universität innsbruck

Department of Computer Science

STI INNSBRUCK

SE Research Seminar: Knowledge Graphs

Final Presentation

By Penz Manuel, Rasmusen Sven

Agenda

- Domain Selection
- Metrics & Dimensions definition
- Intermediate External Source Conclusion
- Web Scraper
- Mapping
- Assessment
- Data source comparison & final data source conclusion
- Duplicate detection
- Error detection

Domain Selection

- 259 hotel instances

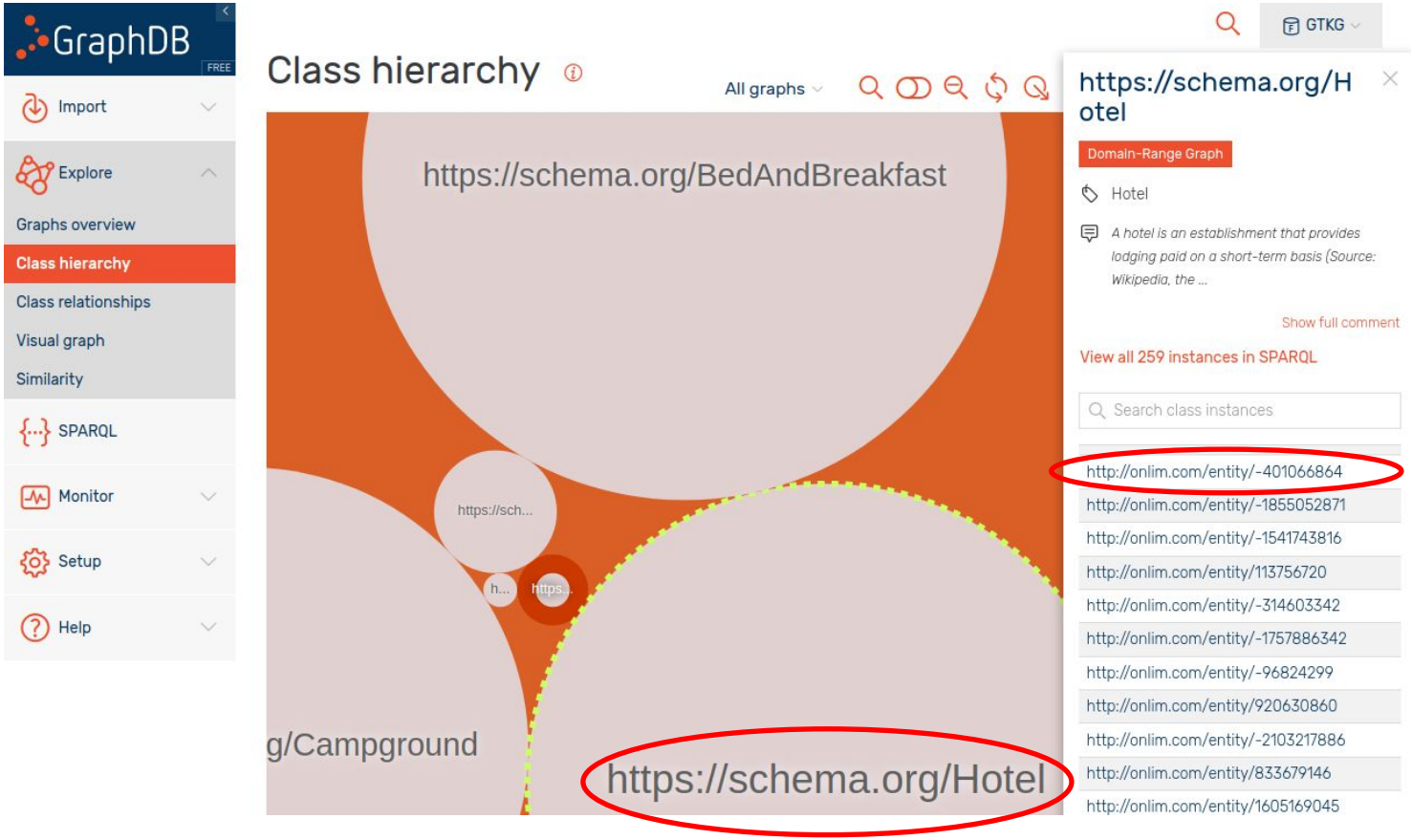
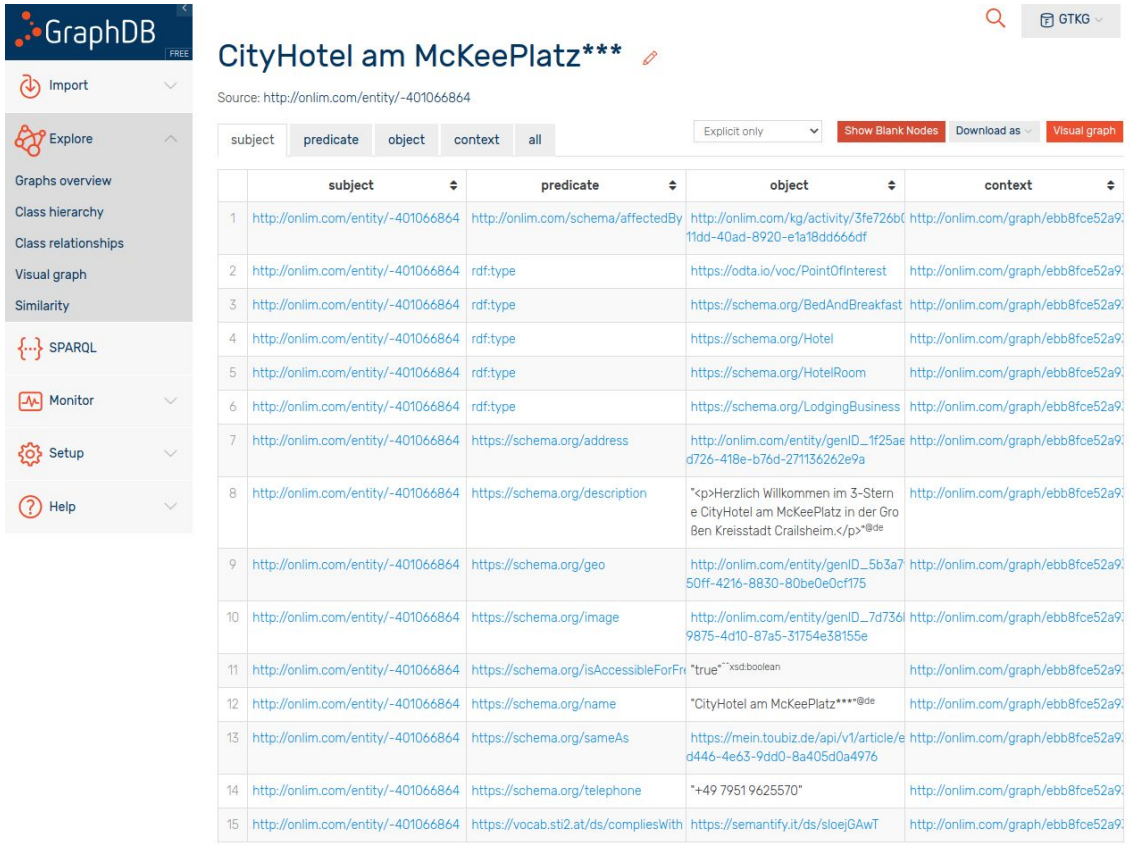


