Automated Knowledge Base Quality Assessment and Validation based on Evolution Analysis

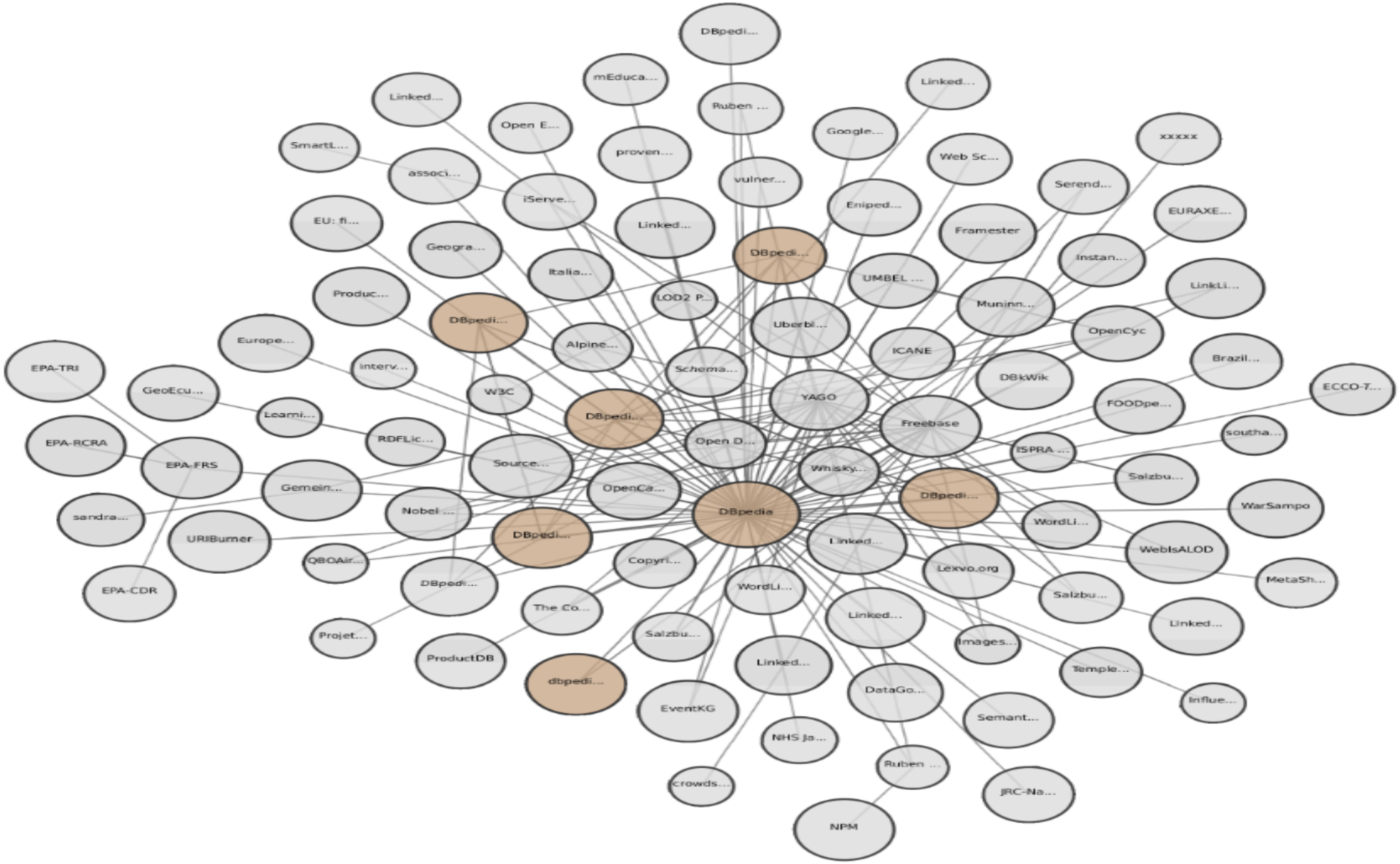
Mohammad Rashid

Supervisor: Prof. Marco Torchiano

Introduction

Knowledge Base Evolution

→DBpedia Knowledge Base – 2018*



Source: http://lod-cloud.net/

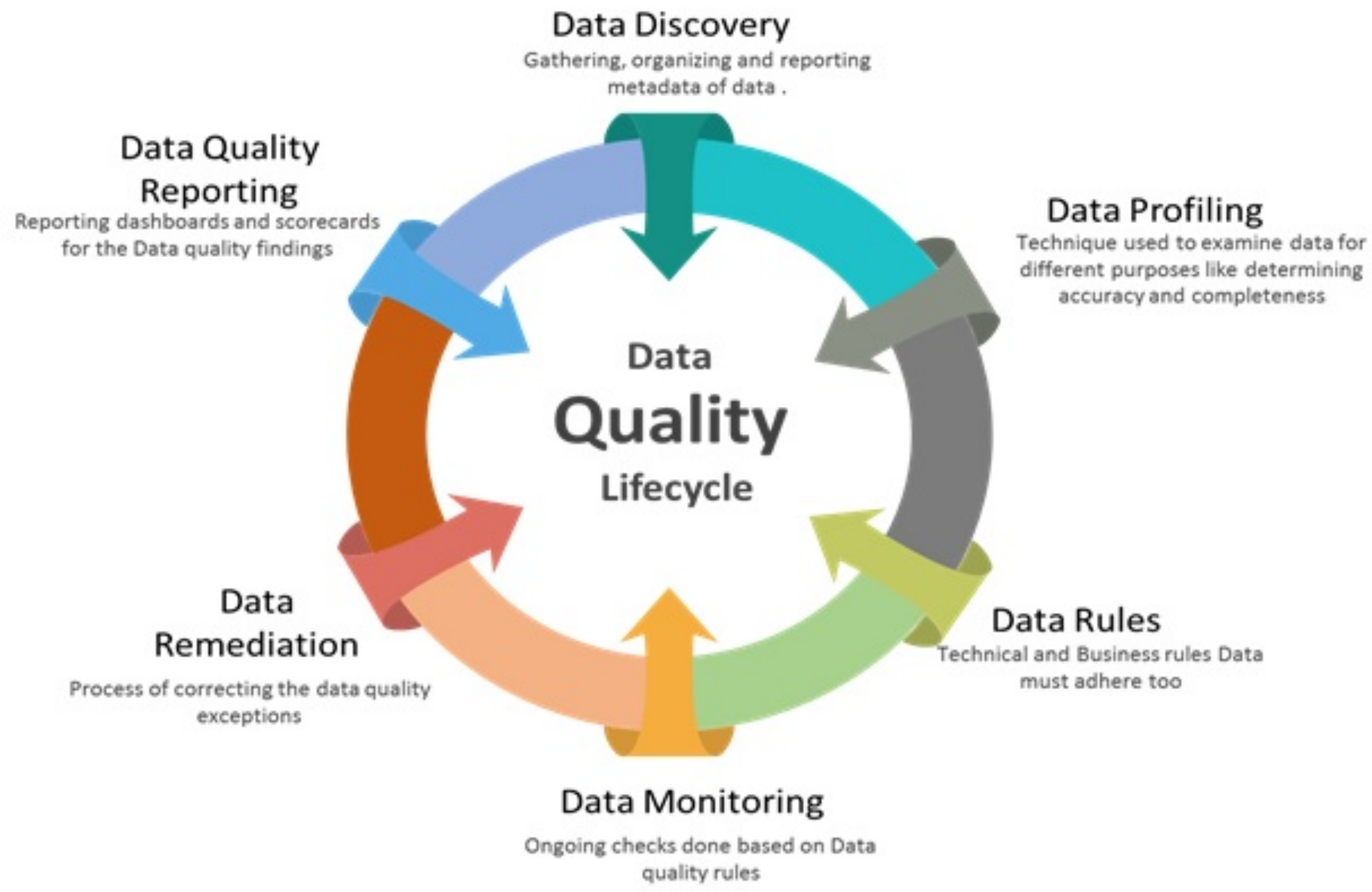
Knowledge Base Evolution

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

Evolution of KBs is unrestrained

Data Quality Analysis for **Evolving KBs**

Data Quality Life Cycle*


Data Discovery
Gathering, organizing and reporting metadata of data .
Data Profiling
Technique used to examine data for different purposes like determining accuracy and completeness
Data Rules
Technical and Business rules Data must adhere too
Data Monitoring
Ongoing checks done based on Data quality rules
Data Remediation
Process of correcting the data quality exceptions
Data Quality Reporting
Reporting dashboards and scorecards for the Data quality findings
Data
Quality
Lifecycle

