concise representation
5. consistent representation
6. cost-effectiveness
7. flexibility
8. interoperability
9. relevancy
10. timeliness (velocity)
12. trustworthiness
13. understandability
14. variety

an extended list can be found in [Fensel et al., 2020]

4. Knowledge Assessment: Metrics

Each dimension has a set of metrics. Each metric has a calculation function:

Example metric calculation from Understandability dimension:

$$m_{VariousLang}(r) = \begin{cases} 1 & labels\ provided\ in\ English\ and\ one\ other\ language \\ 0.5 & labels\ provided\ in\ only\ one\ language \\ 0 & otherwise \end{cases}$$

4. Knowledge Assessment: Metrics

Some dimensions are more contextual, i.e., needs external information alongside the Knowledge Graph

Example metric calculation from Relevancy dimension:

$$m_{DomainCoverage}(r) = \frac{Average\ DS\ Property\ Occurance\ on\ an\ Instance\ in\ r}{|Properties\ of\ DS|}$$

4. Knowledge Assessment: A Process Model

Decide on Dimension Weights

Each dimension may have have different level of importance for different domains or tasks.

Decide on Metric Weights

Each metric may have different impact on the calculation of the dimension to which they belong

Calculate the assessment score

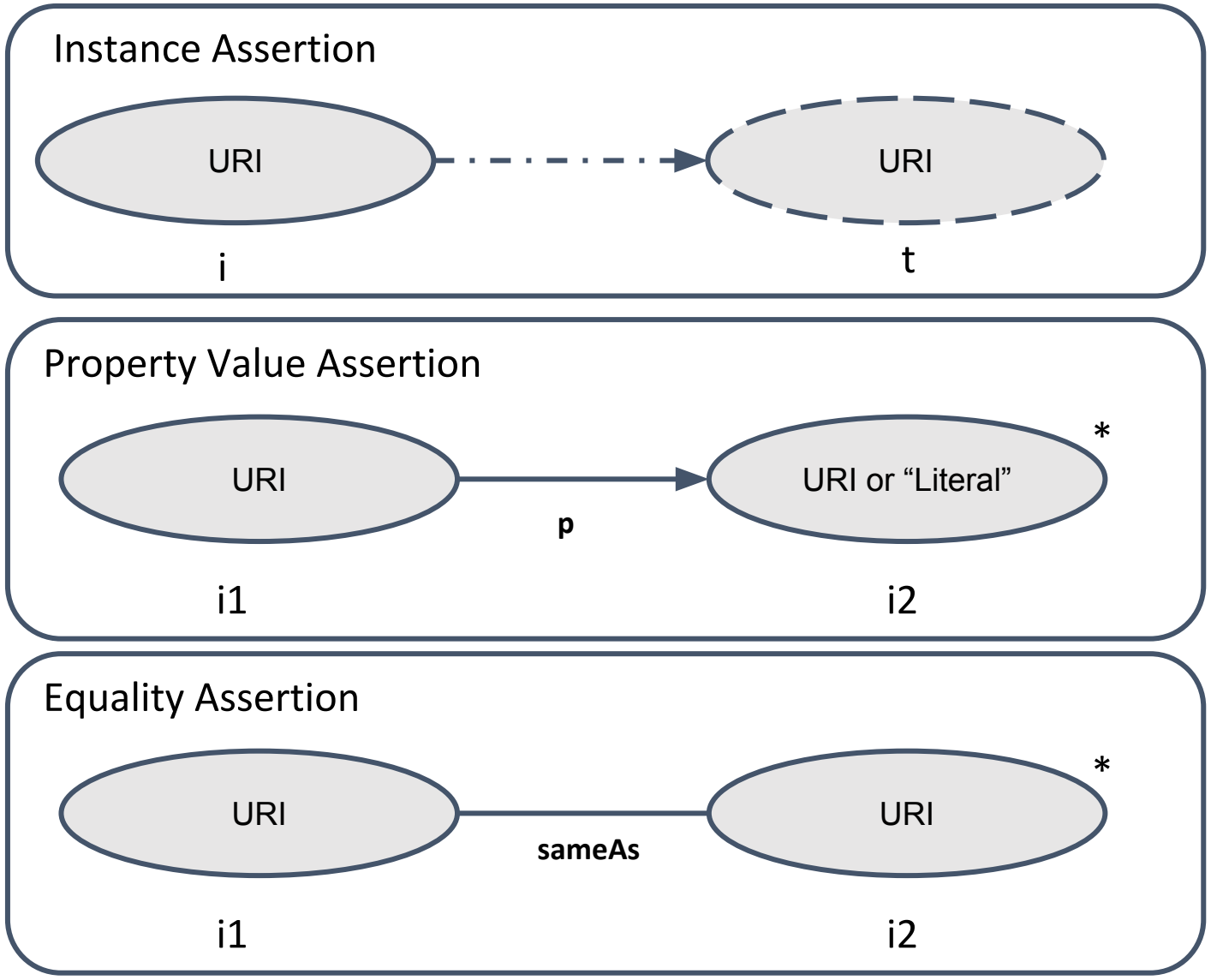
Calculate a weighted aggregate score for the Knowledge Graph for each domain or task.

Check out the workshop website for a list of tools!

A Running Example for Knowledge Cleaning and Enrichment

Domain	Property	Range
s:LandmarksOrHistoricalBuildings	s:address	s:PostalAddress
	s:containedInPlace	s:Place
s:PostalAddress	s:streetAddress	s:Text
	s:addressLocality	s:Text
	s:addressCountry	s:Country
	s:postalCode	s:Text
s:TouristAttraction	s:availableLanguage	s:Text

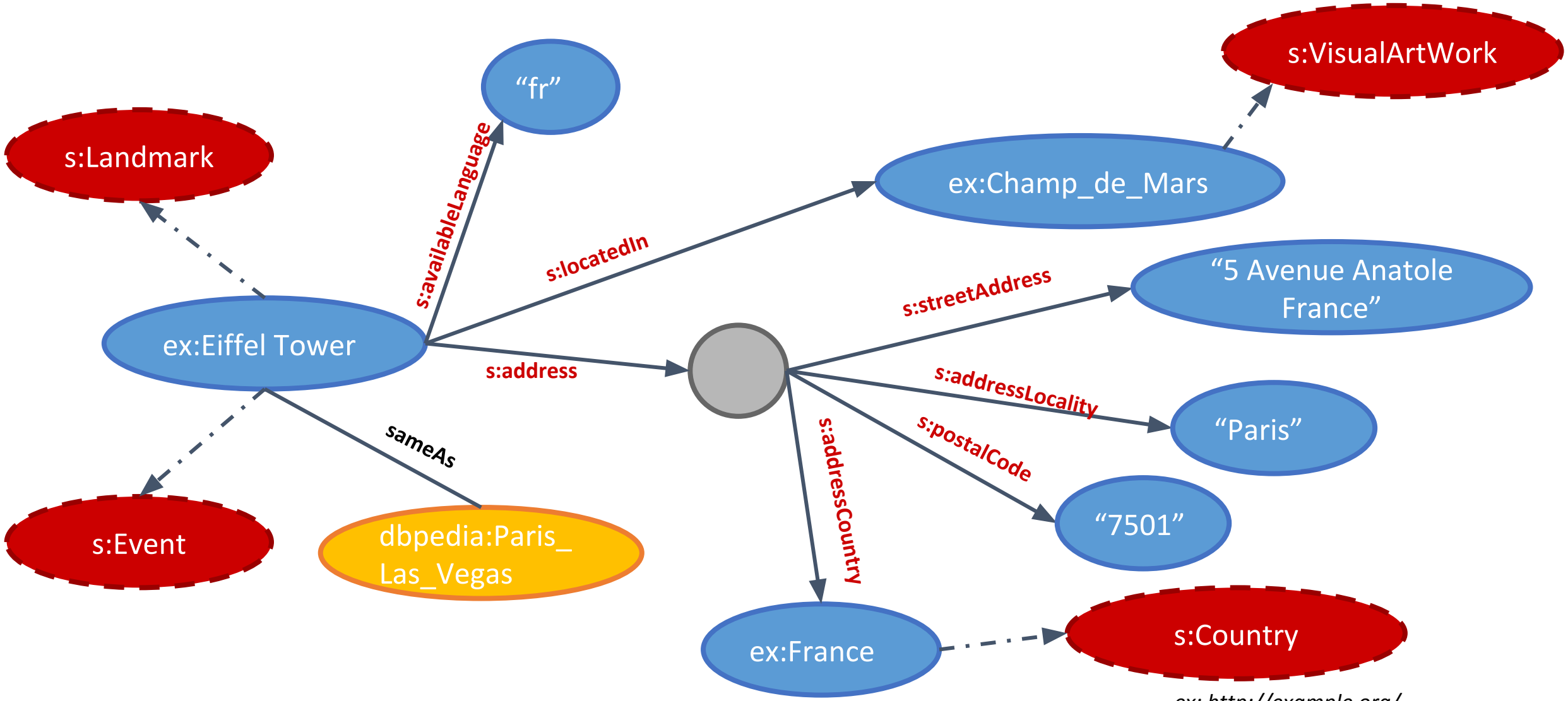
A subset of schema.org for the running example