Figure 1: The class hierarchy visualisation of GTKG displaying the existing hotel instances.

Domain Selection

- Properties:
  - address
  - description
  - geo
  - image
  - isAccessibleForFree
  - name
  - sameAs
  - telephone
  - compliesWith
- Missing Properties:
  - review
  - rating
  - owner



	subject	predicate	object	context
1	http://onlim.com/entity/-401066864	http://onlim.com/schema/affectedBy	http://onlim.com/kg/activity/3fe726b0 11dd-40ad-8920-e1a18dd666df	http://onlim.com/graph/ebb8fce52a9
2	http://onlim.com/entity/-401066864	rdf:type	https://odta.io/voc/PointOfInterest	http://onlim.com/graph/ebb8fce52a9
3	http://onlim.com/entity/-401066864	rdf:type	https://schema.org/BedAndBreakfast	http://onlim.com/graph/ebb8fce52a9
4	http://onlim.com/entity/-401066864	rdf:type	https://schema.org/Hotel	http://onlim.com/graph/ebb8fce52a9
5	http://onlim.com/entity/-401066864	rdf:type	https://schema.org/HotelRoom	http://onlim.com/graph/ebb8fce52a9
6	http://onlim.com/entity/-401066864	rdf:type	https://schema.org/LodgingBusiness	http://onlim.com/graph/ebb8fce52a9
7	http://onlim.com/entity/-401066864	https://schema.org/address	http://onlim.com/entity/genID_1f25ae d726-418e-b76d-271136262e9a	http://onlim.com/graph/ebb8fce52a9
8	http://onlim.com/entity/-401066864	https://schema.org/description	" Herzlich Willkommen im 3-Stern e CityHotel am McKeePlatz in der Gro Ben Kreisstadt Crailsheim. "@de	http://onlim.com/graph/ebb8fce52a9
9	http://onlim.com/entity/-401066864	https://schema.org/geo	http://onlim.com/entity/genID_5b3a7 50ff-4216-8830-80be0e0cf175	http://onlim.com/graph/ebb8fce52a9
10	http://onlim.com/entity/-401066864	https://schema.org/image	http://onlim.com/entity/genID_7d736 9875-4d10-87a5-31754e38155e	http://onlim.com/graph/ebb8fce52a9
11	http://onlim.com/entity/-401066864	https://schema.org/isAccessibleForFr	"true"^^xsd:boolean	http://onlim.com/graph/ebb8fce52a9
12	http://onlim.com/entity/-401066864	https://schema.org/name	"CityHotel am McKeePlatz***"@de	http://onlim.com/graph/ebb8fce52a9
13	http://onlim.com/entity/-401066864	https://schema.org/sameAs	https://mein.toubiz.de/api/v1/article/e d446-4e63-9dd0-8a405d0a4976	http://onlim.com/graph/ebb8fce52a9
14	http://onlim.com/entity/-401066864	https://schema.org/telephone	"+49 7951 9625570"	http://onlim.com/graph/ebb8fce52a9
15	http://onlim.com/entity/-401066864	https://vocab.sti2.at/ds/compliesWith	https://semantify.it/ds/sloejGAwT	http://onlim.com/graph/ebb8fce52a9

Figure 2: A detailed view of a specific hotel instance within the GTKG.

Dimensions & Metrics

- **Accessibility**
  - **Provisioning of public endpoint**
    - Weight = 0.45
    - 1 If SPARQL and REST API
    - 0.75 either SPARQL or REST API
    - 0.5 any form of offline data (e.g. csv)
    - 0 Otherwise
  - **Retrievable format**
    - Weight = 0.45
    - 1 If RDF export available
    - 0.75 If JSON export available
    - 0.5 If semi-structured data available
    - 0 Otherwise
  - **Content negotiation**
    - Weight = 0.1
    - 1 If content negotiation is supported
    - 0 Otherwise

Dimensions & Metrics

- **Completeness**
  - Instance completeness
    - Weight = 0.5
    - $$m = \frac{1}{N} \Sigma \frac{number\ of\ values\ from\ classes\ \&\ properties\ in\ instance\ in\ subset}{number\ of\ total\ values\ from\ \ classes\ \&\ properties\ according\ to\ DS}; \ N \ldots \ subset\ size$$
  - Population completeness
    - Weight = 0.5
    - $$m = \frac{number\ of\ objects\ per\ domain\ represented\ in\ the\ data\ source}{total\ number\ objects\ per\ domain}$$

Dimensions & Metrics

- **Accuracy**
  - Formal Syntactic Validity
    - Weight = 0.5
    - $$m = \frac{|\{ o \mid (s,p,o) \in r \wedge o \in L \wedge synValid(o) \}|}{|\{ o \mid (s,p,o) \in r \wedge o \in L \}|}$$
  - *synValid()* rule examples:
    - Postal Code:
      - length of 5
      - starting from 01 to 99
    - Phone number:
      - start with +49 or 0049 followed by a valid area code
        - starting from 02 to 09
        - total length between 3 and 5

Dimensions & Metrics

- **Accuracy**
  - Formal Semantic Validity
    - Weight = 0.5
    - $$m = \frac{|\{ o \mid (s,p,o) \in r \wedge o \in L \wedge semValid(o) \}|}{|\{ o \mid (s,p,o) \in r \wedge o \in L \}|}$$
  - *semValid()* rule examples:
    - website:
      - reachable or not?
    - phone:
      - does the number belong to the correct hotel?

Conclusion to External Sources

- www.wikidata.org:
  - SPARQL endpoint available!