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

Analysis Level

Analysis Level[1]	Detail	Volume	Stakeholder
Low-level	Fine-grained	Large	Data end-user
High-level	Coarse-grained	Small	Data Curator

[1] Vicky Papavasileiou, Giorgos Flouris, Irini Fundulaki, Dimitris Kotzinos, and Vassilis Christophides. High-level Change Detection in RDF(S) KBs. ACM Transactions on Database Systems (TODS), 38(1):1:1–1:42, April 2013.

Quality Issues

Identification of quality issues due to **unrestrained KB evolution**

Identification of **erroneous conceptualizations of resources**

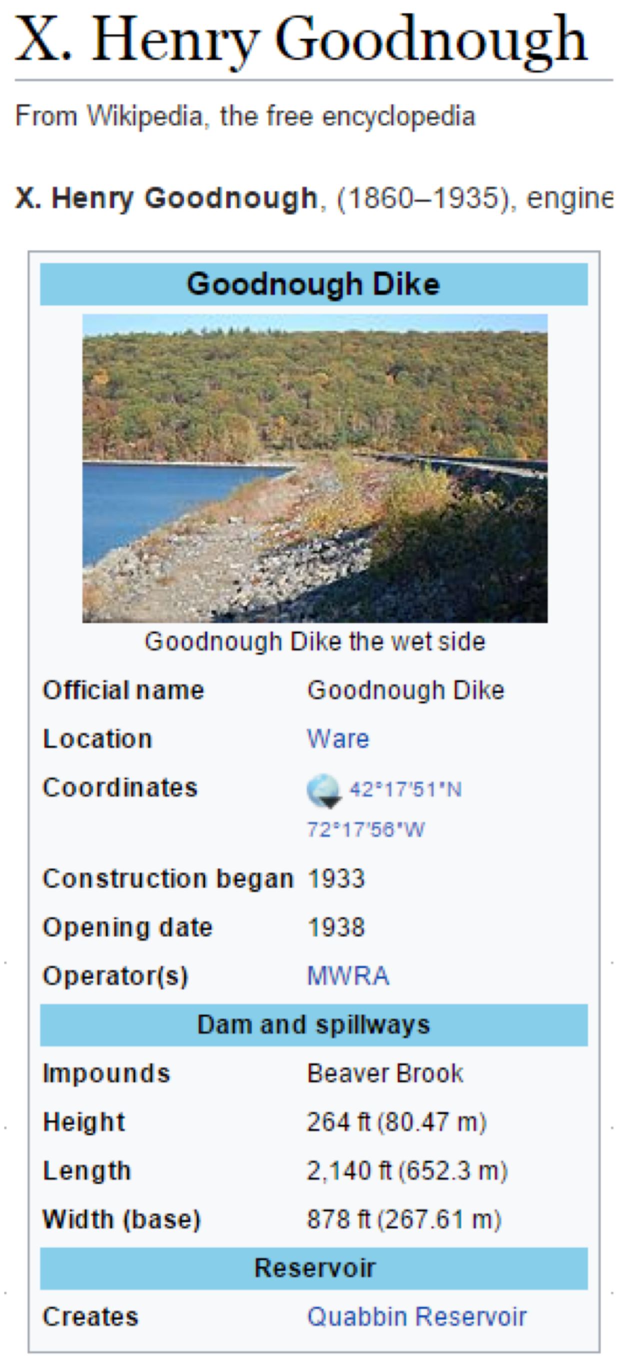
Quality Issues

❑ **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.



