Automated Knowledge Base Quality Assessment and Validation based on Evolution Analysis

Mohammad Rashid

Supervisor: Prof. Marco Torchiano
Introduction
Introduction

Knowledge Base Evolution

→ DBpedia Knowledge Base – 2008*

*Source: http://lod-cloud.net/
Knowledge Base Evolution

Knowledge Bases (KBs) evolve over time: their data instances and schema can be updated, extended, revised and refactored.

Evolution of KBs is unrestrained.
Introduction

Data Quality Life Cycle*

*Source: https://www.experfy.com/blog/automating-data-quality-remediations-through-cognitive-rpa
## Analysis Level

<table>
<thead>
<tr>
<th>Analysis Level¹</th>
<th>Detail</th>
<th>Volume</th>
<th>Stakeholder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-level</td>
<td>Fine-grained</td>
<td>Large</td>
<td>Data end-user</td>
</tr>
<tr>
<td>High-level</td>
<td>Coarse-grained</td>
<td>Small</td>
<td>Data Curator</td>
</tr>
</tbody>
</table>

Identification of quality issues due to unrestrained KB evolution

Identification of erroneous conceptualizations of resources
Introduction

Quality Issues

- **Lack of Consistency** relates to a fact being inconsistent in a KB. Inconsistency relates to the presence of unexpected properties.

DBpedia resource of type `foaf:Person`: X. Henry Goodnough

Property of `dbo:birthDate`

Unexpected property of `dbo:Infrastructure/length`

In resources of type `foaf:Person` there are 1035 distinct properties, among which 142 occur only once for DBpedia version 201604.
Introduction

Quality Issues

- **Lack of Completeness** relates to the resources or properties missing from a knowledge base. This happens when information is missing or has been removed.

DBpedia resource of type `dbo:Person/Astronauts: Abdul Ahad Mohmand`

This property is missing from DBpedia but it is present in Wikipedia.

In particular, in the release of 2016-04 there are 419 occurrences of the `dbo:Astronaut/TimeInSpace` property over 634 astronaut resources, while in the previous version they were 465 out of 650 astronauts.
Quality Issues

- **Lack of Persistency**
  relates to resources that were present in a previous KB release, but disappeared from more recent ones.

One 3cixty Nice resource of type lode:Event has as label the following: “Modéliser, piloter et valoriser les actifs des collectivités et d’un territoire grâce aux maquettes numériques: retours d’expériences et bonnes pratiques”.

In 3cixty Nice KB 2016-09-09 release there was an unexpected drop of resources of type event with respect to the previous release dated 2016-06-15.
Introduction

Problem

Identification of quality issues due to unrestrained KB evolution

Hypothesis

Dynamic features from data profiling can help to detect quality issues

Research Questions

RQ1 | How can we identify quality issues with respect to KB evolution?

RQ2 | Which quality assessment approach can be defined on top of the evolution based quality characteristics?
Introduction

Problem

Identification of erroneous conceptualizations of resources

Hypothesis

Learning models can be used for validation with data profiling information as predictive features

Research Question

| RQ3 | Which approaches can be used to validate a KB evolution based quality assessment approach? |
Overview of our approach

RQ1: KB Evolution Analysis
RQ2: Evolution based Quality Assessment and Validation Approach
RQ3: Evaluation
Evolution-based Quality Characteristics

RQ1: KB Evolution Analysis

RQ2: Evolution based Quality Assessment and Validation Approach

RQ3: Evaluation

Experimental Analysis
Evolution-based Quality Characteristics

**Evolution Analysis**

Factors

- Frequency of update
- Domain Area
- Data Acquisition
- Link between data sources
Evolution-based Quality Characteristics

Dynamic Features

- Degree of change
- Lifespan
- Update history

Features

## Evolution-based Quality Characteristics

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Characteristics</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic</td>
<td>Persistence</td>
<td>Degree of change</td>
</tr>
<tr>
<td></td>
<td>Historical Persistency</td>
<td>Lifespan</td>
</tr>
<tr>
<td>Representational</td>
<td>Consistency²</td>
<td>Update history</td>
</tr>
<tr>
<td></td>
<td>Completeness²</td>
<td></td>
</tr>
</tbody>
</table>

Basic Measure Elements

- The first measure element is the count of the instances of a class $C$:

\[ \text{Count}(C) = |\{s : \exists (s, \text{typeof}, C) \in V\}| \]