- www.firmenregister.de
  - No endpoint available!
  - Solution:
    - Scrape website

Data source: www.firmenregister.de

Building a web scraper, schema alignment, mapping, and assessment.

Scraper

- Python
  - BeautifulSoup
  - Proxy Server
  - Scraper procedure
    1) Grab the URL of every page listing lodging businesses
    2) Grab the URL of every lodging business on each page
    3) Grab data from a table inside each lodging business' page

Scraper

- **Exports data of almost 9k german lodging businesses into JSON file**


```
….
        },
        {
                "Firmenname": "Hotel Find GmbH",
                "Adresse": "Hauptstätter Str. 53B",
                "PLZ": "70178",
                "Ort": "Stuttgart",
                "Bundesland": "Baden-Württemberg",
                "Telefon": "+49 711 6404076",
                "Fax": "+49 711 6409417",
                "E-Mail": "info@hotel-find.de",
                "Homepage": "http://www.hotel-find.de",
                "Kontakt": "Herr Culum"
        },
        {
….
```


Figure 3: firmenregister.json generated from scraper

Mapping - firmenregister.de

- Properties:
  - name - “Firmenname”
  - telephone - “Telefon”
  - faxNumber - “Fax”
  - email - “E-Mail”
  - url - “Homepage”
  - description - “Produkte/Infos”
  - #AddressMapping_JSON -> PostalAddress
    - streetAddress - “Adresse”
    - addressLocality - “Ort”
    - postalCode - “PLZ”
    - addressRegion - “Bundesland”
  - #ContactMapping_JSON -> ContactPoint
    - name - “Kontakt”

```
…..

<#LOGICALSOURCE>
rml:source "firmenregister.json";
rml:referenceFormulation ql:JSONPath;
rml:iterator "$.[*]".

<#LodgingBusinessMapping>
rml:logicalSource <#LOGICALSOURCE>;

rr:subjectMap [
   rr:template
"https://lodgingbusiness.example.com/{Firmenname}";
   rr:class schema:Hotel;
];

rr:predicateObjectMap [
   rr:predicate schema:name;
   rr:objectMap [
      rml:reference "Firmenname"
   ];
];

…..
```

Figure 4: Mapping file for firmenregister.de

Assessment - firmenregister.de

Accessibility	0,40	Provisioning of public endpoint	0,45	0 otherwise	0,00	0,23
		Retrievable format	0,45	0,5 HTML semi structured	0,50	
		Content negotiation	0,10	0 no format given	0,00	
Completeness	0,30	Instance completeness	0,50	175 out of 895 are complete	0,20	0,60
		Population completeness	0,50	more data than in GTKG	1,00	
Accuracy	0,30	Formal syntactic validity	0,50	551 out of 895 are valid	0,62	0,63
		Formal semantic validity	0,50	895 out of 895 are programmatically valid 3 out of 10 are valid after manual checking	0,65	

overall assessment score	0,46

Figure 5: The assessment results for www.firmenregister.de after programmatic and manual assessment

Data source: www.wikidata.org

SPARQL query, schema alignment, mapping, and assessment.

SPARQL - Query

- **Focus on mandatory data first**

```
SELECT ?hotelLabel ?countryLabel ?email_address ?phone_number ?street_address ?postal_
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
  ?hotel wdt:P31 wd:Q27686;
    wdt:P17 wd:Q183.
  OPTIONAL { ?hotel wdt:P17 ?country. }
  OPTIONAL { ?hotel wdt:P968 ?email_address. }
  OPTIONAL { ?hotel wdt:P1329 ?phone_number. }
  OPTIONAL { ?hotel wdt:P6375 ?street_address. }
  OPTIONAL { ?hotel wdt:P281 ?postal_code. }
  OPTIONAL { ?hotel wdt:P18 ?image. }
  OPTIONAL { ?hotel wdt:P856 ?official_website. }
  OPTIONAL { ?hotel wdt:P10290 ?hotel_rating. }
  OPTIONAL {
  ?hotel p:P625 ?coordinate_location.
  ?coordinate_location psv:P625 ?coordinate_node .
  ?coordinate_node wikibase:geoLatitude ?lat .
  ?coordinate_node wikibase:geoLongitude ?lon .}
  OPTIONAL { ?hotel wdt:P281 ?postal_code. }
  OPTIONAL { ?hotel wdt:P571 ?inception. }
  OPTIONAL { ?hotel wdt:P127 ?owned_by. }
  OPTIONAL { ?hotel wdt:P8746 ?check_out_time. }
  OPTIONAL { ?hotel wdt:P8745 ?check_in_time. }
  OPTIONAL { ?hotel wdt:P276 ?location. }
}
```

Figure 6: The SPARQL query used on www.wikidata.org .