X. Henry Goodnough

From Wikipedia, the free encyclopedia

X. Henry Goodnough, (1860–1935), engine

Goodnough Dike	
Goodnough Dike the wet side	
Official name	Goodnough Dike
Location	Ware
Coordinates	42°17′51″N 72°17′56″W
Construction began	1933
Opening date	1938
Operator(s)	MWRA
Dam and spillways	
Impounds	Beaver Brook
Height	264 ft (80.47 m)
Length	2,140 ft (652.3 m)
Width (base)	878 ft (267.61 m)
Reservoir	
Creates	Quabbin Reservoir

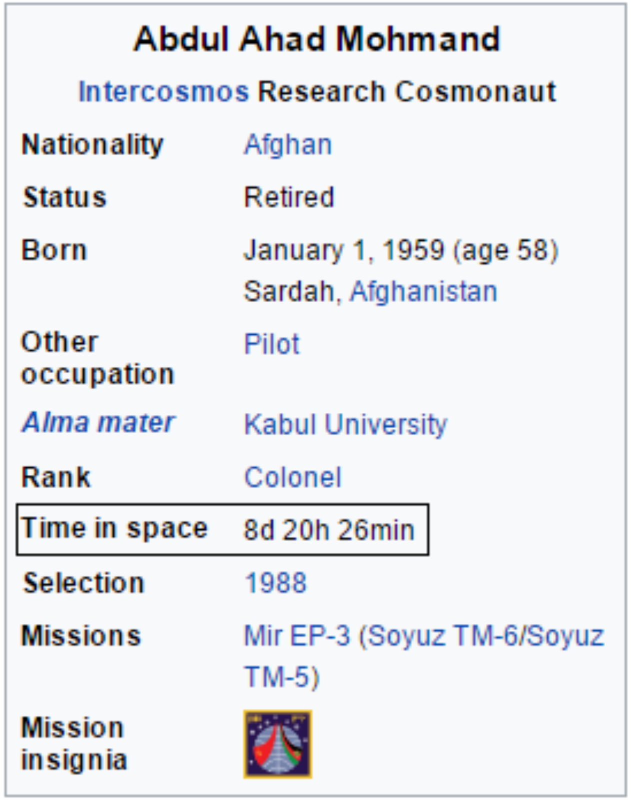
Quality Issues

- ❑ **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.



Abdul Ahad Mohmand

Intercosmos Research Cosmonaut

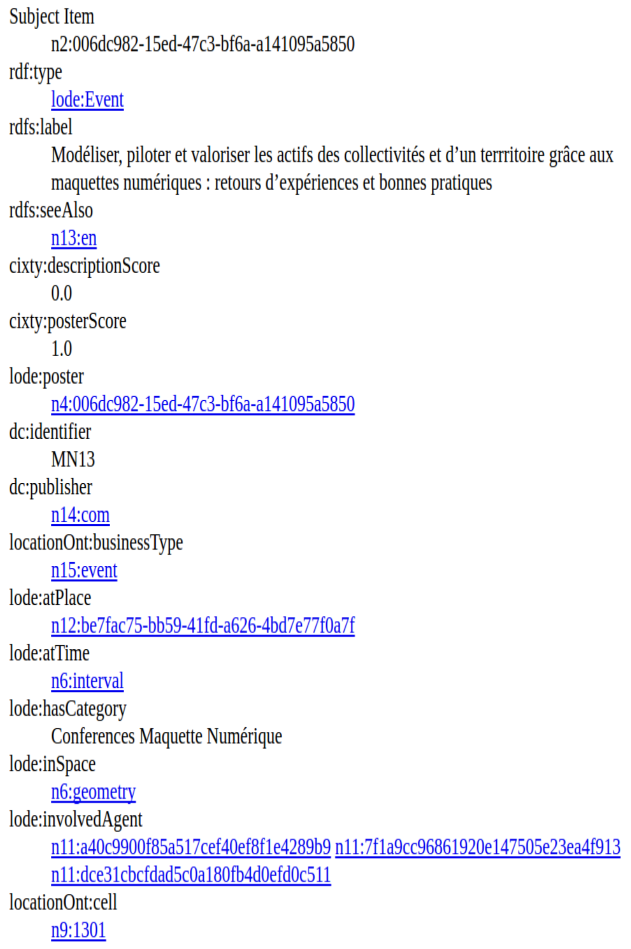
Nationality	Afghan
Status	Retired
Born	January 1, 1959 (age 58) Sardah, Afghanistan
Other occupation	Pilot
Alma mater	Kabul University
Rank	Colonel
Time in space	8d 20h 26min
Selection	1988
Missions	Mir EP-3 (Soyuz TM-6/Soyuz TM-5)
Mission insignia	