i is an instance of the type t

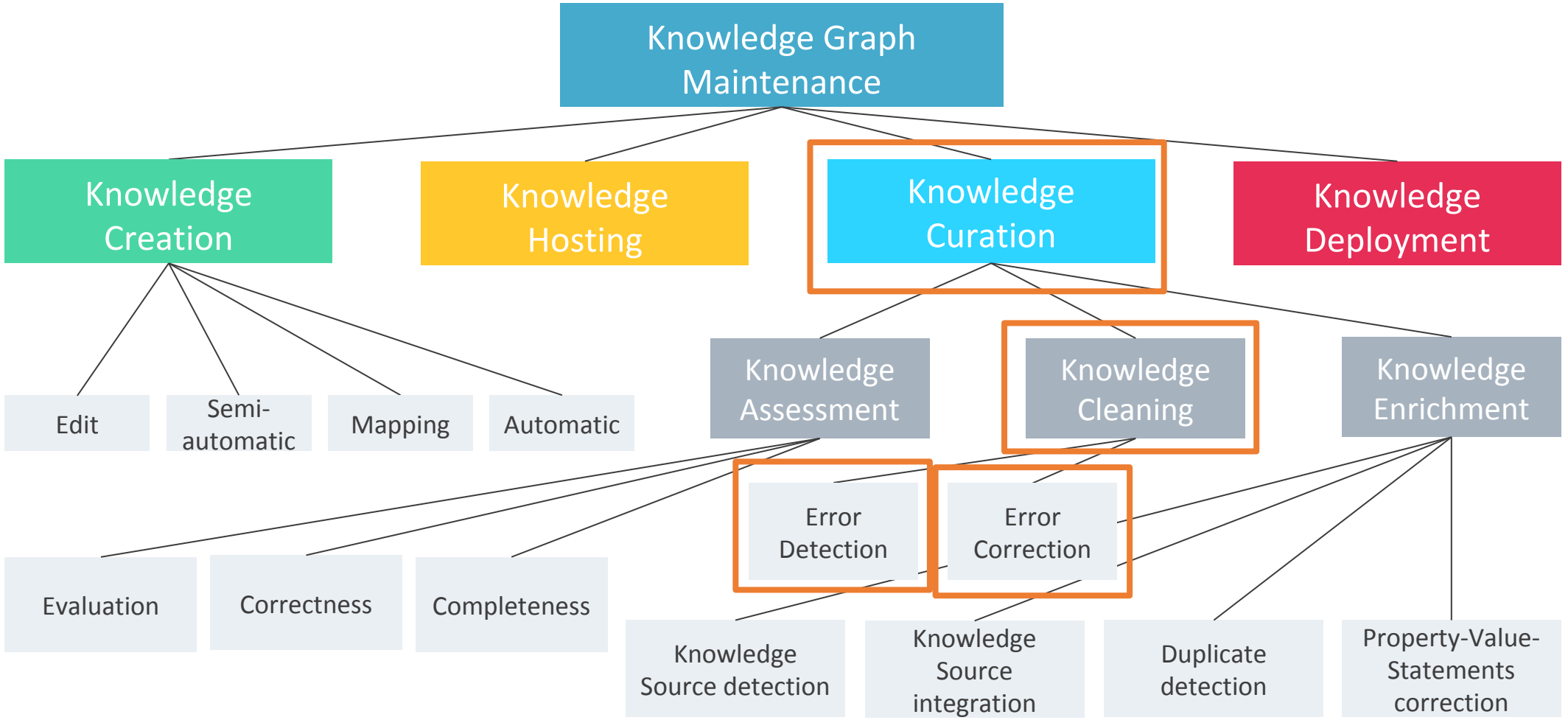
the value of property **p** on instance **i1** is **i2**

i1 is the same instance as **i2**



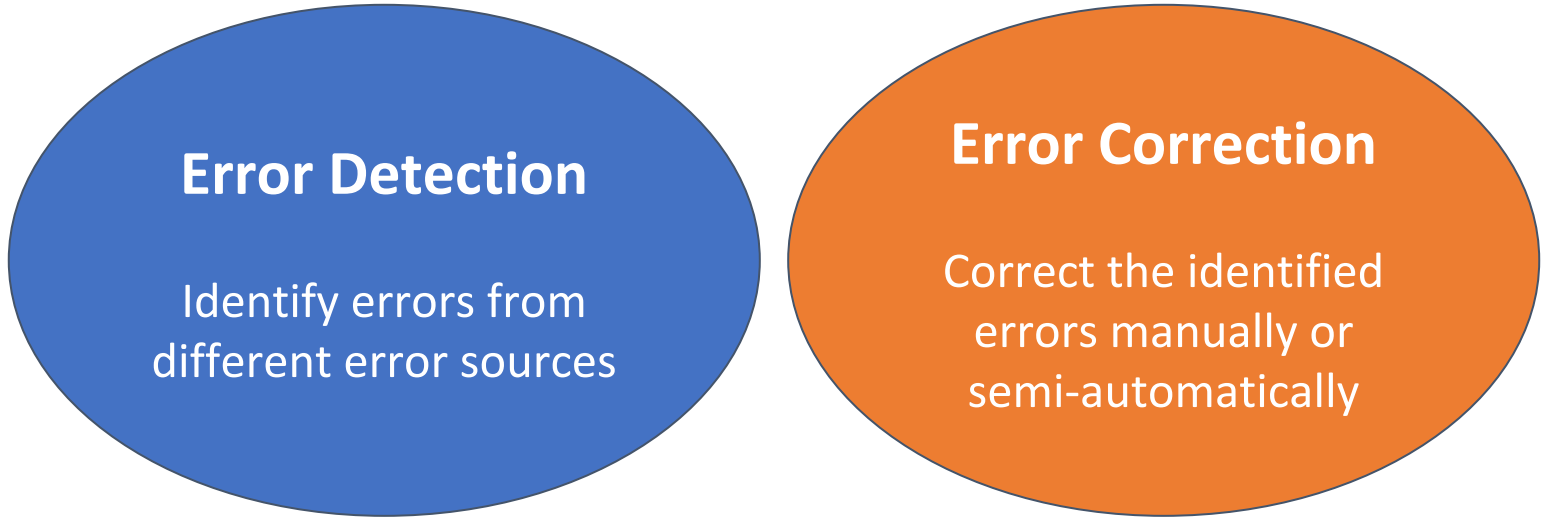
ex: http://example.org/
s: http://schema.org/
dbpedia: http://dbpedia.org/resource/

A broken Knowledge Graph


Knowledge Graph Maintenance
Knowledge Creation
Knowledge Hosting
Knowledge Curation
Knowledge Deployment
Edit
Semi-automatic
Mapping
Automatic
Knowledge Assessment
Knowledge Cleaning
Knowledge Enrichment
Error Detection
Error Correction
Evaluation
Correctness
Completeness
Knowledge Source detection
Knowledge Source integration
Duplicate detection
Property-Value-Statements correction

4. Knowledge Cleaning

Actions taken to improve the accuracy of Knowledge Graphs



Instance Assertions	• Syntactic errors in the instance identifiers • Type does not exist in the vocabulary • Assertion is semantically wrong
Property Value Assertions	• Syntactic errors in i1, i2 or p • p does not exist in the vocabulary • Domain and range violations • Assertion is semantically wrong
Equality Assertions	• Syntactic errors in i1 or i2 • Assertion is semantically wrong

Error sources and types

4. Knowledge Cleaning: Error Detection

Statistical approaches

Knowledge-driven approaches

Integrity Constraints

```
ex:LandmarkShape a sh:NodeShape;
   sh:targetClass
s:LandmarksOrHistoricalBuildings;
   sh:property [
       sh:path s:address;
       sh:class s:PostalAddress;
       sh:node [
           sh:property [
               sh:path s:streetAddress;
               sh:datatype xsd:string;
           ];
           sh:property [
               sh:path s:addressLocality;
               sh:datatype xsd:string;
           ];
. . .
```

SHACL

```
PREFIX ex: <http://example.org>
PREFIX s: <http://schema.org/>
PREFIX dbpedia:
<http://dbpedia.org/resource/>
PREFIX xsd:
<http://www.w3.org/2001/XMLSchema#>
PREFIX sh: <http://www.w3.org/ns/shacl#>

my:LandmarkShape {
  s:address {
    rdf:type s:PostalAddress;
    s:streetAddress xsd:string;
    s:addressLocality xsd:string:
    s:addressCountry s:Country;
    s:postalCode xsd:string
  };
  s:containedInPlace s:Place
}
```

