

Knowledge Graphs

Methodology, Tools and Selected Use Cases

Chapter1-2.3

2021.01.14

Ch 1. What Is a Knowledge Graph?

- Introduction
 - From smart speakers such as Alexa and Google Home introduced AI-based (AI) communication to Schemed web.
 - Data, content, and services become semantically annotated, allowing a software agent, to search through the web understanding its content.
 - For giving a meaningful answer, an agent needs knowledge.
 - Therefore, Google started in 2012 to develop a so-called Knowledge Graph.

A Conceptual Definition Of Knowledge Graphs

- Knowledge Graph
 - An agent has/generates knowledge by interpreting a graph, i.e., relates its elements to so-called real-world objects and actions.
- Knowledge Base vs. Knowledge Graph
 - KB: containing the **knowledge** and an **inference engine** that can be used to derive new facts or answering questions.
 - Rigidly defined scheme can be used.
 - KG: Logical formulas are missing, and the terminological knowledge is hosted at the same layer as the assertions.
 - There is little to reason with a KG.

An Empirical Definition Of Knowledge Graphs

- Open Knowledge Graphs
 - DBpedia: It is the de facto central dataset on the Semantic Web since it is linked to many other datasets.
 - Freebase: Collaborative knowledge base. Shut down and included in Wikidata.
 - YAGO: Fuses entities extracted from Wikipedia articles with WordNet synsets to enrich the type hierarchy. only extracts a handful number of relations and focuses on keeping the Knowledge Graph compact but highly accurate and consistent.
 - NELL: NELL agents run continuously and keep improving the knowledge base by creating new facts and delete obsolete and wrong ones.
 - Wikidata: Based on the knowledge of Wikipedia but can be also edited by community members.
 - KBpedia: a knowledge base that contains mappings to Wikipedia, Wikidata, schema.org, DBpedia, GeoNames, OpenCyc, and UMBEL.
 - datacommons.org: About geographic and administrational areas, demographics, and other publicly available data such as weather and real estate.