SPARQL - Query Result

- **Focus on mandatory data first**
- **3273 instances from wikidata**


```
….
        },
        {
                "hotelLabel": "Hilton Munich Park",
                "countryLabel": "Germany",
                "email_address": "mailto:info.munich@hilton.com",
                "phone_number": "+49-89-38450",
                "street_address": "Am Tucherpark 7",
                "postal_code": "80538",
                "official_website":
"https://www.hilton.com/en/hotels/muchitw-hilton-munich-park/",
                "lat": "48.152449",
                "lon": "11.598353",
                "inception": "1972-07-01T00:00:00Z",
                "owned_byLabel": "Hilton Worldwide"
        },
        {
….
```


Figure 7: The JSON data received after using the SPARQL query on www.wikidata.org .

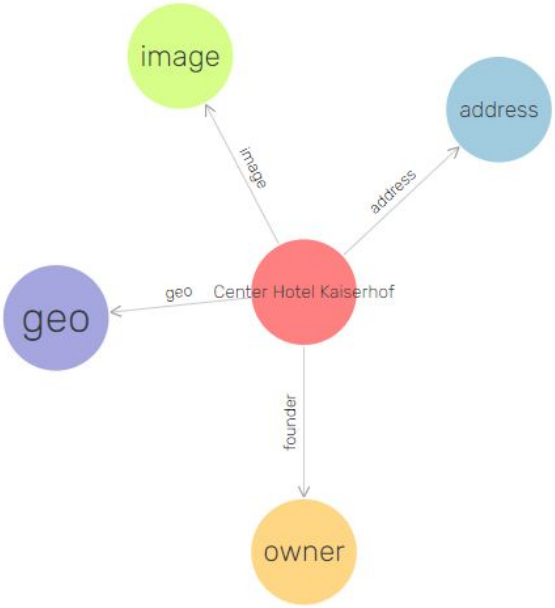
Mapping - wikidata.org

- Properties:
  - name - “hotelLabel”
  - telephone - “phone_number”
  - faxNumber - “fax_number”
  - email - “email_address”
  - url - “official_website”
  - foundingDate- “inception”
  - checkinTime- “check_in_timeLabel”
  - checkoutTime- “check_out_timeLabel”
  - #Address -> PostalAddress
  - #GeoCoords -> GeoCoordinates
  - #Image -> ImageObject
  - #Owner -> Person

```
…..

<#Mapping>
 rml:logicalSource <#LOGICALSOURCE>;

  rr:subjectMap [
     rr:template "https://schema.org/Hotel/{hotelLabel}";
     rr:class schema:Hotel;
  ];

 rr:predicateObjectMap [
     rr:predicate schema:name;
     rr:objectMap [ rml:reference "hotelLabel" ];
 ];

   rr:predicateObjectMap [
     rr:predicate schema:email;
     rr:objectMap [ rml:reference "email_address" ];
 ];

 rr:predicateObjectMap [
     rr:predicate schema:telephone;
     rr:objectMap [ rml:reference "phone_number" ];
 ];

…..
```

Figure 8: A code snippet used for mapping a hotel.