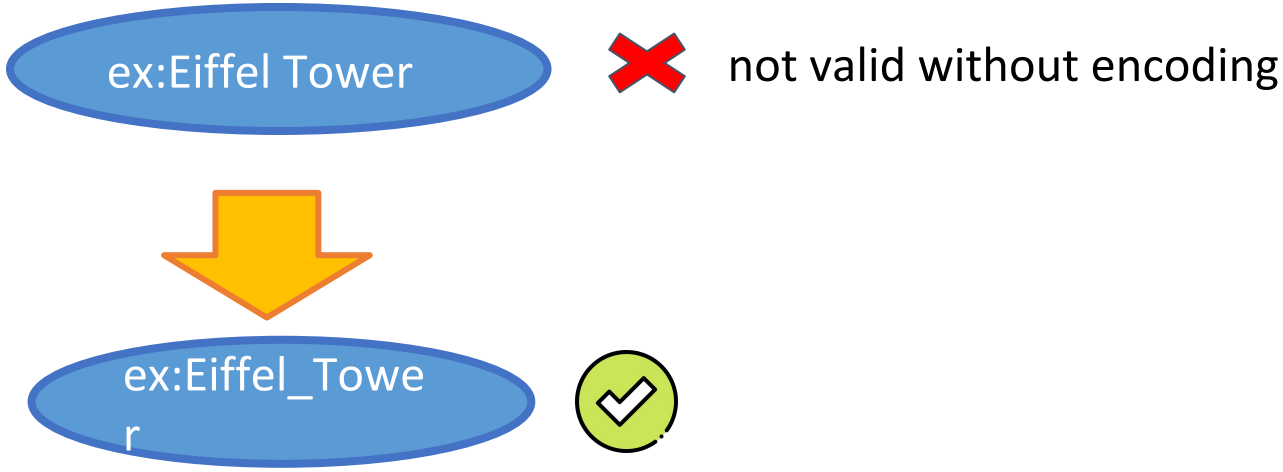
ShEx

See full examples of integrity constraints on the workshop website

4. Knowledge Cleaning: Error Correction

Wrong Instance assertions:

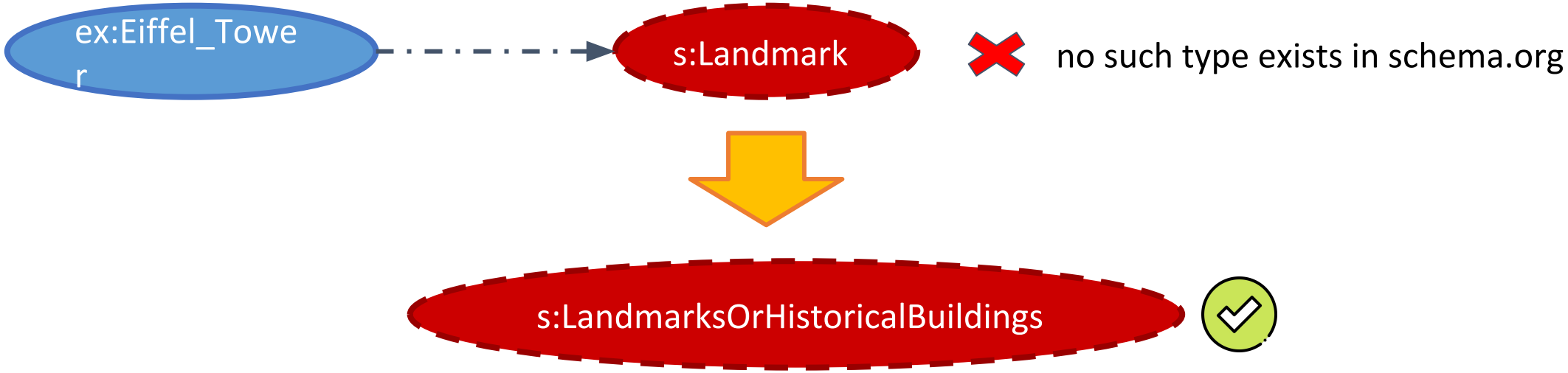
There can be syntactic errors in instance identifiers



4. Knowledge Cleaning: Error Correction

Wrong Instance assertions:

The type may not exist in the vocabulary



4. Knowledge Cleaning: Error Correction

Wrong Instance assertions:

The assertion may be semantically wrong

the Eiffel Tower is not an Event. (at least in this example)

delete the instance assertion

4. Knowledge Cleaning: Error Correction

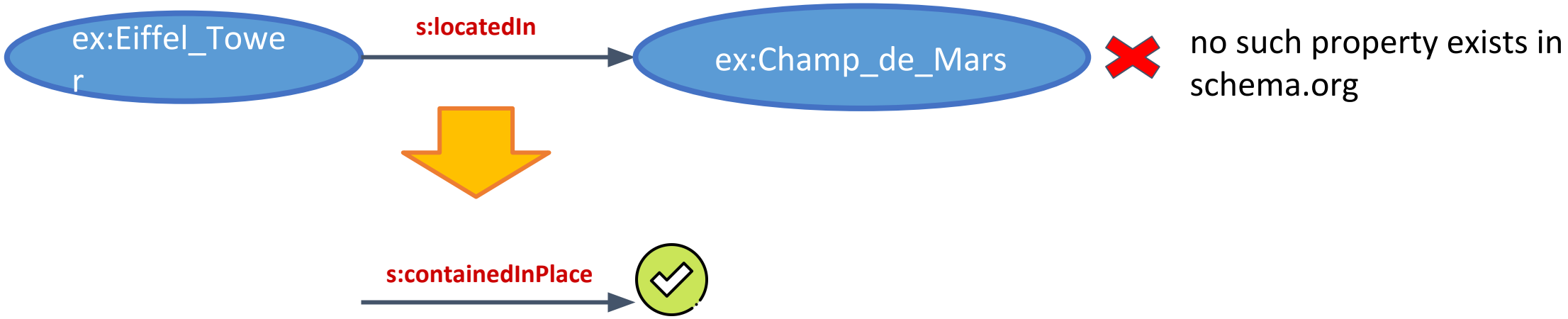
Wrong property value assertions

There may be syntactic errors in i1, i2 or p in an assertion.

4. Knowledge Cleaning: Error Correction

Wrong property value assertions

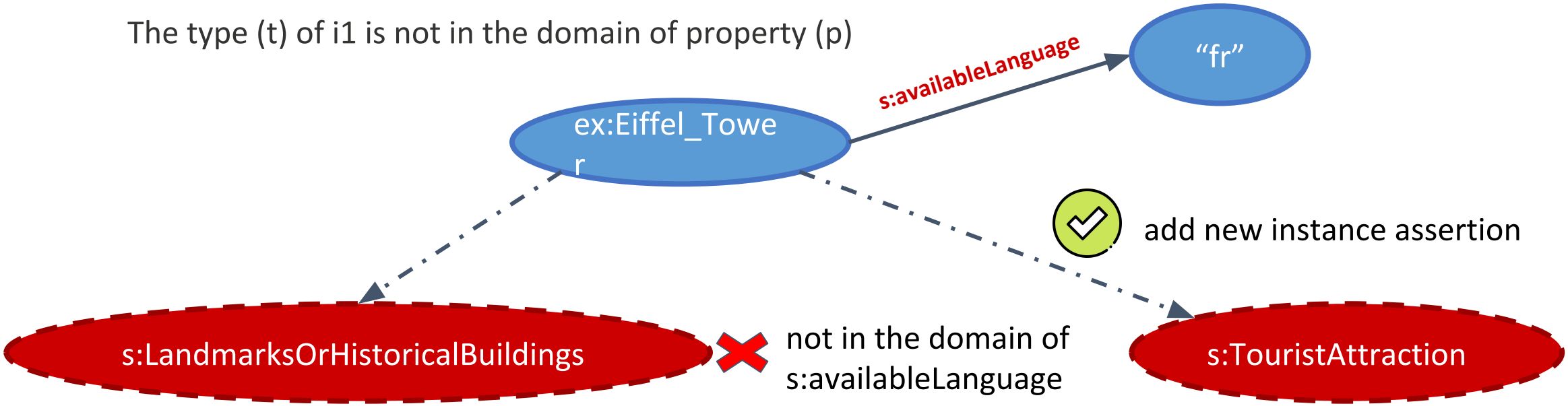
There is no property p in the vocabulary



4. Knowledge Cleaning: Error Correction

Wrong property value assertions

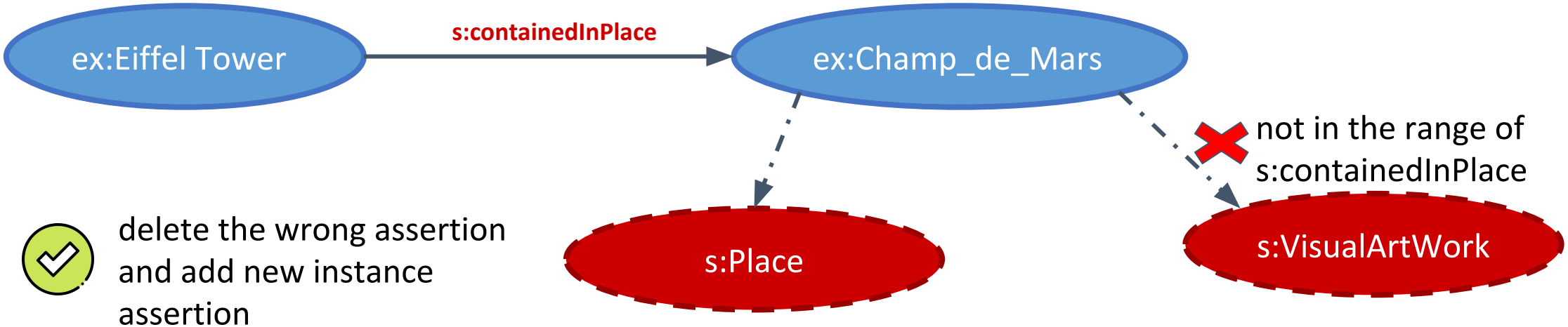
The type (t) of i1 is not in the domain of property (p)