An Empirical Definition Of Knowledge Graphs

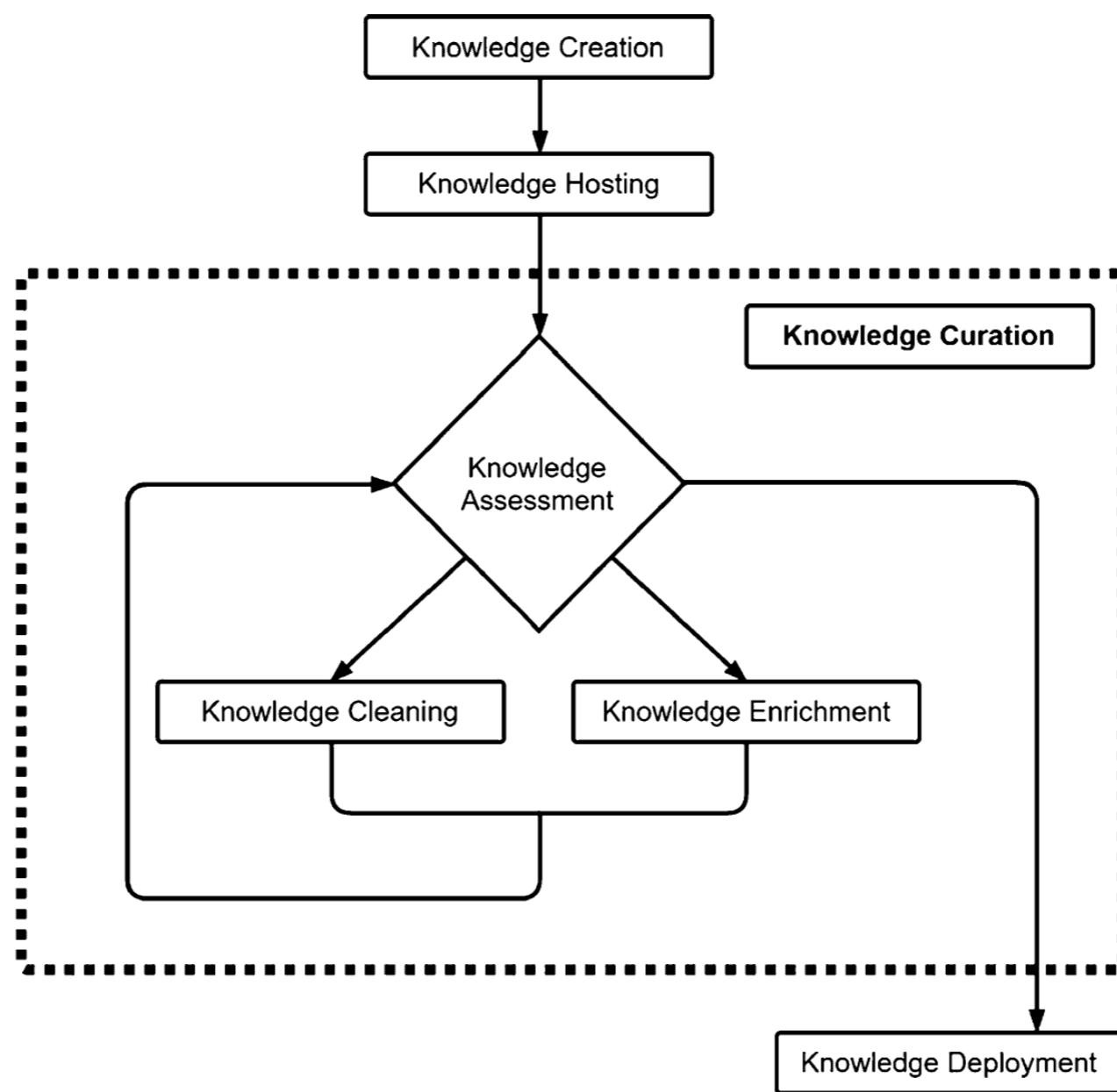
- Proprietary Knowledge Graphs
 - Cyc: Common-sense knowledge base. It adopts tools and methods to retrieve knowledge from external sources when necessary.
 - Facebook's Entities Graph: About the Facebook users, namely, their profile information, interests, and connections.
 - Google's KG: To improve Google's search engine results, effectively converting Google to a question-answering engine.
 - Yahoo's KG: Acquires data from heterogenous sources and fuses them under a common OWL Ontology.
 - Knowledge Vault: Aims to create a large probabilistic knowledge base extracted from different kinds of web content and data.

Ch 2. How to Build a Knowledge Graph

- Major steps of an overall process model:
 - Knowledge creation
 - Knowledge hosting
 - Knowledge curation
 - Knowledge deployment