Importing contd. - wikidata.org



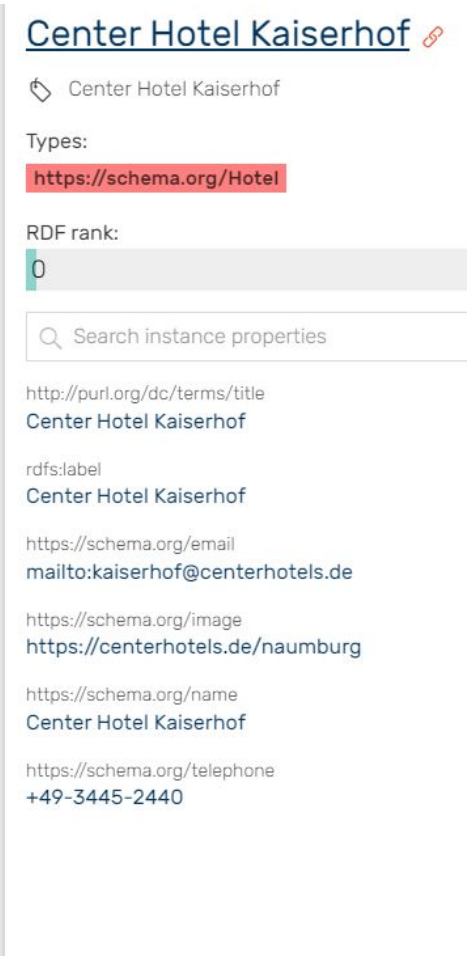


Figure 9: A detailed view of a newly inserted hotel instance.

Assessment - wikidata.org

Accessibility	0.40	Provisioning of public endpoint	0.45	0.75 SPARQL or REST API	0.75	0.78
		Retrievable format	0.45	0,75 JSON export	0.75	
		Content negotiation	0.10	1 content negotiation	1.00	
Completeness	0.30	Instance completeness	0.50	13 out of 358 are complete	0.04	0.52
		Population completeness	0.50	more data than in GTKG	1.00	
Accuracy	0.30	Formal syntactic validity	0.50	331 out of 358 are valid	0.92	0.88
		Formal semantic validity	0.50	347 out of 358 are valid 7 out of 10 are valid after manual checking	0.83	
		overall assessment score	0.73			

Figure 10: The assessment results for www.wikidata.org .

Assessment - Comparison

Data source: www.firmenregister.de

Accessibility	0,40	Provisioning of public endpoint	0,45	0 otherwise	0,00	0,23
		Retrievable format	0,45	0,5 HTML semi structured	0,50	
		Content negotiation	0,10	0 no format given	0,00	
Completeness	0,30	Instance completeness	0,50	175 out of 895 are complete	0,20	0,60
		Population completeness	0,50	more data than in GTKG	1,00	
Accuracy	0,30	Formal syntactic validity	0,50	551 out of 895 are valid	0,62	0,63
		Formal semantic validity	0,50	895 out of 895 are programmatically valid 3 out of 10 are valid after manual checking	0,65	

overall assessment score	0,46

Data source: www.wikidata.org

Accessibility	0.40	Provisioning of public endpoint	0.45	0.75 SPARQL or REST API	0.75	0.78
		Retrievable format	0.45	0,75 JSON export	0.75	
		Content negotiation	0.10	1 content negotiation	1.00	
Completeness	0.30	Instance completeness	0.50	13 out of 358 are complete	0.04	0.52
		Population completeness	0.50	more data than in GTKG	1.00	
Accuracy	0.30	Formal syntactic validity	0.50	331 out of 358 are valid	0.92	0.88
		Formal semantic validity	0.50	347 out of 358 are valid 7 out of 10 are valid after manual checking	0.83	

overall assessment score	0.73

Figure 11: The assessment results for both data sources (www.firmenregister.de and www.wikidata.org) .

Duplicate Detection

What tool? Why?

Duplicate Detection - Duke

- Duke[1] tool for detection
- Pros:
  - Interactive mode
  - Easy to start
- Cons:
  - Difficult Tuning
  - Time intensive

[1] https://github.com/largsa/Duke

Duplicate Detection - Properties

- Compared properties:
  - Name
  - Email
  - Postalcode
  - Phone
  - Locality
  - Url
  - Fax
- Ignored properties:
  - country
  - checkinTime
  - checkoutTime
  - image
  - foundingDate
  - latitude
  - longitude
  - address
  - founder

Duplicate Detection - Configuration Values

- **Configuration values:**
  - **Name** **- LOW = 0.1** **HIGH = 0.6**
  - **Email** **- LOW = 0.45** **HIGH = 0.85**
  - **Postalcode** **- LOW = 0.1** **HIGH = 0.6**
  - **Phone** **- LOW = 0.2** **HIGH = 0.85**
  - **Locality** **- LOW = 0.1** **HIGH = 0.6**
  - ***Url*** ***- LOW = 0.1*** ***HIGH = 0.85***
  - ***Fax*** ***- LOW = 0.1*** ***HIGH = 0.85***