4. Knowledge Cleaning: Error Correction

Wrong property value assertions

The type (t) of i2 is not in the range of p for any of the types in its domains

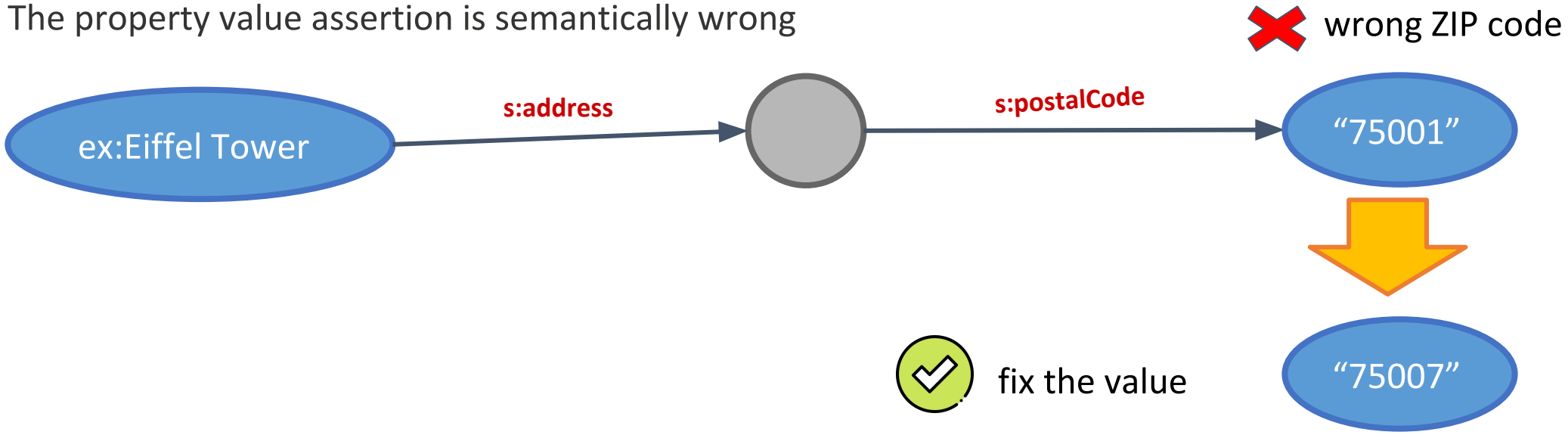


delete the wrong assertion and add new instance assertion

4. Knowledge Cleaning: Error Correction

Wrong property value assertions

The property value assertion is semantically wrong



4. Knowledge Cleaning: Error Correction

Wrong equality assertions

The i1 or i2 may be syntactically wrong

Fix the issue in a manner similar to previous error types.

4. Knowledge Cleaning: Error Correction

Wrong equality assertions

The equality assertion may be semantically wrong

two related, but different things

delete the assertion or
create a “weaker” link

4. Knowledge Cleaning: Tools

The existing tools mainly focus on detection of errors. Common approaches:

- Statistical distribution of instance and property value assertions
- Integrity constraints with SPARQL and shapes

Correction approaches typically use certain heuristics for syntactical errors and external trusted Knowledge Graphs for other error types

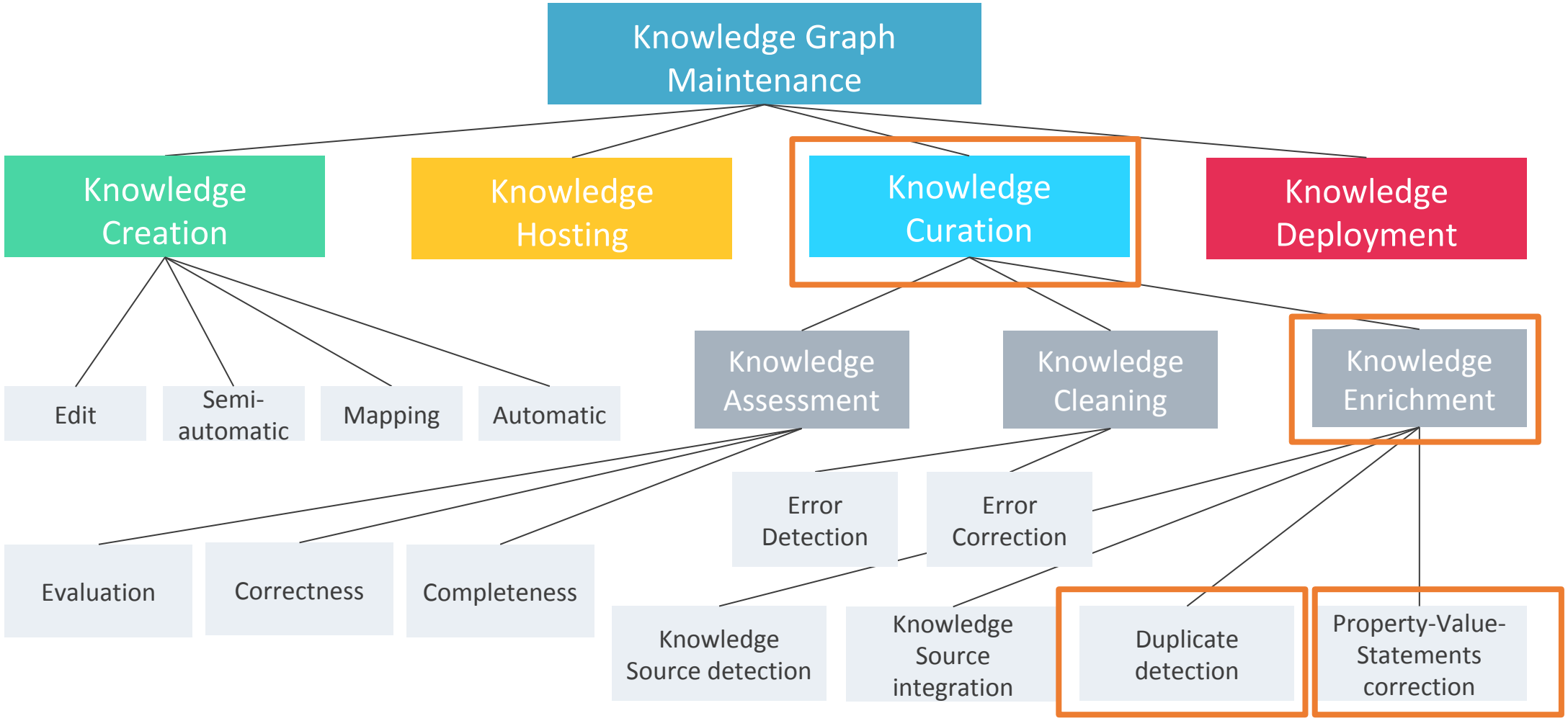
4. Knowledge Cleaning: Tools

Automating detection of semantically wrong assertions is tricky. How do we touch the “real world”?

- Take an existing, trustworthy Knowledge Graph as an oracle
- See the websites from where annotations are collected as the source of truth.

    Similar to Semantify.it Validator approach

Check out the workshop website for a list of tools for Knowledge Cleaning!


Knowledge Graph Maintenance
Knowledge Creation
Knowledge Hosting
Knowledge Curation
Knowledge Deployment
Edit
Semi-automatic
Mapping
Automatic
Knowledge Assessment
Knowledge Cleaning
Knowledge Enrichment
Evaluation
Correctness
Completeness
Error Detection
Error Correction
Knowledge Source detection
Knowledge Source integration
Duplicate detection
Property-Value-Statements correction

4. Knowledge Enrichment

A process for improving the completeness of a knowledge graph by adding new statements