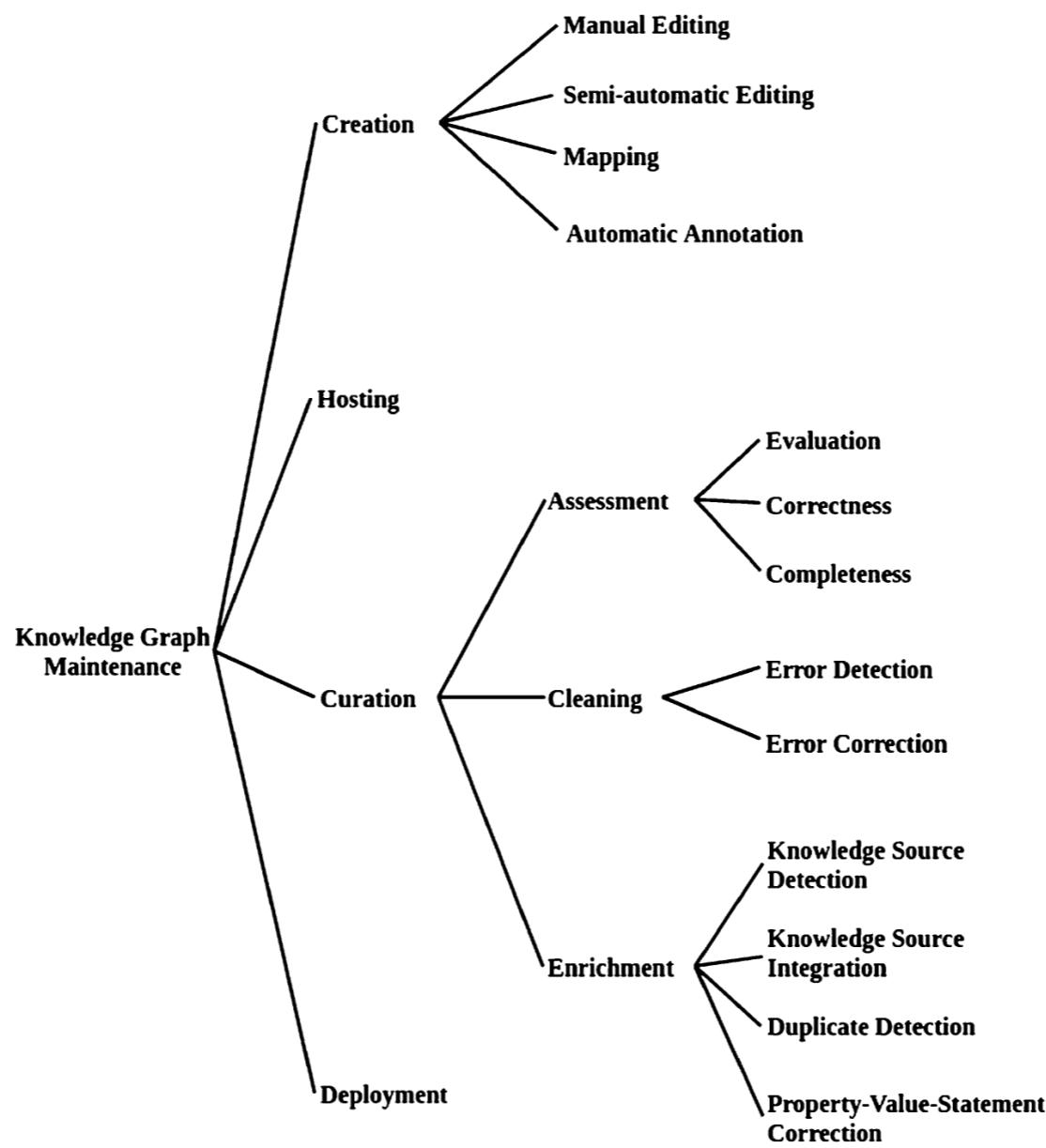
Introduction

- Process model for KG generation



Introduction

- Process model for KG generation



2.2 Knowledge Creation

- Knowledge acquisition describes the process of extracting information from different sources.