- The second measure element focuses on the frequency of the properties, within a class $C$. The frequency of a property can be defined (in the scope of class $C$) as:

\[ \text{freq}(p, C) = |\{(s, p, o) \in V : \exists (s, \text{typeof}, C) \in V\}| \]
Persistency

- The Persistency of a class C in a release i : i > 1 is defined as:

\[
\text{Persistency}_i = \begin{cases} 
1 & \text{if } \text{count}_i(C) \geq \text{count}_{i-1}(C) \\
0 & \text{if } \text{count}_i(C) < \text{count}_{i-1}(C)
\end{cases}
\]

- Persistency at the knowledge base level, i.e. when all classes are considered, can be computed as the proportion of persistent classes:

\[
\text{Persistency}_i = \frac{\sum_{j=1}^{NC} \text{Persistency}_j(C_j)}{NC}
\]

where NC is the number of classes analyzed in the KB.
The Historical Persistency measure evaluates the persistency over the history of the KB and is computed as the average of the pairwise persistency measures for all releases.

\[
H_{\text{Persistency}}(C) = \frac{\sum_{i=2}^{n} \text{Persistency}_i(C)}{n - 1}
\]
Consistency

- This measure evaluates the consistency of a property on the basis of the frequency distribution. The consistency of a property $p$ in the scope of a class $C$ is:

$$Consistency_i(p, C) = \begin{cases} 
1 & \text{if } N f_i(p, C) > T \\
0 & \text{if } N f_i(p, C) < T 
\end{cases}$$

Where $T$ is a threshold that can be either a KB dependent constant or it is defined on the basis of the count of the scope class.
Completeness

- The completeness measure uses the frequency of properties. Normalized frequency:

\[ Nf_i(p, C) = \frac{freq_i(p, C)}{\text{count}_i(C)} \]

- Completeness of a property \( p \) in the scope of a class \( C \) is:

\[ \text{Completeness}_i(p, C) = \begin{cases} 1 \text{ if } Nf_i(p, C) \geq Nf_{i-1}(p, C) \\ 0 \text{ if } Nf_i(p, C) < Nf_{i-1}(p, C) \end{cases} \]

- At the class level the completeness is the proportion of complete properties and it can be computed as:

\[ \text{Completeness}_i(C) = \frac{\sum_{k=1}^{NP_i(C)} \text{Completeness}_i(p_k, C)}{NP_i(C)} \]
Evolution-based Quality Assessment and Validation Approach

RQ1: KB Evolution Analysis

RQ2: Evolution based Quality Assessment and Validation Approach

RQ3: Evaluation
Evolution-based Quality Assessment and Validation Approach

Data Life Cycle

ISO/IEC 25024 Data Life Cycle

Data design

Data Collection

Data integration

External Data acquisition

Data processing

Presentation

Other Use

Data Store

Delete

Data Quality Assessment
Proposed Approach

Evolution-based Quality Assessment and Validation Approach

Data Collection
- Data Profiling
- KB(KB1,...,n)
- SPARQL Endpoints

Quality Assessment
- Quality Profiler
  - Statistical Profiler
  - Preprocessing
- Measure 1
- Measure 2
- Measure n

Validation
- Feature Extraction
  - Quality Measures
  - Shape Generation
  - Constraints Generation
- Manual Validation
  - Instances
  - Inspections
  - Report
  - Feature Dataset

Modeling
- Quality Problem Report
- Model Evaluation
Evolution-based Quality Assessment and Validation Approach

Data Collection

Data Collection

Data Profiling

KB(KB1...n)

SPARQL Endpoints

KB1...n

Classes (C)

Vector of KB releases (T_i)
Evolution-based Quality Assessment and Validation Approach

Quality Assessment

Quality Profiler

Statistical Profiler

Preprocessing

Measure 1

Measure 2

................

Measure n
Validation Approaches

- Feature Extraction
  - Quality Measures
  - Shape Generation
  - Constraints Generation

- Manual Validation
  - Instances
  - Inspections
  - Report
  - Feature Dataset
Evolution-based Quality Assessment and Validation Approach

Modeling and Quality Problem Report

- Measure 1
- Measure 2
- ............
- Measure \( n \)

Modeling

Model Evaluation

Quality Problem Report
Experimental Analysis

RQ1: KB Evolution Analysis
RQ2: Evolution based Quality Assessment and Validation Approach
RQ3: Evaluation
Use case: 3cixty

- Cultural and tourist information\(^1\).