4. Knowledge Enrichment: A Process Model

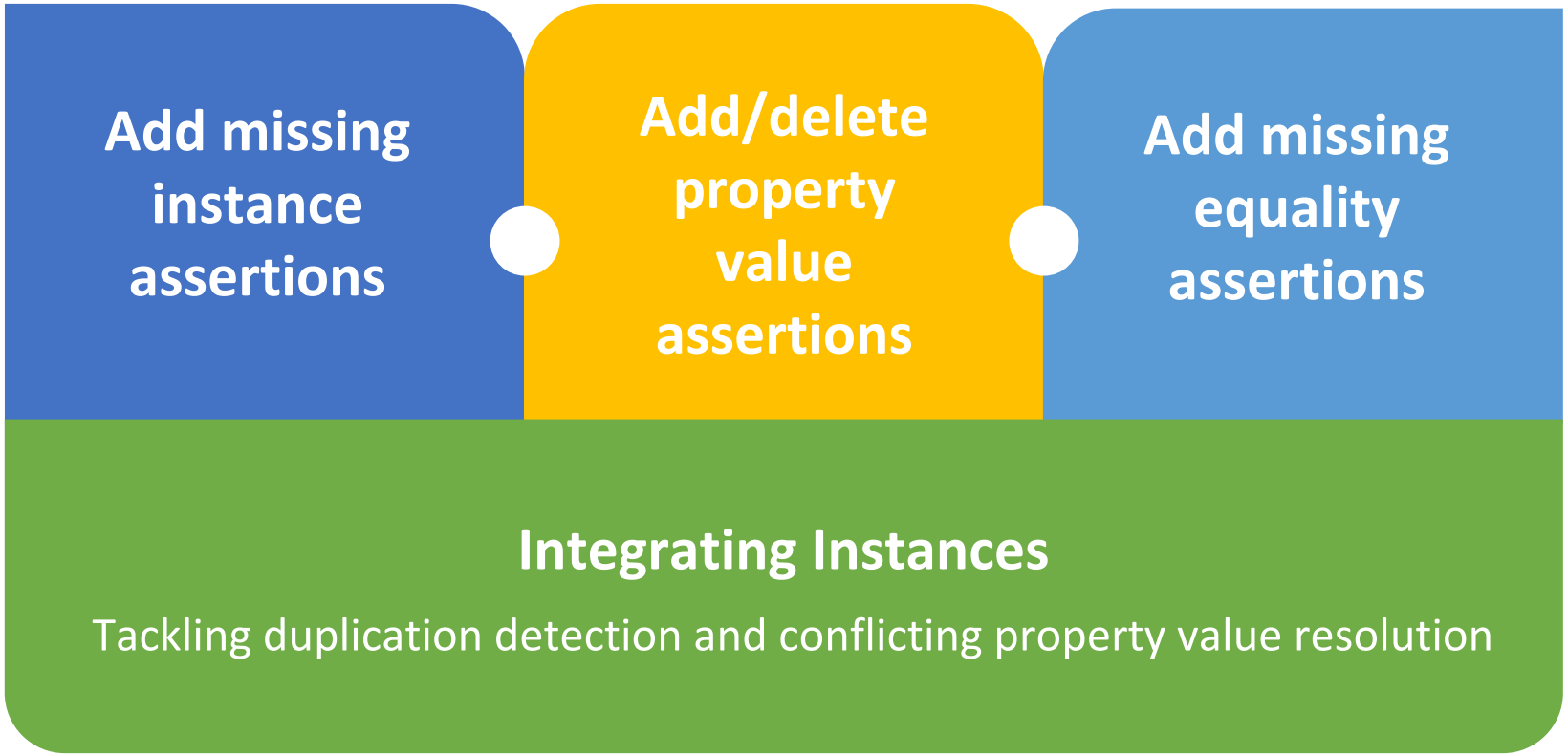
Identify New Sources → **Integrate the Schema** → **Integrate the Instances**

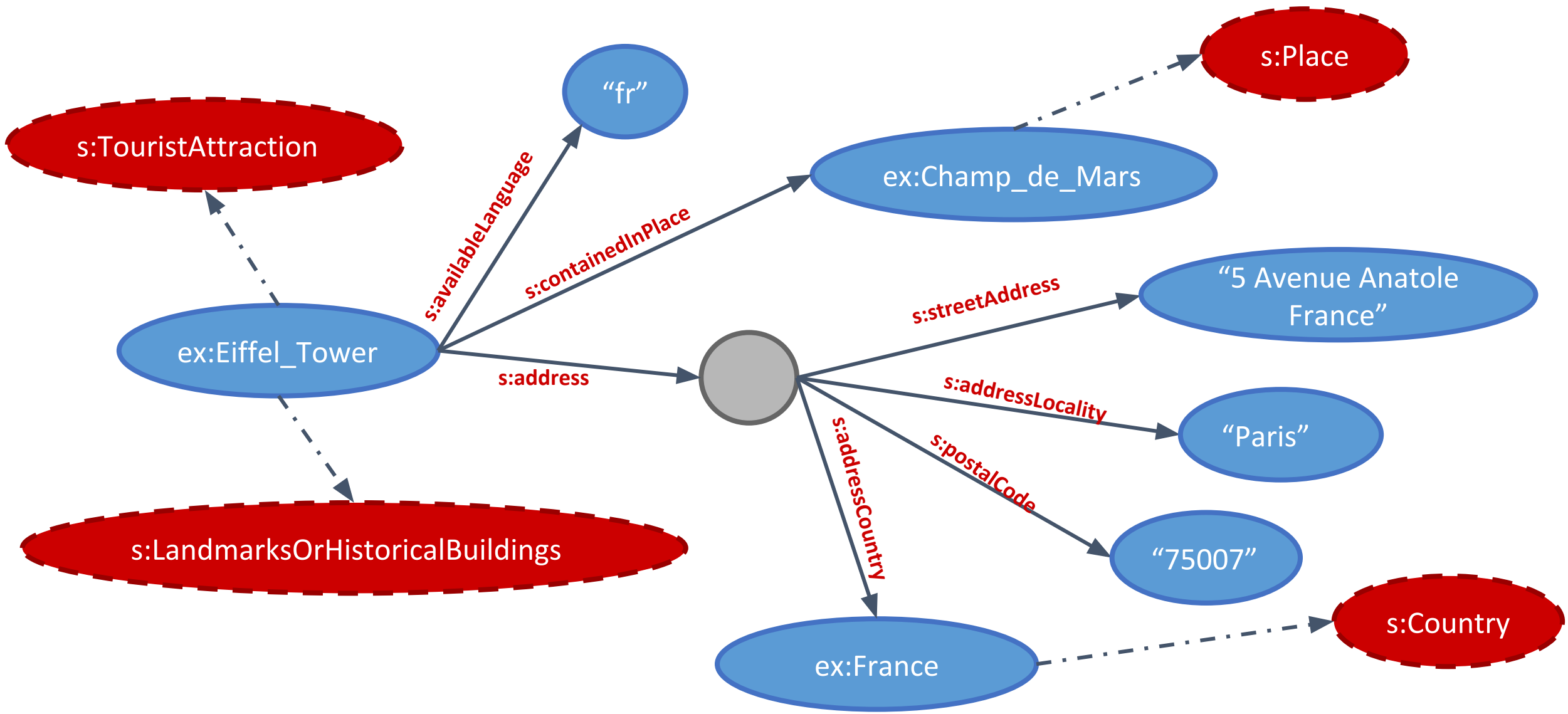
This process can be automated to some extend for Open Knowledge Graphs. Identifying proprietary sources automatically is tricky.

The relevant parts of the schemas of new sources are mapped to schema.org

Two major issues:

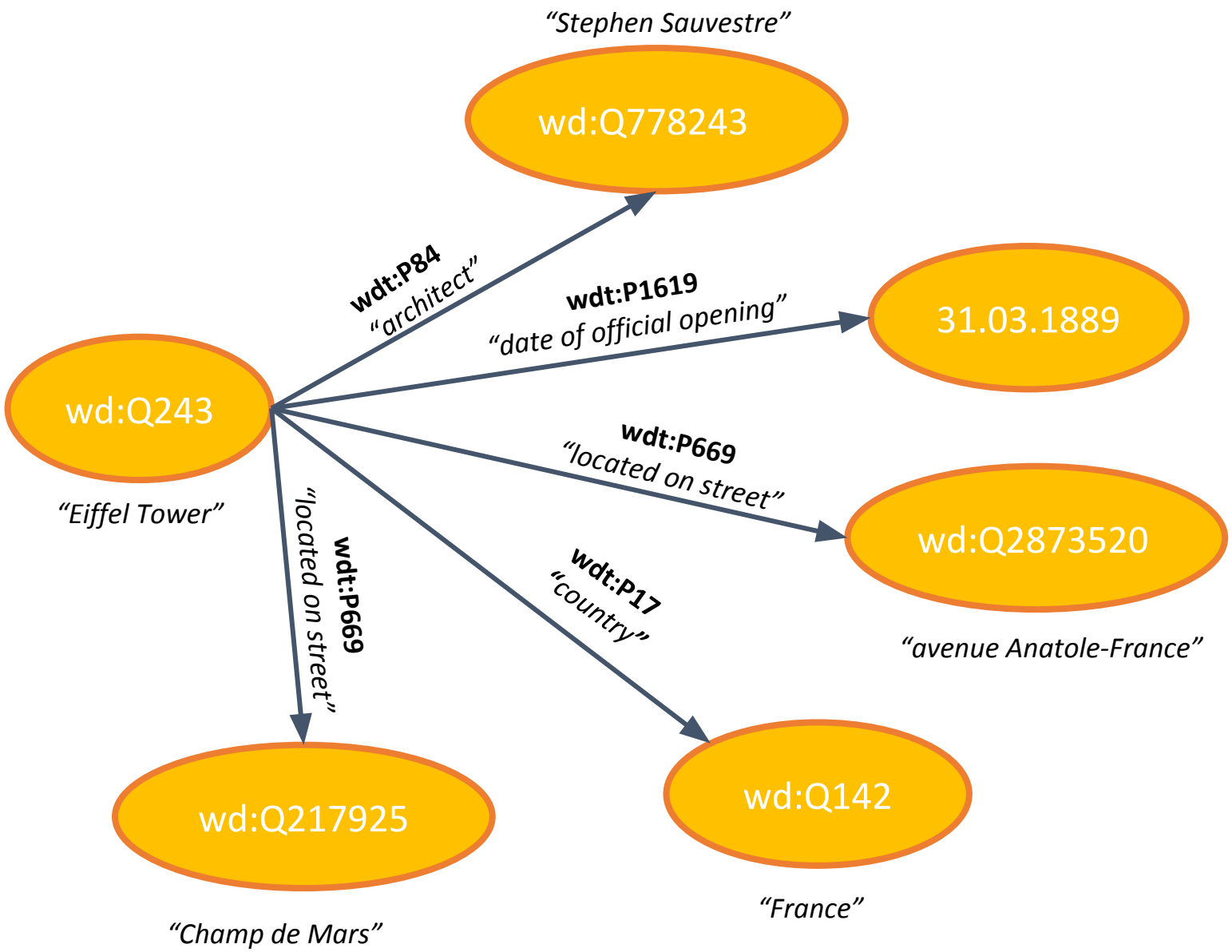
1. Identifying and resolving duplicates
2. Resolving conflicting property value assertions


Add missing instance assertions
Add/delete property value assertions
Add missing equality assertions
Integrating Instances
Tackling duplication detection and conflicting property value resolution



ex: http://example.org/
s: http://schema.org/

The clean Knowledge Graph



An excerpt from the Wikidata entity of Eiffel Tower

4. Knowledge Enrichment

Assume, we want to enrich the landmarks in our Knowledge Graph.