Knowledge Creation Methodology

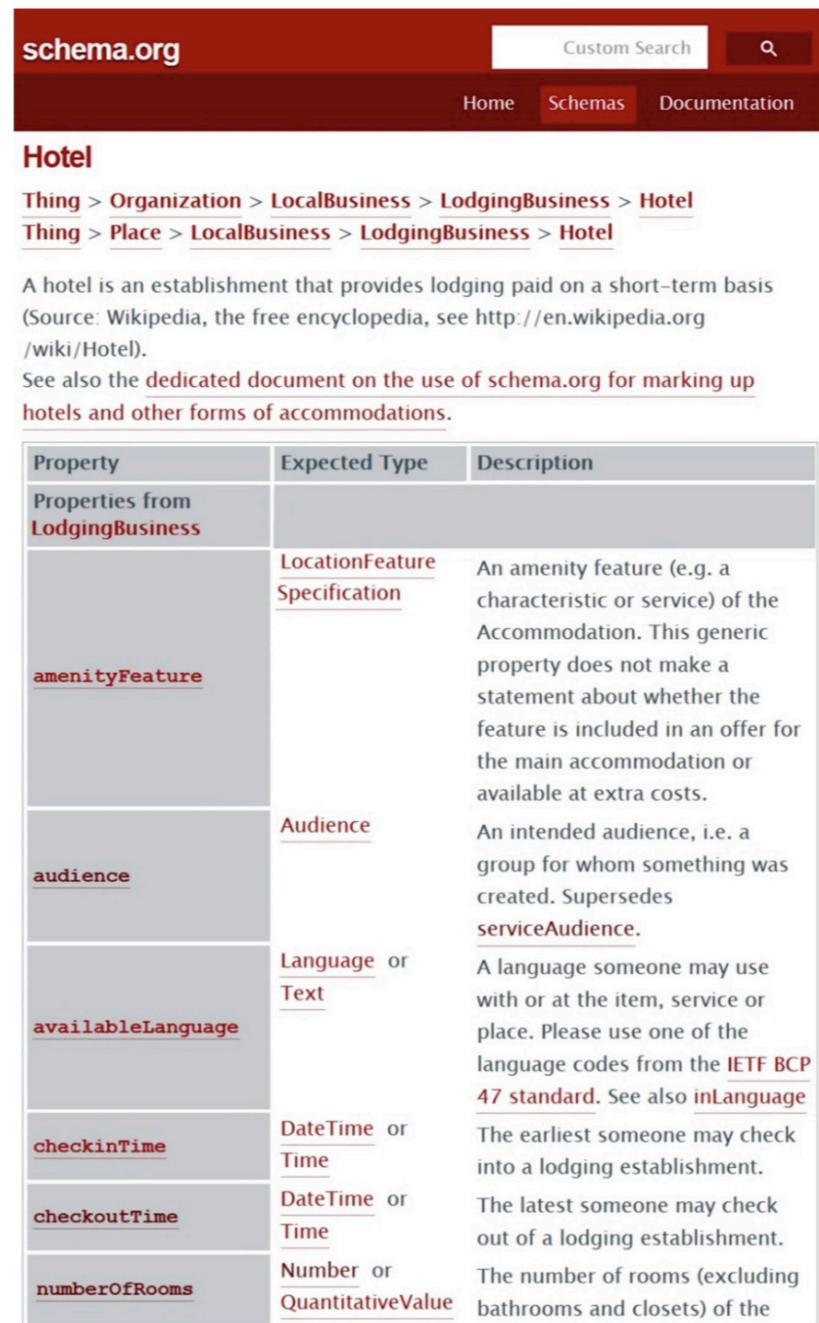
- **Bottom-up part:** describes the steps of a first annotation process
 1. Analysis of domain entities and their online representation
 2. Definition of a vocabulary based on restricting and extending semantic vocabularies
 3. Domain specifications and mapping to the semantic vocabularies
 4. Annotation development and deployment
 5. Evaluation and analysis of the annotations
- **Domain specification modeling:** reflects these results
- **Top-down part:** applies the constructed models to further knowledge acquisition
 1. Mapping according to domain specifications
 2. Annotation development according to domain specifications

Our Modeling Language

- Schema.org
 - A collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond.
 - The provided corpus of types, properties, range definitions, and enumeration values cover a large number of different domains.
 - It has become a de facto standard for annotating data.

Our Modeling Language

- Properties and ranges for annotating a hotel with the type *Hotel* from [schema.org](#)



The screenshot shows the schema.org page for the **Hotel** type. It includes the navigation bar with links to Home, Schemas, and Documentation, and a search bar. Below the navigation, the **Hotel** type is defined as a subclass of LocalBusiness and LodgingBusiness. A detailed description of a hotel is provided, mentioning Wikipedia as the source. A note encourages users to refer to a dedicated document for marking up hotels and other accommodations. A table lists various properties and their expected types and descriptions.