- Events, places (sights and businesses), transportation facilities and social activities


Use case: DBpedia

- This knowledge base is the output of the DBpedia\textsuperscript{1} project that was initiated by researchers from the Free University of Berlin and the University of Leipzig, in collaboration with OpenLinkSoftware.

- DBpedia is roughly updated every year since the first public release in 2007.

- DBpedia is created from automatically extracted structured information contained in Wikipedia, such as infobox tables, categorization information, geo-coordinates, and external links.

### Experimental Settings

<table>
<thead>
<tr>
<th>Knowledge Bases</th>
<th>Datasets</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Classes</td>
<td>Properties</td>
<td>Releases</td>
<td></td>
</tr>
<tr>
<td>DBpedia</td>
<td>10</td>
<td>4477</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>3cixty</td>
<td>2</td>
<td>149</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
**Quantitative Analysis: Persistency & Historical persistency**

*3cixty Knowledge Base*

![Graph showing persistency and historical persistency over time.](image-url)
Experimental Analysis

Quantitative Analysis: Persistency & Historical persistency

DBpedia Knowledge Base
### Quantitative Analysis: Consistency

#### DBpedia Knowledge Base

<table>
<thead>
<tr>
<th>Class</th>
<th>Total</th>
<th>Inconsistent</th>
<th>Consistent</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo:Animal</td>
<td>162</td>
<td>123</td>
<td>39</td>
</tr>
<tr>
<td>dbo:Artist</td>
<td>429</td>
<td>329</td>
<td>100</td>
</tr>
<tr>
<td>dbo:Athlete</td>
<td>436</td>
<td>298</td>
<td>138</td>
</tr>
<tr>
<td>dbo:Film</td>
<td>450</td>
<td>298</td>
<td>152</td>
</tr>
<tr>
<td>dbo:MusicalWork</td>
<td>325</td>
<td>280</td>
<td>45</td>
</tr>
<tr>
<td>dbo:Organisation</td>
<td>1,014</td>
<td>644</td>
<td>370</td>
</tr>
<tr>
<td>dbo:Place</td>
<td>1,090</td>
<td>589</td>
<td>501</td>
</tr>
<tr>
<td>dbo:Species</td>
<td>99</td>
<td>57</td>
<td>42</td>
</tr>
<tr>
<td>dbo:Work</td>
<td>935</td>
<td>689</td>
<td>276</td>
</tr>
<tr>
<td>foaf:Person</td>
<td>381</td>
<td>158</td>
<td>223</td>
</tr>
</tbody>
</table>
Quantitative Analysis: Completeness

3cixty Knowledge Base: \textit{lode:Events}
### Quantitative Analysis: Completeness

#### DBpedia Knowledge Base

<table>
<thead>
<tr>
<th>Class</th>
<th>Properties</th>
<th>Incomplete</th>
<th>Complete</th>
<th>Complete(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo:Animal</td>
<td>170</td>
<td>50</td>
<td>120</td>
<td>70.58%</td>
</tr>
<tr>
<td>dbo:Artist</td>
<td>372</td>
<td>21</td>
<td>351</td>
<td>94.35%</td>
</tr>
<tr>
<td>dbo:Athlete</td>
<td>404</td>
<td>64</td>
<td>340</td>
<td>84.16%</td>
</tr>
<tr>
<td>dbo:Film</td>
<td>461</td>
<td>34</td>
<td>427</td>
<td>92.62%</td>
</tr>
<tr>
<td>dbo:MusicalWork</td>
<td>335</td>
<td>46</td>
<td>289</td>
<td>86.17%</td>
</tr>
<tr>
<td>dbo:Organisation</td>
<td>975</td>
<td>134</td>
<td>841</td>
<td>86.26%</td>
</tr>
<tr>
<td>dbo:Place</td>
<td>1,060</td>
<td>141</td>
<td>920</td>
<td>86.69%</td>
</tr>
<tr>
<td>dbo:Species</td>
<td>101</td>
<td>27</td>
<td>74</td>
<td>73.27%</td>
</tr>
<tr>
<td>dbo:Work</td>
<td>896</td>
<td>89</td>
<td>807</td>
<td>90.06%</td>
</tr>
<tr>
<td>foaf:Person</td>
<td>396</td>
<td>131</td>
<td>265</td>
<td>66.92%</td>
</tr>
</tbody>
</table>
Experimental Analysis

Qualitative Analysis: Manual Validation

- Precision for evaluating the effectiveness of our approach

- Precision is defined as the proportion of accurate results of a quality measure over the total results

- For a given quality measure, we define an item, either a class or a property, as:
  
  - **True positive (TP)** if according to the interpretation criteria, the item presents an issue and an actual problem was detected in the KB.
  
