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
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/
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?

**Graphs**
a natural and pragmatic way of structuring knowledge.

**No Schema**
at least not in the sense of Relational Databases.

**Heterogenous**
data from different sources can be easily integrated

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

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.
2. KNOWLEDGE CREATION

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

```html
<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.

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)

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.
Der Imbißstand 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.

- **Domain Specification Modeling**
  - Evaluation and analysis of the annotations
  - Annotation development and deployment
  - Domain definition and mapping to semantic vocabularies
  - Defining a vocabulary based on restricting and extending semantic vocabularies
  - Analysis of domain entities and their online representation

- **Mapping according to domain specifications**
- **Annotation development according to domain specifications**

Preparation for modeling → Application of models
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

<table>
<thead>
<tr>
<th>Preparation for modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation and analysis of the annotations</td>
</tr>
<tr>
<td>Annotation development and deployment</td>
</tr>
<tr>
<td>Domain definition and mapping to semantic vocabularies</td>
</tr>
<tr>
<td>Defining a vocabulary based on restricting and extending semantic vocabularies</td>
</tr>
<tr>
<td>Analysis of domain entities and their online representation</td>
</tr>
</tbody>
</table>
2. Knowledge Creation - Methodology

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
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
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)

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

<table>
<thead>
<tr>
<th>Domain</th>
<th>Property</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>s:LandmarksOrHistoricalBuildings</td>
<td>s:address</td>
<td>s:PostalAddress</td>
</tr>
<tr>
<td></td>
<td>s:containedInPlace</td>
<td>s:Place</td>
</tr>
<tr>
<td>s:PostalAddress</td>
<td>s:streetAddress</td>
<td>s:Text</td>
</tr>
<tr>
<td></td>
<td>s:addressLocality</td>
<td>s:Text</td>
</tr>
<tr>
<td></td>
<td>s:addressCountry</td>
<td>s:Country</td>
</tr>
<tr>
<td></td>
<td>s:postalCode</td>
<td>s:Text</td>
</tr>
<tr>
<td>s:TouristAttraction</td>
<td>s:availableLanguage</td>
<td>s:Text</td>
</tr>
</tbody>
</table>

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

---

KGC 2020 | Kärle & Simsek | May 4th, 2020
2. RocketRML - Performance

![JSON Format Chart](chart1.png)

![XML Format Chart](chart2.png)

KGC 2020 | Kärle & Simsek | May 4th, 2020
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
1 { "persons": [
2   { "firstname": "Elias",
3     "lastname": "Kärle",
4     "speak": [ "de", "en", "fr", "Tyrolean"]
5   },
6   { "firstname": "Umutcan",
7     "lastname": "Simsek",
8     "speak": [ "tr", "en", "de", "Hessisch"]
9   }
10 ]
```

Mapping file (YARRRML*):

```yaml
1 prefixes:
2   schema: "http://schema.org/"
3   myfunc: "http://myfunc.com/"
4 mappings:
5   person:
6     sources:
7       - ["input-jsonpath", 
8         
9         "$.persons[*]"
10     ]
11   po: [ a, schema:Person]
12   [ [schema:name, $(firstname)]
13     [schema:language, $(speaks.*)]
```

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

GET /annotation/{annotationId}

GET /annotation/{annotationId}/statistics

GET /organisation/{organisationId}/annotation

GET /website/{websiteId}/annotation
2. Knowledge Creation

**Evaluator**

validation & verification

- **verification** against schema.org
- **verification** against DS
- **validation** against website →
<table>
<thead>
<tr>
<th>URL Path</th>
<th>Crawling</th>
<th>Schema.org Verification</th>
<th>Domain-specific Verification</th>
<th>Annotation Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>66</td>
</tr>
<tr>
<td>/annotating-e-hotel-offering-rooms-with-the-new-schema-org-version-3-1/</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>/annotating-sk-resorts-lifts-and-slopes-with-schema-org/</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>/becoming-an-entity-in-the-google-knowledge-graph/</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>/bibTeX-style-for-simon-press-journal-of-mobile-multimedia-ppm/</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>/blog/</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>/projects/</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>82</td>
</tr>
<tr>
<td>/publications/</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>43</td>
</tr>
<tr>
<td>/sitemap.html</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>
# 2. Knowledge Creation - tools - semantify.it