Property	Expected Type	Description
Properties from LodgingBusiness		
amenityFeature	LocationFeature Specification	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

Knowledge Generation Tools

- Annotation process
 - In order to support the semantic annotation process, tools are required:
 - Manual and semi-automatic editing,
 - Mappings of external schemas,
 - Automatic annotation,
 - Evaluation

Knowledge Generation Tools

- Hotel annotation in JSON-LD format

Ferienhof Hotel Garni Oblasser ×

```
{
  "@context": "http://schema.org",
  "@type": [
    "Hotel"
  ],
  "url": [
    "http://www.ferienhof-oblasser.at",
    "http://maps.mayrhofen.at/?foreignResource=C5EC2DF7-5E30-42AB-9D58-5607FB343ECA"
  ],
  "address": {
    "@type": "PostalAddress",
    "name": "Oblasser, Ferienhof",
    "streetAddress": "Hochstegen 835",
    "addressLocality": "Mayrhofen",
    "postalCode": "6290",
    "addressCountry": "AT",
    "telephone": "0043 5285 64666",
    "faxNumber": "0043 5285 64559",
    "email": "info@ferienhof-oblasser.at",
    "url": "http://www.ferienhof-oblasser.at"
  },
  "name": "Ferienhof Hotel Garni Oblasser",
  "description": [
    "Children up to the age of 6 go free, up to the age of 12 a 50% price reduction.\nIncludes: Heating, garage, water, electricity, local taxes and bed linen.\nExcludes: final cleaning charge",
    "The hotel offers a range of services including heating, garage access, water, electricity, and local taxes. Children under 6 years old stay free, and those aged 12 and above receive a 50% discount. Excluded from the package are final cleaning charges."
  ]
}
```

Knowledge Generation Tools

1. Manual Editing

- Manually annotated process via semantify.it Annotation Editor.
- User selects a domain specification, and fills in.

Knowledge Generation Tools

1. Manual Editing

- semantify.it Annotation Editor

Annotate Restaurant

<input type="text"/> i name	AKROPOLIS
<input type="text"/> i description	Wir heißen Sie herzlich im Restaurant AKROPOLIS willkommen!
<input type="text"/> i url	https://akropolis-innsbruck.com/
<input type="text"/> i telephone	+43 512 57 57 61
<input type="text"/> i email	contact@akropolis-innsbruck.com
<input type="text"/> i address	
<input type="text"/> i addressCountry	
<input type="text"/> i name	Österreich
<input type="text"/> i addressLocality	Innsbruck
<input type="text"/> i addressRegion	Tirol
<input type="text"/> i postalCode	6020
<input type="text"/> i streetAddress	Inrain 13
<input type="text"/> i aggregateRating	
<input type="text"/> i bestRating	5
<input type="text"/> i ratingCount	1
<input type="text"/> i ratingValue	5
<input type="text"/> i acceptsReservations	URL v https://bda.bookatable.com/

Knowledge Generation Tools

2. Semi-automatic Editing

- Users fill in the fields in the editor by extracting information from the given URI or source file.
- The information can be extracted from the source web page by tracking the appropriate HTML tags. (Title, bold, lists, ...)

Knowledge Generation Tools

3. Mapping

- The data are often provided by different institutions and might be in different formats and using different conceptual structures.
- To make this data accessible in a Knowledge Graph,
 - need to **transfer** it into the format and schema of our knowledge representation formalism. → **Data lifting**
- Tools
 - XLWrap, SLIPO, Mapping Master, XMLtoRDF, ...
 - Logical Source: determines the input source that contains the data to be mapped.

Knowledge Generation Tools

3. Mapping

- RML mapping for type *Person* and type *PostalAddress*

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix schema: <http://schema.org/> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .
@base <http://sti2.at/> .

<#LOGICALSOURCE>
rml:source ".../exampleJSONtoSDO/input.json";
rml:referenceFormulation ql:JSONPath;
rml:baseSource <#Mapping>;
rml:iterator "$.*".
<#Mapping>
rml:logicalSource <#LOGICALSOURCE>;
rr:subjectMap [
  rr:termType rr:BlankNode;
  rr:class schema:Person;
];
rr:predicateObjectMap [
  rr:predicate schema:name;
  rr:objectMap [ rml:reference "name" ];
];
rr:predicateObjectMap [
  rr:predicate schema:age;
  rr:objectMap [ rml:reference "age" ];
];
rr:predicateObjectMap [
  rr:predicate schema:address;
  rr:objectMap [
    rr:parentTriplesMap <#ADDRESSmapping>;
  ];
].
```