  - **False positive (FP)** if the interpretation identifies a possible issue but no actual problem is found.
Manual Validation: Source Inspection

DBpedia Version 2016-04

About: X. Henry Goodnough

X. Henry Goodnough, (1860–1935), engineer, was chairman of Boscobel Associates, advocate for creation of the Quabbin Reservoir project. Goodnough

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo:Infrastructure/length</td>
<td>0.652272</td>
</tr>
<tr>
<td>dbo:abstract</td>
<td>X. Henry Goodnough, (1860–1935), engineer, was chairman of Boscobel Associates, advocate for creation of the Quabbin Reservoir project. Goodnough</td>
</tr>
<tr>
<td>dbo:birthDate</td>
<td>1860-1-1</td>
</tr>
<tr>
<td>dbo:buildingStartDate</td>
<td>1933</td>
</tr>
<tr>
<td>dbo:buildingStartYear</td>
<td>1933-01-01 (xsd:date)</td>
</tr>
<tr>
<td>dbo:deathDate</td>
<td>1935-1-1</td>
</tr>
<tr>
<td>dbo:height</td>
<td>80.467200 (xsd:double)</td>
</tr>
</tbody>
</table>
### Qualitative Analysis: Manual Validation

<table>
<thead>
<tr>
<th>KB</th>
<th>Quality Characteristics</th>
<th>Level</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3cixty</td>
<td>Persistency &amp; Historical Persistency</td>
<td>Class</td>
<td>lode:Event</td>
</tr>
<tr>
<td>Nice</td>
<td>Completeness</td>
<td>Property</td>
<td>lode:Event 8 properties</td>
</tr>
<tr>
<td></td>
<td>Consistency</td>
<td>Property</td>
<td>lode:Event 10 properties</td>
</tr>
<tr>
<td></td>
<td>Persistency &amp; Historical Persistency</td>
<td>Class</td>
<td>dbo:Species and dbo:Film</td>
</tr>
<tr>
<td>DBpedia</td>
<td>Completeness</td>
<td>Property</td>
<td>foaf:Person and dbo:Place</td>
</tr>
<tr>
<td></td>
<td>Consistency</td>
<td>Property</td>
<td>foaf:Person class 158 properties and dbo:Place class 114 properties</td>
</tr>
</tbody>
</table>
## Experimental Analysis

### Qualitative Analysis: Manual Validation

<table>
<thead>
<tr>
<th>Quality Characteristics</th>
<th>3cixty</th>
<th>DBpedia</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Persistency &amp; Historical</strong></td>
<td><strong>Error in reconciled algorithm</strong></td>
<td><strong>Fixed in the current version</strong></td>
</tr>
</tbody>
</table>
| **Persistency**                 | **No real issues were found in the properties.**  
   | **Scheme remains consistent for all the KB**  
   | **releases**                               | **We found issues in the properties due to erroneous conceptualization** |
| **Completeness**                | **Error in reconciled algorithm.**  
   | **Precision 95%**                          | **Error due to erroneous conceptualization and missing resources.**  
   | **Precision 94%**                        | **Precision 94%**                       |
Drawbacks of High-level Analysis

High-level change detection at the instance level, being coarse-grained, cannot capture all possible quality issues.