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Score</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>author.name</td>
<td>Elias Karslo</td>
<td>100</td>
<td>high</td>
</tr>
<tr>
<td>datePublished</td>
<td>2017-01-01</td>
<td>0</td>
<td>medium</td>
</tr>
<tr>
<td>headline</td>
<td>semantify: it, a Platform for Creation, Publication and Distribution of Semantic Annotations</td>
<td>100</td>
<td>high</td>
</tr>
<tr>
<td>image</td>
<td><a href="https://semantify-it.org/images/logo_text.png">https://semantify-it.org/images/logo_text.png</a></td>
<td>100</td>
<td>high</td>
</tr>
<tr>
<td>publisher:logo:url</td>
<td><a href="http://www.thinkermind.org/images/logo.png">http://www.thinkermind.org/images/logo.png</a></td>
<td>0</td>
<td>medium</td>
</tr>
<tr>
<td>publisher:name</td>
<td><a href="http://www.thinkermind.org/index.php?view=article&amp;articleid=semapru_2017_2_10_30007">http://www.thinkermind.org/index.php?view=article&amp;articleid=semapru_2017_2_10_30007</a></td>
<td>100</td>
<td>high</td>
</tr>
<tr>
<td>date:modified</td>
<td>2020-09-28T15:56</td>
<td>0</td>
<td>medium</td>
</tr>
</tbody>
</table>
3. KNOWLEDGE HOSTING
Knowledge Graph Maintenance

Knowledge Creation
- Edit
- Semi-automatic
- Mapping
- Automatic

Knowledge Hosting
- Knowledge Assessment
- Error Detection
- Knowledge Source detection

Knowledge Curation
- Knowledge Cleaning
- Error Correction
- Knowledge Source integration

Knowledge Deployment
- Knowledge Enrichment
- 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
30 years
from Innsbruck
researcher

schema:name = “Max”
schema:birthDate = “1990”
schema:homeLocation = “Innsbruck”
schema:hasOccupation = “researcher”
3. Knowledge Hosting

But what is RDF?

Resource Description Framework

<table>
<thead>
<tr>
<th>Subject (s)</th>
<th>Predicate (p)</th>
<th>Object (o)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td><a href="http://max.cc">http://max.cc</a></td>
<td>is a</td>
</tr>
<tr>
<td></td>
<td>rdf:type</td>
<td>Person</td>
</tr>
<tr>
<td></td>
<td></td>
<td>schema:Person</td>
</tr>
<tr>
<td>Max</td>
<td><a href="http://max.cc">http://max.cc</a></td>
<td>has name</td>
</tr>
<tr>
<td></td>
<td>schema:name</td>
<td>Max</td>
</tr>
<tr>
<td></td>
<td></td>
<td>schema:Text</td>
</tr>
<tr>
<td>Max</td>
<td><a href="http://max.cc">http://max.cc</a></td>
<td>was born in</td>
</tr>
<tr>
<td></td>
<td>schema:birthDate</td>
<td>1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td>schema:Date</td>
</tr>
<tr>
<td>Max</td>
<td><a href="http://max.cc">http://max.cc</a></td>
<td>lives in</td>
</tr>
<tr>
<td></td>
<td>schema:homeLocation</td>
<td>Innsbruck</td>
</tr>
<tr>
<td></td>
<td></td>
<td>schema:Place</td>
</tr>
<tr>
<td>Max</td>
<td><a href="http://max.cc">http://max.cc</a></td>
<td>works as a</td>
</tr>
<tr>
<td></td>
<td>schema:hasOccupation</td>
<td>researcher</td>
</tr>
<tr>
<td></td>
<td></td>
<td>schema:Occupation</td>
</tr>
</tbody>
</table>
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)
   
   ```json
   {"@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](http://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
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 in named graphs
- ~12 Billion statements
3. Knowledge Hosting

Hosting as Knowledge Graph:

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

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th># triples tested with</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Oracle Spatial and Graph with Oracle Database 12c</td>
<td>1.08 T</td>
</tr>
<tr>
<td>2</td>
<td>AnzoGraph DB by Cambridge Semantics</td>
<td>1.065 T</td>
</tr>
<tr>
<td>3</td>
<td>AllegroGraph</td>
<td>1+ T</td>
</tr>
<tr>
<td>4</td>
<td>Stardog</td>
<td>50 B</td>
</tr>
<tr>
<td>5</td>
<td>OpenLink Virtuoso v7+</td>
<td>39.8 B</td>
</tr>
<tr>
<td>6</td>
<td>GraphDB™ by Ontotext</td>
<td>17 B</td>
</tr>
</tbody>
</table>
4. KNOWLEDGE CURATION
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:

\[ m_{\text{VariousLang}}(r) = \begin{cases} 
1 & \text{labels provided in English and one other language} \\
0.5 & \text{labels provided in only one language} \\
0 & \text{otherwise}
\end{cases} \]
4. Knowledge Assessment: Metrics

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

Example metric calculation from Relevancy dimension:

\[
m_{\text{DomainCoverage}}(r) = \frac{\text{Average DS Property Occurrence on an Instance in } r}{|\text{Properties of DS}|}
\]
4. Knowledge Assessment: A Process Model

Decide on Dimension Weights

Each dimension may have different levels 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
<table>
<thead>
<tr>
<th>Domain</th>
<th>Property</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>s:LandmarksOrHistoricalBuildings</td>
<td>s:address</td>
<td>s:PostalAddress</td>
</tr>
<tr>
<td></td>
<td>s:containedInPlace</td>
<td>s:Place</td>
</tr>
<tr>
<td>s:PostalAddress</td>
<td>s:streetAddress</td>
<td>s:Text</td>
</tr>
<tr>
<td></td>
<td>s:addressLocality</td>
<td>s:Text</td>
</tr>
<tr>
<td></td>
<td>s:addressCountry</td>
<td>s:Country</td>
</tr>
<tr>
<td></td>
<td>s:postalCode</td>
<td>s:Text</td>
</tr>
<tr>
<td>s:TouristAttraction</td>
<td>s:availableLanguage</td>
<td>s:Text</td>
</tr>
</tbody>
</table>

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
A broken Knowledge Graph
4. Knowledge Cleaning

Actions taken to improve the accuracy of Knowledge Graphs

- **Error Detection**
  - Identify errors from different error sources

- **Error Correction**
  - Correct the identified errors manually or semi-automatically
Error sources and types

- **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
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;
      ];
    ];
  ];

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

ex:Eiffel Tower  not valid without encoding

ex:Eiffel_Tower
4. Knowledge Cleaning: Error Correction

Wrong Instance assertions:
The type may not exist in the vocabulary