Identify New Sources → **Integrate the Schema** → **Integrate the Instances**

Schema.org Type	Wikidata Type	Schema.org Property*	Wikidata Property
LandmarksOrHistoricalBuildings	landmark	address/streetAddress	located on street.label
		address/addressCountry	country
		ex:architect	architect
		ex:openingDate	date of official opening

*Includes properties from an extension

Identify New Sources → **Integrate the Schema** → **Integrate the Instances**

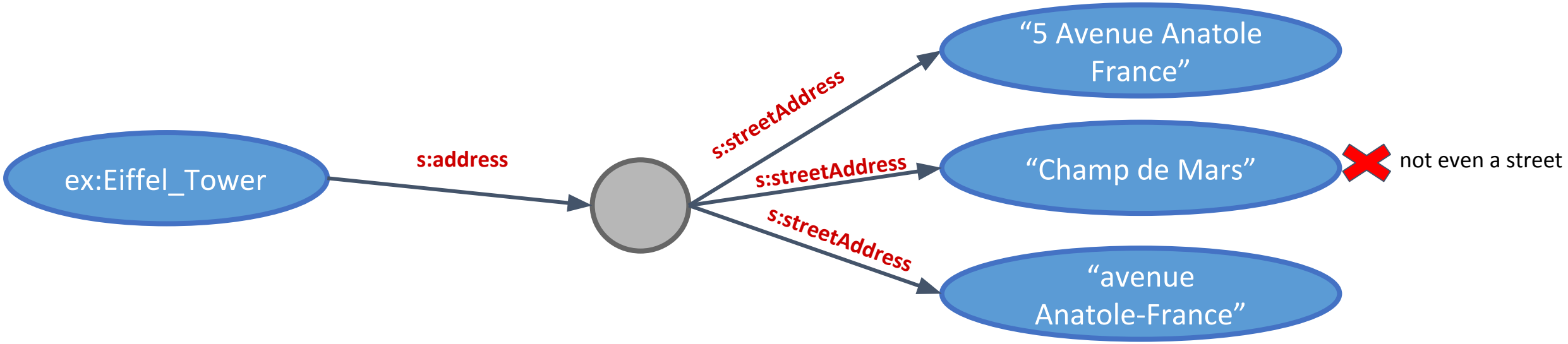
WIKIDATA

We found a duplicate instance after integrating landmark instances from Wikidata. Identification of duplicates is typically done by applying similarity metrics to a set of property values on both instances.

Identify New Sources → **Integrate the Schema** → **Integrate the Instances**

WIKIDATA

Duplicates



Too many street addresses!
Delete two property value assertions

Identify New Sources → **Integrate the Schema** → **Integrate the Instances**

WIKIDATA

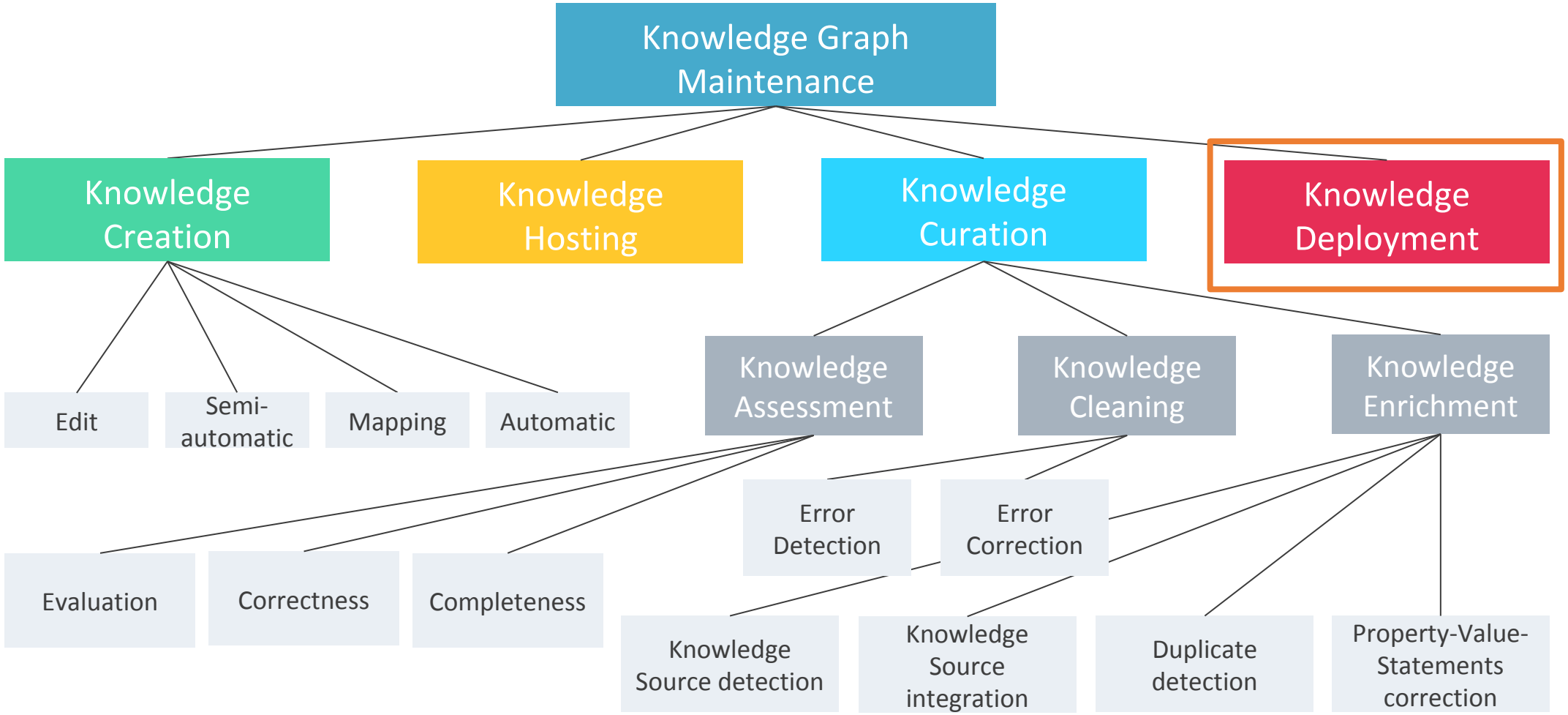
Duplicates **Conflicting Property Values**

DEMO

Duplication Detection with Duke

Check out the workshop website for more tools for Knowledge Enrichment!