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 terrritoire 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.



```
Subject Item
      n2:006dc982-15ed-47c3-bf6a-a141095a5850
rdf:type
      lode:Event
rdfs:label
      Modéliser, piloter et valoriser les actifs des collectivités et d’un terrritoire grâce aux
      maquettes numériques : retours d’expériences et bonnes pratiques
rdfs:seeAlso
      n13:en
cixty:descriptionScore
      0.0
cixty:posterScore
      1.0
lode:poster
      n4:006dc982-15ed-47c3-bf6a-a141095a5850
dc:identifier
      MN13
dc:publisher
      n14:com
locationOnt:businessType
      n15:event
lode:atPlace
      n12:be7fac75-bb59-41fd-a626-4bd7e77f0a7f
lode:atTime
      n6:interval
lode:hasCategory
      Conferences Maquette Numérique
lode:inSpace
      n6:geometry
lode:involvedAgent
      n11:a40c9900f85a517cef40ef8f1e4289b9 n11:7f1a9cc96861920e147505e23ea4f913
      n11:dce31cbcfdad5c0a180fb4d0efd0c511
locationOnt:cell
      n9:1301
```

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?**

Problem

Identification of erroneous conceptualizations of resources

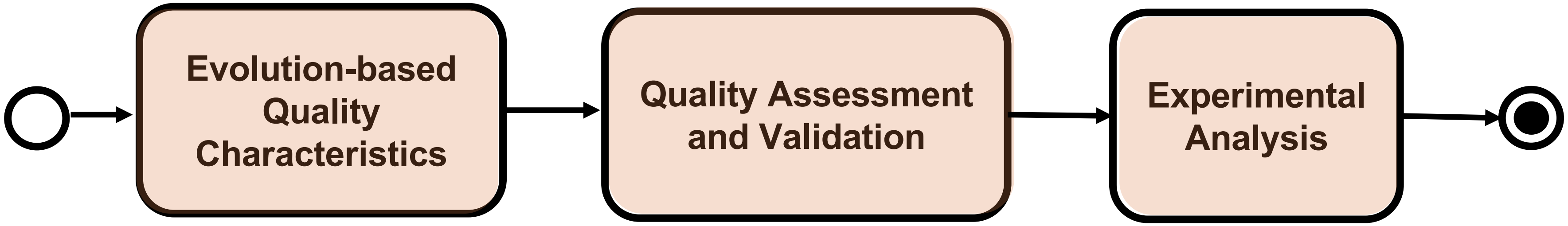
Hypothesis

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

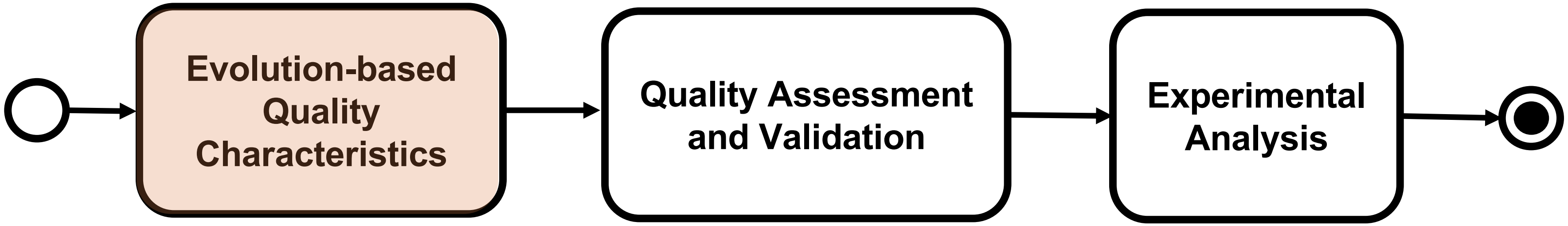

Evolution-based Quality Characteristics
Quality Assessment and Validation
Experimental Analysis