```
ex:Eiffel_Tower  s:Landmark  no such type exists in schema.org
```

```
s:LandmarksOrHistoricalBuildings
```

KGC 2020 | Kärle & Simsek | May 4th, 2020
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.

ex:Eiffel Tower \( \xrightarrow{s:locatedIn} \) ex:Champ_de_Mars

```
| ex:Eiffel_Tower | ✓ |
```

i1 is not valid without encoding
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 t2 is not in the range of p for any of the types in its domains

- ex:Eiffel Tower
  - s:containedInPlace
  - ex:Champ_de_Mars
    - not in the range of s:containedInPlace
    - s:Place
    - s:VisualArtWork

*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

ex:Eiffel Tower \(\text{s:address}\) \(\text{s:postalCode}\) “75001”

wrong ZIP code

fix the value “75007”
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

- ex:Eiffel Tower
- dbpedia:Paris_Las_Vegas

- 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!
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**

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

**Integrate the Schema**

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

**Integrate the Instances**

Two major issues:

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

Tackling duplication detection and conflicting property value resolution
The clean Knowledge Graph

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

s:TouristAttraction

s:LandmarksOrHistoricalBuildings

"fr"

"5 Avenue Anatole France"

"Paris"

"75007"

s:Country

s:Place

s:containInPlace

s:availableLanguage

s:address

s:addressCountry

s:addressLocality

s:postalCode

ex:France

ex:Eiffel_Tower

ex:Champ_de_Mars
An excerpt from the Wikidata entity of Eiffel Tower
4. Knowledge Enrichment

Assume, we want to enrich the landmarks in our Knowledge Graph.
<table>
<thead>
<tr>
<th>Schema.org Type</th>
<th>Wikidata Type</th>
<th>Schema.org Property*</th>
<th>Wikidata Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>LandmarksOrHistoricalBuildings</td>
<td>landmark</td>
<td>address/streetAddress</td>
<td>located on street.label</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address/addressCountry</td>
<td>country</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ex:architect</td>
<td>architect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ex:openingDate</td>
<td>date of official opening</td>
</tr>
</tbody>
</table>

*Includes properties from an extension

Identify New Sources  >  Integrate the Schema  >  Integrate the Instances
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.
Too many street addresses! Delete two property value assertions.
DEMO

Duplication Detection with Duke

Check out the workshop website for more tools for Knowledge Enrichment!
5. KNOWLEDGE DEPLOYMENT
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)

€200.00
€150.00
€100.00
€50.00

June  July  August  September  October  November  December

€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

Avg. Min  Avg. Max
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
Ordentliche Studierende mit einer Staatsbürgerschaft aus EU/EWR/CH (+ Gleichgestellte), die ein Bachelor-/Diplom-/Master- oder PhD-Studium absolvieren, müssen zu Beginn jedes Semesters nur den OH-Beitrag entrichten (€ 20,20).

Die Bezahlung erfolgt innerhalb der allgemeinen Zulassungsfrist an einem Service-Point oder online und berechtigt zur Fortsetzung des Studiums. Der Studienbeitrag ist nur zu bezahlen, wenn die Studiendauer + Toleranz überschritten wird, aufgrund der Staatszugehörigkeit und beim außerordentlichen Studium. Ausnahme bilden Studien, die in Kooperation mit der UMIT angeboten werden und das
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)
References


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

https://doi.org/10.1007/b116303