A quality analysis using high-level change detection may lead to increasing the number of false positives, if the KB was deployed with design issues, such as incorrect mappings.
**Experimental Analysis**

**SHACL Shape for dbo:Person Class**

```
ex:DBpediaPersonShape
  a sh:NodeShape ;
  sh:targetClass dbo:Person
  sh:property [
    sh:path foaf:name ;
    sh:minCount 1;
    sh:datatype sh:Literal
  ] ;
  sh:property [
    sh:path dbo:birthDate ;
    sh:datatype xsd:date ;
    sh:minCount 1;
    sh:maxCount 1;
    sh:nodeKind sh:Literal
  ] ;
  sh:property [
    sh:path dbo:birthPlace ;
    sh:datatype dbo:Place;
    sh:nodeKind sh:BlankNodeOrIRI;
    sh:minCount 1;
    sh:maxCount 1;
  ] .
```

- **Target class**: `Target classes` specify which nodes in the data graph must conform to a shape.

- **Constraints components**: Determine how to validate a node.

- **Node shapes**: Declare constraints directly on a node.

- **Property shapes**: Declare constraints on the values associated with a node through a path.

- **Shape**: Contains a collection of targets and constrains components.
### Constraints Components

<table>
<thead>
<tr>
<th>Constraints Type</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardinality</td>
<td>minCount, maxCount</td>
</tr>
<tr>
<td>Types of Values</td>
<td>Node Kind</td>
</tr>
<tr>
<td>Range of Values</td>
<td>minInclusive, maxInclusive, minExclusive, maxExclusive</td>
</tr>
<tr>
<td>String Based</td>
<td>minLength, maxLength, pattern,</td>
</tr>
<tr>
<td>Property pair</td>
<td>lessThan, lessThanOrEquals, disjoint, equal</td>
</tr>
<tr>
<td>Others</td>
<td>class, datatype, in, hasValue, ignoredProperties</td>
</tr>
</tbody>
</table>

We explore cardinality constraints to identify the correct mapping of properties for a specific class.

We explore the type of values to evaluate contradictions within the data.
Cardinality Constraints

For the cardinality constraints, our goal is to generate two cardinality constraints:

**minimal cardinality**
- Restricts minimum number of triples involving the focus node and a given predicate.
- Default value: 0

**maximum cardinality**
- Restricts maximum number of triples involving the focus node and a given predicate.
- Default value: unbounded

<table>
<thead>
<tr>
<th>Cardinality</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal cardinality</td>
<td>MIN0</td>
<td>Minimum Cardinality = 0</td>
</tr>
<tr>
<td></td>
<td>MIN1+</td>
<td>Minimum Cardinality &gt; 1</td>
</tr>
<tr>
<td>Maximum cardinality</td>
<td>MAX1</td>
<td>Maximum Cardinality = 1</td>
</tr>
<tr>
<td></td>
<td>MAX1+</td>
<td>Maximum Cardinality &gt; 1</td>
</tr>
</tbody>
</table>
Experimental Analysis

Feature Extraction: Cardinality Constraints

SPARQL Query:

```sparql
select ?card (count (?s) as ?count ) where {
  select ?s (count (?o) as ?card) where {
    ?s a ?class ;
    ?p ?o
  } group by ?s
} group by ?card
order by desc(?count)
```

Cardinalities of class property dbo:Sport / dbo:union:

<table>
<thead>
<tr>
<th>Cardinality</th>
<th>Instance Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1662</td>
<td>0.84883</td>
</tr>
<tr>
<td>1</td>
<td>279</td>
<td>0.14249</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>0.00511</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.00255</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.00102</td>
</tr>
</tbody>
</table>

Cardinalities of class property dbo:Sport / dbo:union:

<table>
<thead>
<tr>
<th>MIN0</th>
<th>Maximum Cardinality = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX1+</td>
<td>Maximum Cardinality &gt; 1</td>
</tr>
</tbody>
</table>

Raw cardinalities: 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 2, 3, 0, 0, 0, 2, 1, 4, ...
## Range Constraints

For the range constraints, we want to estimate if the range of a class-property is literal or object (IRI, blank node, blank node or IRI).

<table>
<thead>
<tr>
<th>IRI</th>
<th>Blank Node</th>
<th>Literal</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Any</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td></td>
<td>BlankNodeOrIRI</td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td>IRI</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
<td>BlankNode</td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>X</td>
<td>Literal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IRIOrLiteral</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
<td>BlankNodeOrLiteral</td>
</tr>
</tbody>
</table>

---
Feature Extraction: Range Constraints

Object node type information: IRI or LIT?