RQ1: KB Evolution Analysis

RQ2: Evolution based Quality Assessment and Validation Approach

RQ3: Evaluation

Evolution-based Quality Characteristics

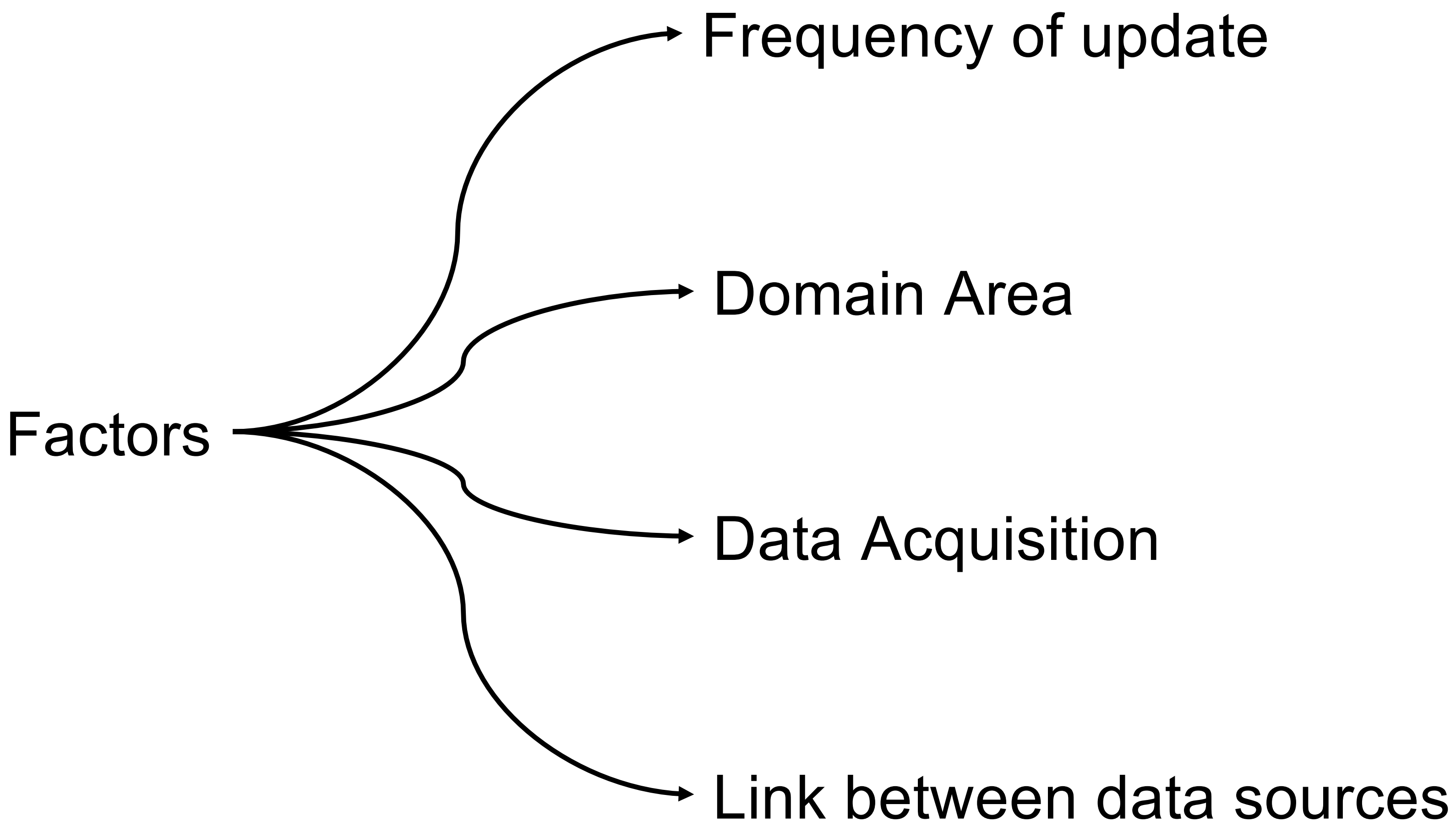

Evolution-based Quality Characteristics
Quality Assessment and Validation
Experimental Analysis