```
rr:predicateObjectMap [
  rr:predicate schema:addressCountry;
  rr:objectMap [ rml:reference "livesIn.country" ];
];
rr:predicateObjectMap [
  rr:predicate schema:addressLocality;
  rr:objectMap [ rml:reference "livesIn.city" ];
].
```

Knowledge Generation Tools

3. Mapping

- Example of input (JSON) and output (JSON-LD)

```
[  
  { "name": "Tom A.",  
    "age": 15,  
    "livesIn": {  
      "country": "Austria",  
      "city": "Innsbruck" }  
  },  
  { "name": "Ralph S.",  
    "age": 25,  
    "livesIn": {  
      "country": "Austria",  
      "city": "Vienna" }  
  },  
  { "name": "Anngelika B.",  
    "age": 77,  
    "livesIn": {  
      "country": "Germany",  
      "city": "Munich" }  
  }  
]
```

```
[  
  { "@type": "http://schema.org/Person",  
    "http://schema.org/name": "Tom A.",  
    "http://schema.org/age": 15,  
    "http://schema.org/address": {  
      "@type": "http://schema.org/PostalAddress",  
      "http://schema.org/addressCountry": "Austria",  
      "http://schema.org/addressLocality": "Innsbruck" }  
  },  
  { "@type": "http://schema.org/Person",  
    "http://schema.org/name": "Ralph S.",  
    "http://schema.org/age": 25,  
    "http://schema.org/address": {  
      "@type": "http://schema.org/PostalAddress",  
      "http://schema.org/addressCountry": "Austria",  
      "http://schema.org/addressLocality": "Vienna" }  
  },  
  { "@type": "http://schema.org/Person",  
    "http://schema.org/name": "Anngelika B.",  
    "http://schema.org/age": 77,  
    "http://schema.org/address": {  
      "@type": "http://schema.org/PostalAddress",  
      "http://schema.org/addressCountry": "Germany",  
      "http://schema.org/addressLocality": "Munich" }  
  }  
]
```

Knowledge Generation Tools

4. Automatic Annotation Tools

- Extract data from the web using natural language processing and machine learning.
- The typical tasks of NLP:
 - Tokenization, Stemming, Lemmatization, NER, POS tagging, Chunking, Syntactic parsing, Relation extraction, Semantic role labeling, Co-reference resolution.
- Tools
 - GATE, OpenNLP, RapidMiner, ...
- Automated annotation comes with a **high training effort** to adapt generic methods for specific domain and tasks. Only then a certain level of quality can be achieved.

Knowledge Generation Tools

5. Evaluation

- Quality assessment methodology to evaluate the results of the knowledge engineering process.
- Validation
 - Compare the semantic annotations with the web resource they annotate.
- Verification
 - Evaluate semantic annotations against a formal specification.
- Errors may be caused by:
 - Not following guidelines, technical or human errors, ...

Knowledge Generation Tools

5. Evaluation

- semantify.it Verifier

2. Verification Report

✓ Verification Result

Nr.	Type	Markup	View	Schema.org Verification	Domain-specific Verification
1	Event	jsonld		Conform with Warnings	Not Conform
2	BreadcrumbList	microdata		Conform	No DS
3	PostalAddress	microdata		Conform with Warnings	No DS

▼ Input

<https://www.mayrhofen.at/events/detail/event/7-zillertaler-weisenblaesertreffen/>

`<title>7. Zillertaler Weisenbläsertreffen : TVB Mayrhofen</title>`
`<meta name="description" content=<p class="MsoNormal" style="margin: 0cm 0cm 12pt;">Teilnehmer: Ebbser Kaiserklang, Zillertaler Weisenbläser, die Gfirmtn, Klarinetten-Ensemble Zillertal>`
`<meta property="og:type" content="website">`
`<meta property="og:title" content="7. Zillertaler Weisenbläsertreffen : TVB`

Knowledge Generation Tools

5. Evaluation

- semantify.it Verification Report

Domain Specification Verification Report

▼ Errors (9):

- ✖ **Missing Property:** The annotation is missing a property ('schema:height') that is defined as required by the domain specification.
↳ Event.image > 0
- ✖ **Missing Property:** The annotation is missing a property ('schema:name') that is defined as required by the domain specification.
↳ Event.image > 0
- ✖ **Missing Property:** The annotation is missing a property ('schema:width') that is defined as required by the domain specification.

▼ Warnings (8):

- ⚠ **Non-conform property:** The annotation has a property ('schema:caption') that is not specified by the domain specification.
↳ Event.image > 0
- ⚠ **Non-conform property:** The annotation has a property ('schema:caption') that is not specified by the domain specification.
↳ Event.image > 1
- ⚠ **Non-conform property:** The annotation has a property ('schema:email') that is not specified by the domain specification.

▼ Annotation