<table>
<thead>
<tr>
<th>Class-property</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo:Person-dbp:birthPlace</td>
<td>89,355</td>
</tr>
<tr>
<td>dbo:Person-dbp:name</td>
<td>21,496</td>
</tr>
<tr>
<td>dbo:Person-dbp:deathDate</td>
<td>127</td>
</tr>
<tr>
<td>dbo:Person-dbp:religion</td>
<td>8,374</td>
</tr>
<tr>
<td>dbo:Person-dbp:deathDate</td>
<td>111</td>
</tr>
</tbody>
</table>

@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix dbp: <http://dbpedia.org/property/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .

```turtle
ex:DBpediaPerson a sh:NodeShape;
  sh:targetClass dbo:Person;
  # node_type IRI
  sh:property [ sh:path dbp:birthPlace;
    sh:nodeKind sh:IRI;
    sh:or ( [sh:class schema:Place]
      [ sh:class dbo:Place ] )
    ];

  # node_type literal
  sh:property [ sh:path dbp:deathDate;
    sh:nodeKind sh:Literal;
    sh:datatype xsd:date ] .
```
## Experimental Settings

### Dataset

<table>
<thead>
<tr>
<th>Knowledge Bases</th>
<th>Dataset</th>
<th>Classes</th>
<th>Properties</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBpedia</td>
<td>dbo:Place</td>
<td>200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBpedia</td>
<td>foaf:Person</td>
<td>174</td>
<td></td>
<td>2016-04</td>
</tr>
<tr>
<td></td>
<td>dbo:Organization</td>
<td>219</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3cixty</td>
<td>lode:Events</td>
<td>215</td>
<td></td>
<td>2016-09-09</td>
</tr>
</tbody>
</table>
## Model Evaluation

### Integrity Constraints performance measures for 3cixty

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Minimum Cardinality F1 Score</th>
<th>Maximum Cardinality F1 Score</th>
<th>Range F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>0.91</td>
<td>0.93</td>
<td>0.91</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>0.81</td>
<td>0.81</td>
<td>0.90</td>
</tr>
<tr>
<td>Least Squares SVM</td>
<td>0.74</td>
<td>0.84</td>
<td>0.86</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.70</td>
<td>0.77</td>
<td>0.82</td>
</tr>
<tr>
<td>K-Nearest Neighbor</td>
<td>0.68</td>
<td>0.76</td>
<td>0.80</td>
</tr>
</tbody>
</table>

### Integrity Constraints performance measures for DBpedia

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Minimum Cardinality F1 Score</th>
<th>Maximum Cardinality F1 Score</th>
<th>Range F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>0.97</td>
<td>0.98</td>
<td>0.95</td>
</tr>
<tr>
<td>Least Squares SVM</td>
<td>0.97</td>
<td>0.90</td>
<td>0.89</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>0.95</td>
<td>0.88</td>
<td>0.84</td>
</tr>
<tr>
<td>K-Nearest Neighbor</td>
<td>0.94</td>
<td>0.87</td>
<td>0.83</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.88</td>
<td>0.83</td>
<td>0.84</td>
</tr>
</tbody>
</table>
Summary of findings

Evolution-based Quality Characteristics

RQ1: KB Evolution Analysis

Quality Assessment and Validation

RQ2: Evolution based Quality Assessment and Validation Approach

Experimental Analysis

RQ3: Evaluation
Summary of findings

Evolution Analysis to Drive Quality Assessment

- Causes of quality issues
  - Errors in the data source extraction process
  - Erroneous schema presentation
  - Errors in literal values

- Performance

<table>
<thead>
<tr>
<th>Knowledge Bases</th>
<th>Dataset</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Classes</td>
<td>Properties</td>
</tr>
<tr>
<td>DBpedia</td>
<td>10</td>
<td>4477</td>
</tr>
<tr>
<td>3cixty</td>
<td>2</td>
<td>149</td>
</tr>
</tbody>
</table>
Summary of findings

KBQ

Historical Persistency

What is Historical Persistency?

Historical persistency is a derived measurement function using the persistency measure over all releases of KB. Historical persistency dimensions explore entire KB evolution for a specific entity to detect inconsistencies. This metric assesses the persistency metric to provide insights on the series of KB releases it considers all entities presented in a KB and give an overview of the KB. Data curators can get an overview of knowledge base persistency issues over all releases. This helps data curators to decide which knowledge base release can be used for future data management tasks.