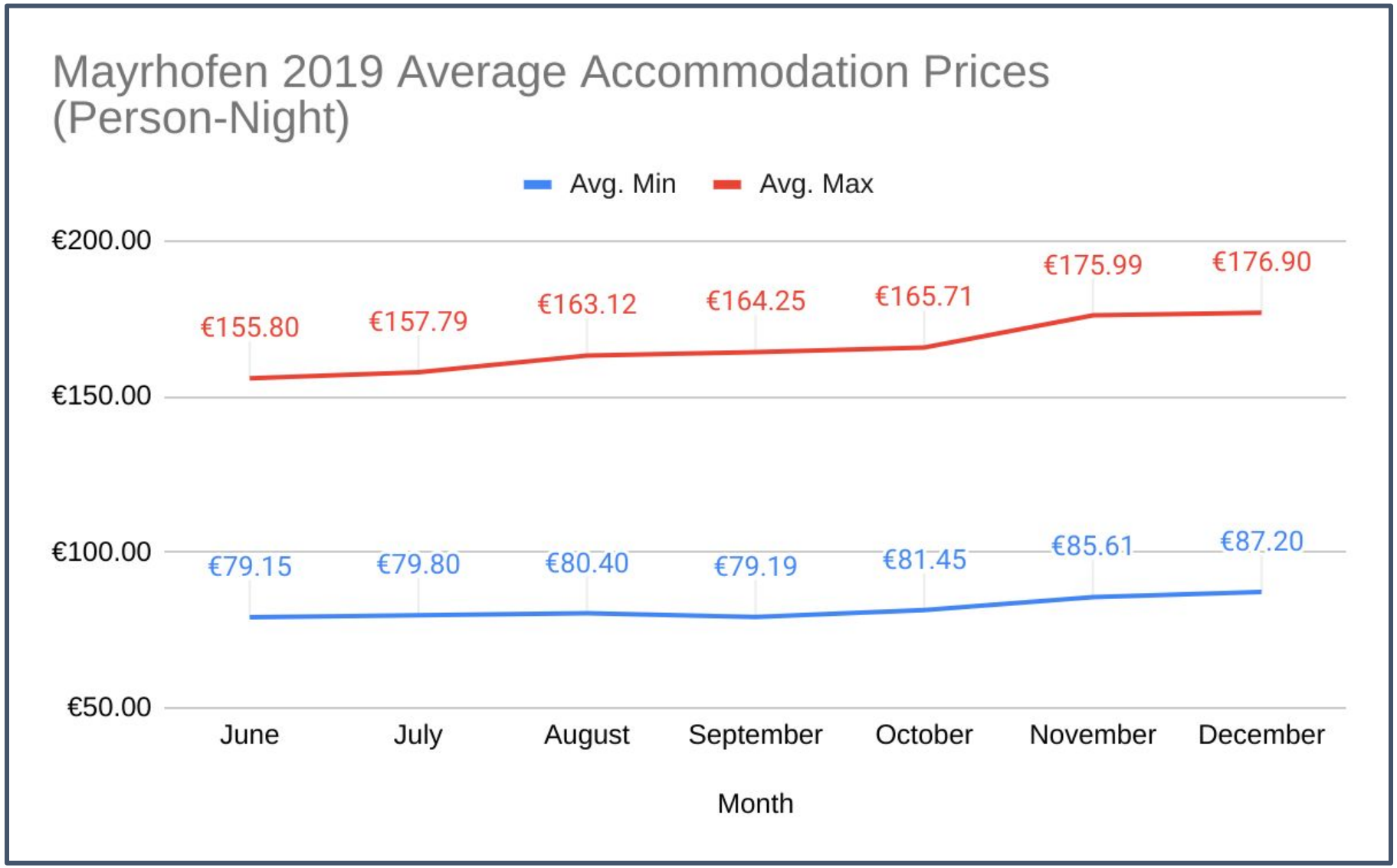
5. KNOWLEDGE DEPLOYMENT


Knowledge Graph Maintenance
Knowledge Creation
Knowledge Hosting
Knowledge Curation
Knowledge Deployment
Edit
Semi-automatic
Mapping
Automatic
Knowledge Assessment
Knowledge Cleaning
Knowledge Enrichment
Evaluation
Correctness
Completeness
Error Detection
Error Correction
Knowledge Source detection
Knowledge Source integration
Duplicate detection
Property-Value-Statements correction

5. Knowledge Deployment: Tyrolean Knowledge Graph

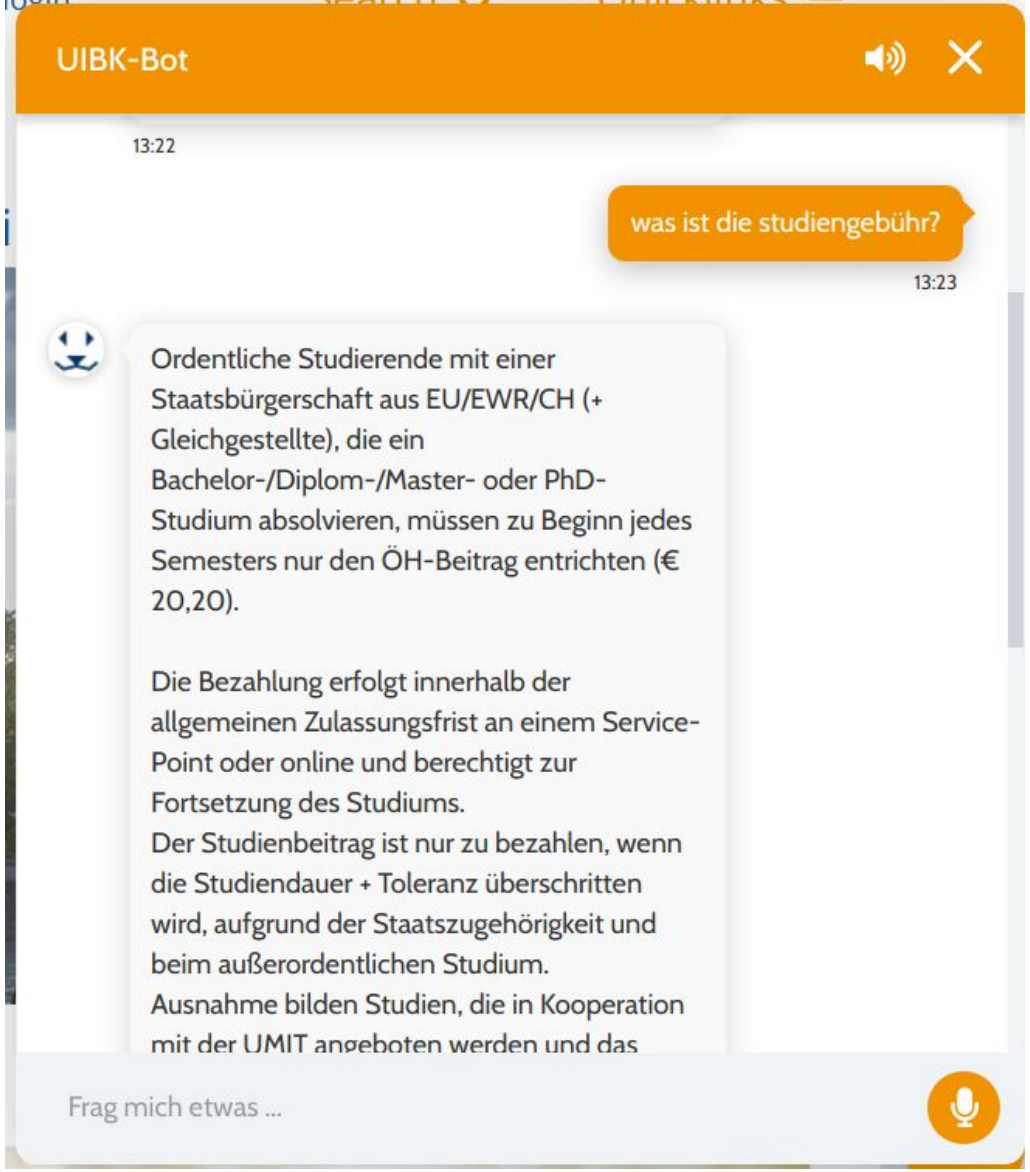
- A Knowledge Graph in the tourism domain.
- ~30M triples per day from a dozen sources
- Historical data is tracked with provenance
- Various use cases: time-series analysis, conversational agents...