RQ1: KB Evolution Analysis

RQ2: Evolution based Quality Assessment and Validation Approach

RQ3: Evaluation

Evolution Analysis



Factors

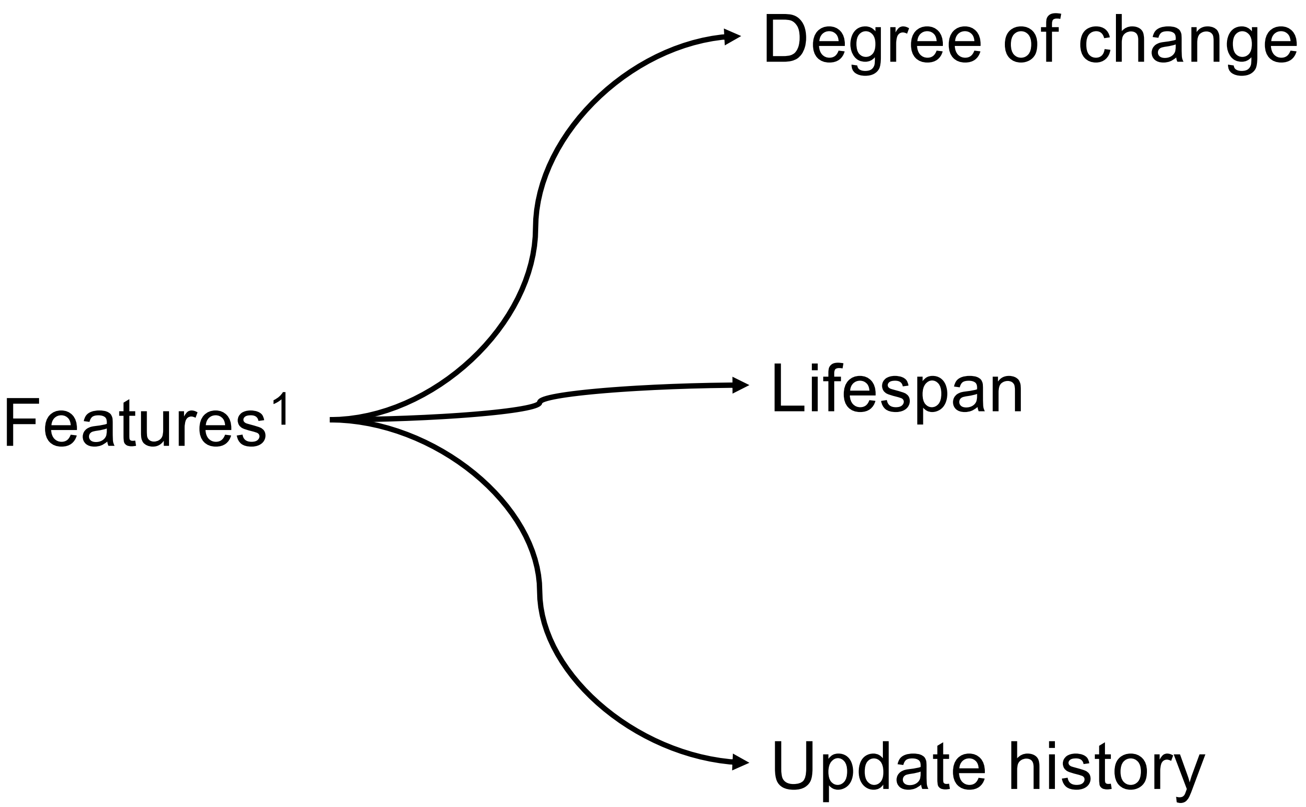
Frequency of update

Domain Area

Data Acquisition

Link between data sources

Dynamic Features



Features[1]

Degree of change

Lifespan

Update history

[1] Mohamed Ben Ellefi, Zohra Bellahsene, J Breslin, Elena Demidova, Stefan Dietze, Julian Szymanski, and Konstantin Todorov. RDF Dataset Profiling – a Survey of Features, Methods, Vocabularies and Applications. Semantic Web, pages 1–29, 2018.