```
{  
  "@context": "http://schema.org",  
  "@type": "Event",  
  "name": "7. Zillertaler Weisenbläsertreffen",  
  "eventStatus": "EventScheduled",  
  "organizer": {  
    "@type": "Organization"  
  }  
}
```

▼ Domain Specification



[View Domain Specification](#)

2.3 Knowledge Hosting

- Methods to populate, store, and retrieve data either from **semantic annotations** of web sites or directly from a **Knowledge Graph**.

2.3 Knowledge Hosting

1. Collection, Storage, and Retrieval of Semantic Annotations

- Collection: Manual or semiautomatic editing, automatic annotation generation, or through mappings.
- Storage: MongoDB - JSON based, lightweight and cost-effective solution for storing.
- Retrieval: Annotations stored in `semantify.it` can be accessed directly through a shortened URL.

2.3 Knowledge Hosting

1. Collection, Storage, and Retrieval of Semantic Annotations

- The result of a query to [semantify.it](#) is a JSON-LD file

GET https://smtfy.it/BJgn06IHNb	{ "@context": "http://schema.org", "@type": "LodgingBusiness", "name": "Haus Olmarausch", "disambiguatingDescription": "Unser Haus liegt in schöner, sonniger Lage inmitten von Leutasch. Wir bieten ein gut ausgestattetes heimeliges Haus und herzliche Gastfreundschaft. Wir wollen vor allem eines: Dass Sie sich von Anfang an wie zu Hause führen. \nDer Loipeneinstieg und befestigte Winterwanderwage sind direkt vis a vis vom Haus. \nIm Sommer Ausgangspunkt für herrliche Wanderungen und Radtouren auf schönen und sicheren Wander - und Radwegen in den Bergen von Leutasch. Das Ortszentrum, Gasthöfe und Bäckerei sind in kurzer Zeit erreichbar.", "@description": "<p>Unser Haus liegt in schöner, sonniger Lage inmitten von Leutasch. Wir bieten ein gut ausgestattetes heimeliges Haus und herzliche Gastfreundschaft. Wir wollen vor allem eines: Dass Sie sich von Anfang an wie zu Hause fühlen. Der Loipeneinstieg und befestigte Winterwanderwage sind direkt vis a vis vom Haus. Im Sommer Ausgangspunkt für herrliche Wanderungen und Radtouren auf schönen und sicheren Wander - und Radwegen in den Bergen von Leutasch. Das Ortszentrum, Gasthöfe und Bäckerei sind in kurzer Zeit erreichbar."}
--	---

2.3 Knowledge Hosting

2. Collection, Storage, and Retrieval of Knowledge Graphs

- Collection: Crawling or Mapping
- Storage: RDF, XML, CSV, etc. → GraphDB
- Retrieval: SPARQL API of GraphDB - SQL for graph database.

```
PREFIX schema: <http://schema.org/>
SELECT DISTINCT ?name ?url WHERE {
    ?s a schema:SkiResort;
        schema:name ?name;
        OPTIONAL {?s schema:url ?url}
}
LIMIT 100
```

Fin.