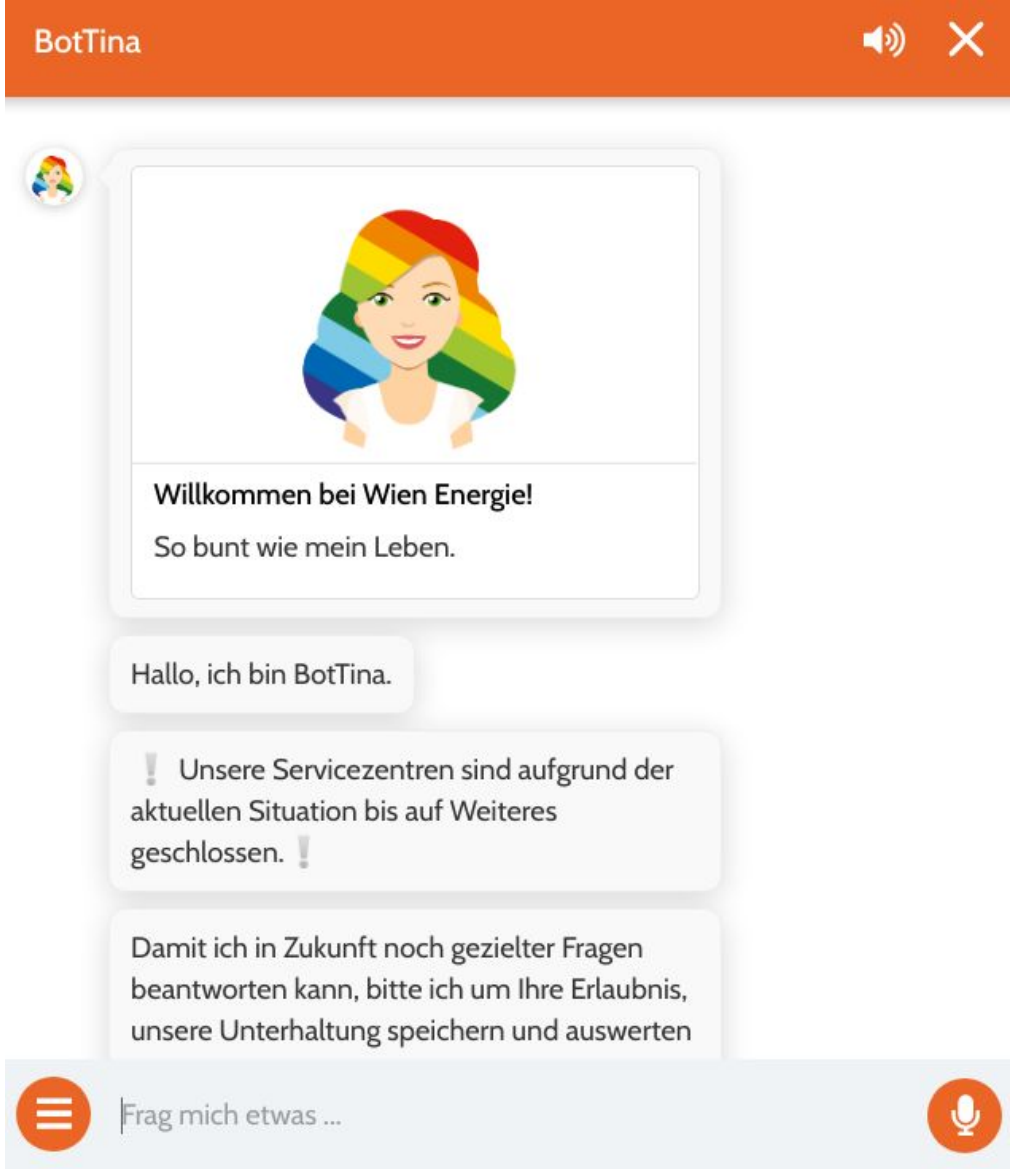
Try it out!
https://tirol.kg


Mayrhofen 2019 Average Accommodation Prices (Person-Night)
Avg. Min
Avg. Max
€200.00
€150.00
€100.00
€50.00
€155.80
€157.79
€163.12
€164.25
€165.71
€175.99
€176.90
€79.15
€79.80
€80.40
€79.19
€81.45
€85.61
€87.20
June
July
August
September
October
November
December
Month

5. Knowledge Deployment: DACH-KG

- Several Destination Management and Marketing Organizations from Germany, Austria, Switzerland and Northern Italy
- The goal is to create standardized schemas for Knowledge Graphs in the tourism domain.
- One of the first applications of these schemas will be the German Tourism Knowledge Graph, contracted by the Germany Tourism Board





MindLab Chatbots from Onlim

5. Knowledge Deployment

- training of ML models based on KGs
  - due to the RDF nature data in KGs is semantically described
  - good training data for ML models
- conversational agents
  - chatbots
  - intelligent personal assistants
  - **question answering over LinkedData**
- OpenData sharing platforms
  - currently Open(Government)Data often makes little sense (scanned pdfs, weird spreadsheets,csv, …)
  - LinkedData is self explaining (see lod-cloud https://lod-cloud.net)

6. OUTLOOK

6. Outlook

We have seen a lifecycle for Knowledge Graphs, from their creation to deployment.

The assessment, cleaning and enrichment processes are crucial for making Knowledge Graphs a useful resource.

but...

Does it scale?

6. Outlook

Knowledge Graphs are..

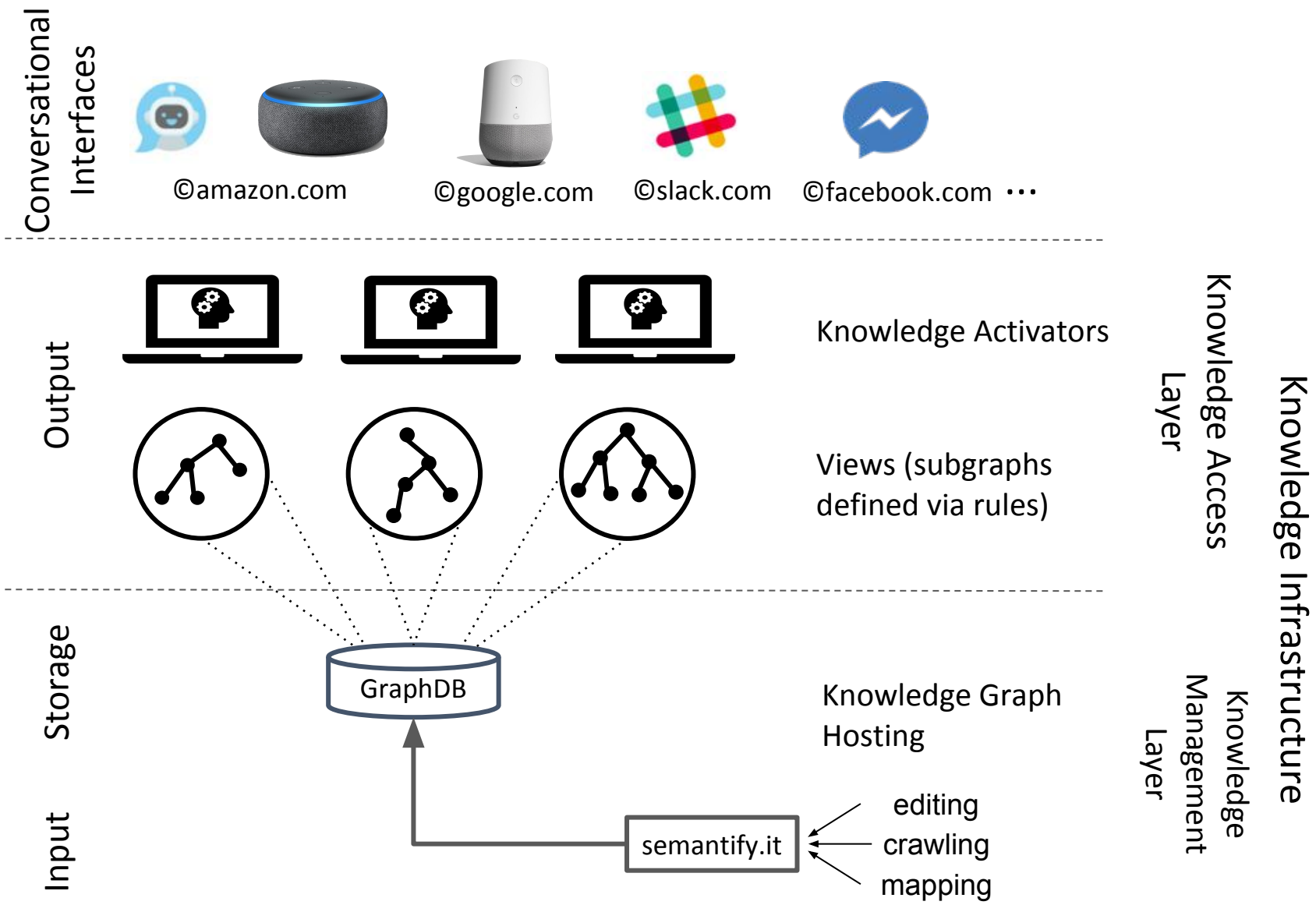
LARGE

HETEROGENOUS

6. Outlook

- For efficient and effective Knowledge Curation

  - Reduce the size of the Knowledge Graph
  - Support different application contexts (i.e. point of views)


Conversational Interfaces
©amazon.com
©google.com
©slack.com
©facebook.com ...
Output
Knowledge Activators
Views (subgraphs defined via rules)
Storage
GraphDB
Knowledge Graph Hosting
Input
semantify.it
editing
crawling
mapping
Knowledge Access Layer
Knowledge Management Layer
Knowledge Infrastructure

@eliaska
@umutsims

Think GREEN
Only print if it's essential

www.uibk.ac.at

References

[Batini & Scannapieco, 2006] Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications, Springer (2006). https://doi.org/10.1007/3-540-33173-5

[Dimou et al.,, 2014] Dimou, A., Sande, M.V., Colpaert, P., Verborgh, R., Mannens, E., de Walle, R.V.: RML: A generic language for integrated RDF mappings of heterogeneous data. In: Proceedings of the Workshop on Linked Data on the Web (LDOW2014) colocated with the 23rd International World Wide Web Conference (WWW2014), Seoul, Korea, April 8, 2014. CEUR Workshop Proceedings, vol. 1184. CEUR- WS.org (2014), http://ceur-ws.org/Vol-1184/ldow2014_paper_01.pdf

[Färber et al., 2018] Farber, M., Bartscherer, F., Menne, C., Rettinger, A.: Linked data quality of dbpedia, freebase, opencyc, wikidata, and YAGO. Semantic Web Journal 9(1), 77–129 (2018). https://doi.org/10.3233/SW-170275

[Pipino et al., 2002] Pipino, L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Communications of the ACM 45(4), 211–218 (2002). https://doi.org/10.1145/505248.5060010

The “tick” Icon used throughout the slides made by Freepik from www.flaticon.com

The Knowledge Graph Lifecycle and Task Model diagrams are drawn by Onlim GmbH.

References

[Wang, 1998] Wang, R.Y.: A product perspective on total data quality management. Communication of the ACM 41(2), 58–65 (1998). https://doi.org/10.1145/269012.269022

[Wang & Strong, 1996] Wang, R.Y., Strong, D.M.: Beyond accuracy: What data quality means to data consumers. Journal of Management Information Systems 12(4), 5–33 (1996), http://www.jmis-web.org/articles/1002

[Wang et al., 2001] Wang, R.Y., Ziad, M., Lee, Y.W.: Data Quality, Advances in Database Systems, vol. 23. Kluwer Academic Publisher (2001). https://doi.org/10.1007/b116303

[Zaveri et al., 2016] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality assessment for linked data: A survey. Semantic Web Journal 7(1), 63–93 (2016)