Duplicate Detection

- Issues during third detection
  - 124 detections
  - 94 duplicates (~76%)
  - Many correct duplicates

```
MATCH 0.8760314977675524
ID
  'https://schema.org/Hotel/Hilton%20Garden%20Inn%20Munich%20City%20West',
  'https://schema.org/Hotel/Hampton%20by%20Hilton%20Munich%20City%20West',
NAME
  'hilton garden inn munich city west',
  'hampton by hilton munich city west',
PHONE
  '+49 892388550',
  '+49 891598500',
URL
  'https://www.hilton.com/en/hotels/mucgigi-hilton-garden-inn-munich-city-west/',
  'https://www.hilton.com/en/hotels/muchxhx-hampton-munich-city-west/',
FAX
  <null>
  '+49-89-159850100',
Correct? (Y/N) n
```

Figure 12: This detection was considered the toughest one, as it was unclear if it was a duplicate or not.

Error Detection

Procedure, Tool and Outcome

Error Detection - Procedure & Tool

- Use the domain specification as an initial guideline
- Mandatory properties first
- Optional properties second
- Check expected formats and assign **sh:patterns**
- Use already existing instances as a guideline
  - example: Onlim's inserted instances may have language tags
- Adjust to accept language tags or strings using **sh:or**
- Tool used: shacl.org/playground/

Class / Property	Range / Type	Cardinality
LodgingBusiness		
name	Text or Localized Text	1..N
telephone	Text	1
address	**PostalAddress**	1
PostalAddress		
email	Text	1

Figure 13: The given domain specification for LodgingBusiness. We used the Range / Type and Cardinality columns as a guideline.

Error Detection - Outcome

- Total Validation Reports: 20’516
  - 12,799 are sh:MinCountConstraintComponent
    - Assessment of a subset: 13 out of 358 for instance completeness
  - 6480 are owner violations
    - sh:MinCountConstraintComponent
    - sh:NodeConstraintComponent

```
…
schema:AddressShape
        a sh:NodeShape ;
        sh:closed false ;
        sh:targetClass schema:PostalAddress ;
        sh:property [
     sh:path schema:addressCountry ;
      sh:datatype xsd:string ;
        sh:minCount 0 ;
        sh:hasValue "Germany" ;
        sh:name "is in Germany" ;
   ] ;
…
```

Figure 14: A code snippet of an AddressShape in SHACL. Here it is checked if the entered country is “Germany”.

Statistics

What has changed?

Statistics

- Before:
  - 10`874`984 triples
  - 259 hotel instances
- After:
  - 10`985`034 triples
  - 3`532 hotel instances

- Changes:
  - 110`050 more triples overall (~1%)
  - 3`273 more hotels

- Our GitHub:
  - https://github.com/csar8594/GTKG

Issues

Obstacles and Lessons Learned

Importing - firmenregister.de

- **Problems:**
  - **data inside the .n3 is not sorted using pyRML -> should look like below**

<https://address.example.com/%C3%84u%C3%9Fere%20Ansbacher%20Str.%203>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <https://schema.org/PostalAddress> .
<https://address.example.com/%C3%84u%C3%9Fere%20Ansbacher%20Str.%203>
<https://schema.org/addressLocality> "Weihenzell" .
<https://address.example.com/%C3%84u%C3%9Fere%20Ansbacher%20Str.%203> <https://schema.org/addressRegion>
"Bayern"

  - **after successful import, the data is still not present inside graphDB due to this sorting problem**
  - **rocketRML does export the data sorted but only if the number of properties is low and no *rr:parentTriplesMap* are used -> heap error**
    - **<u>Solution</u>: using *joinCondition***

      and rocketRML

```
rr:predicateObjectMap [
    rr:predicate schema:PostalAddress;
    rr:objectMap [
        rr:parentTriplesMap <#AddressMapping_JSON>
        rr:joinCondition [
            rr:child "Firmenname";
            rr:parent "Firmenname";
        ]
    ];
];
```

Figure A1: The joinCondition used to fix the heap error.

Issues & Lessons Learned

- Overall:
  - Stick to vocabulary -> mapping used example.com initially
  - Check prefixes (http://schema.org ≠ https://schema.org)
- Assessment:
  - Assign less weight to accessibility
  - Use a hotel oriented data source
- Error Detection:
  - Start earlier (Possibly parallel to mapping)
    - May help mapping process
    - We have many empty Person instances from wrong mapping
- Syntactic validity vs. Semantic validity

Thank You!

We are open to questions and feedback!