(80.0%)

Historical Persistency

Percentage % of historical persistency

Estimation of persistency issue over all KB releases

Interpretation:

High % presents an estimation of fewer issues, and lower % entails more issues present in KB releases.

Historical Persistency measures of selected class

<table>
<thead>
<tr>
<th>Release</th>
<th>version</th>
<th>count</th>
<th>Persistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010-04-11T22:00:00Z</td>
<td>3.5</td>
<td>21286</td>
<td>1</td>
</tr>
<tr>
<td>2011-01-16T23:00:00Z</td>
<td>3.6</td>
<td>46556</td>
<td>1</td>
</tr>
<tr>
<td>2011-06-07T22:00:00Z</td>
<td>3.7</td>
<td>78952</td>
<td>1</td>
</tr>
<tr>
<td>2012-08-05T22:00:00Z</td>
<td>3.8</td>
<td>235526</td>
<td>1</td>
</tr>
<tr>
<td>2013-06-18T22:00:00Z</td>
<td>3.9</td>
<td>256819</td>
<td>1</td>
</tr>
<tr>
<td>2014-06-28T22:00:00Z</td>
<td>2014</td>
<td>177872</td>
<td>0</td>
</tr>
</tbody>
</table>

Showing 1 to 6 of 6 entries

Repository: https://github.com/KBQ/
Limitations

- **Manual validation by inspecting data sources.**

- The **negative impact of erroneous addition of resources.**

- The **evaluation of the annotations requires considerable domain knowledge** to decide if a constraint is correct or incorrect.
Conclusion
Answers for research questions

RQ1  How can we identify quality issues with respect to KB evolution?

- Proposed evolution-based measures to detect quality issues
- Introduced four evolution-based quality characteristics using summary statistics
RQ2 Which quality assessment approach can be defined on top of the evolution based quality characteristics?

- Proposed a novel quality assessment approach using evolution-based quality characteristics
- Developed KBQ, a tool for KB quality assessment and validation using evolution-based quality characteristics
Answers for research questions

RQ3 Which approaches can be used to validate a KB evolution based quality assessment approach?

- Evaluated using qualitative approach based on manual validation
- Completeness characteristic is extremely effective and was able to achieve greater than 90% precision in error detection for both the use cases
- Performed validation by generating RDF shapes and learning models
- The best performing model in the experimental setup is the Random Forest, reaching an F1 value greater than 90% for minimum and maximum cardinality and 84% for range constraints
Future Work

- Extending to other quality characteristics
- Literal value analysis
- Impact of addition of resources
- Schema based validation
Publications


- Mohammad Rashid, Giuseppe Rizzo, Nandana Mihindukulasooriya, Marco Torchiano, and Oscar Corcho, "Knowledge Base Evolution Analysis: A Case Study in the Tourism Domain", In Proceedings of Workshops on Knowledge Graphs on Travel and Tourism co-located with 18th International Conference on Web Engineering (ICWE), Caceres, Spain, 2018

Conclusions

Publications


☐ Other papers published during the PhD


Thank You

Grazie
Introduction

State of the art

- Linked Data Dynamics
- Knowledge Base Quality Assessment
  - Comprehensive Surveys
  - Frameworks
- Knowledge Base Validation
  - Open World Assumption
  - Closed World Assumption

Threshold value analysis by using a histogram of property frequencies distribution.

- Univariate probability distribution is considered due to property frequency is the primary measurement element
- Frequency distribution of properties is unknown for each KB releases
- Update frequency varies with each KB
Lifespan Analysis of Evolving KBs

To measure KB growth, we applied linear regression analysis of entity counts over KB releases. In the regression analysis, we excluded the latest release to measure the normalized distance between an actual and a predicted value.

We define the normalized distance as:

$$ND(C) = \frac{\text{residual}_n(C)}{\text{mean}(|\text{residual}_n(C)|)}$$

Based on the normalized distance, we can measure KB growth of a class C as:

$$KB_{growth}(C) = \begin{cases} 1 & \text{if } ND(C) \geq 1 \\ 0 & \text{if } ND(C) < 1 \end{cases}$$
Lifespan Analysis of Evolving KBs

3cixty Knowledge Base

Summary of findings

**lode:Event** entity type

**dul:Places** entity type
Lifespan Analysis of Evolving KBs

DBpedia Knowledge Base

Summary of findings

**foaf:film entity type**

**dbo:Places entity type**