Evolution-based Quality Characteristics

Dimensions[1]	Characteristics	Features[3]
Intrinsic	Persistency	Degree of change
	Historical Persistency	Lifespan
Representational	Consistency[2]	Update history
	Completeness[2]	

1. Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality Assessment for linked Data: A Survey. Semantic Web, 7(1):63–93, 2016.
2. ISO/IEC. 25012:2008 – software engineering – software product quality requirements and evaluation (square) – data quality model. Technical report, ISO/IEC, 2008.
3. Mohamed Ben Ellefi, Zohra Bellahsene, J Breslin, Elena Demidova, Stefan Dietze, Julian Szymanski, and Konstantin Todorov. RDF Dataset Profiling – a Survey of Features, Methods, Vocabularies and Applications. Semantic Web, pages 1–29, 2018.

Basic Measure Elements

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

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

❑ 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:

$$freq(p, C) = |\{(s, p, o) \in V : \exists (s, typeof, C) \in V\}|$$

Persistency

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

$$Persistency_i = \begin{cases} 1\ if\ count_i(C)\ \geq count_{i-1}(C) \\ 0\ if\ count_i(C) <\ count_{i-1}(C) \end{cases}$$

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

$$Persistency_i = \frac{\sum_{j=1}^{NC} Persistency_j(C_j)}{NC}$$

where *NC* is the number of classes analyzed in the KB.

Historical Persistency

- ❑ 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.

$$H\_Persistency(C) = \frac{\sum_{i=2}^{n} Persistency_i(C)}{n-1}$$

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:

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

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:

$$Nf_i(p, C) = \frac{freq_i(p, C)}{count_i(C)}$$

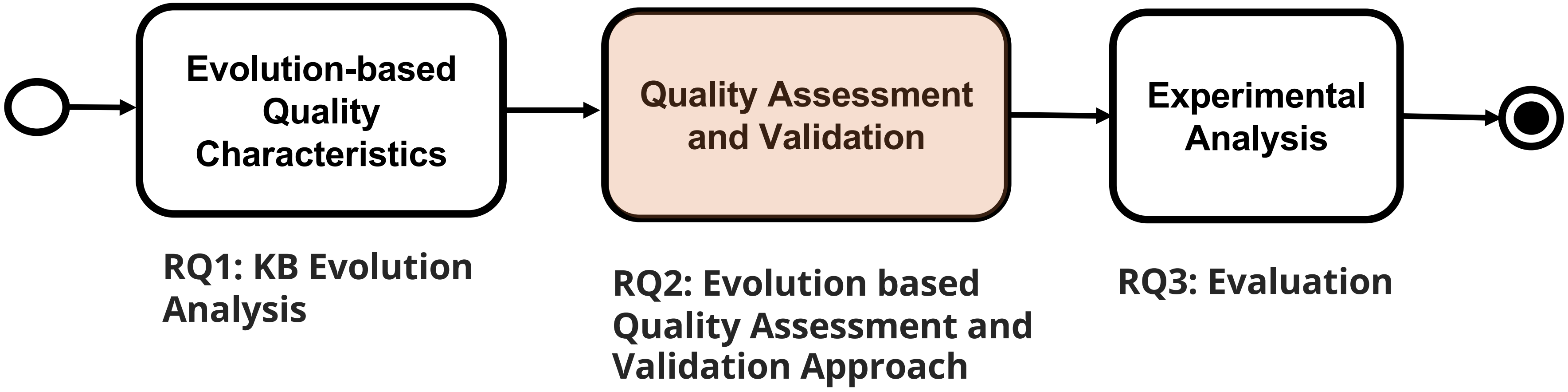
- ❑ Completeness of a property p in the scope of a class C is:

$$Completeness_i(p, C) = \begin{cases} 1 \; if \; Nf_i(p, C) \; \geq Nf_{i-1}(p, C) \\ 0 \; if \; Nf_i(p, C) < \; Nf_{i-1}(p, C) \end{cases}$$

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

$$Completeness_i(C) = \frac{\sum_{k=1}^{NP_i(C)} Completeness_i(p_k, C)}{NP_i(C)}$$

Evolution-based Quality Assessment and Validation Approach



Evolution-based Quality Characteristics

Quality Assessment and Validation

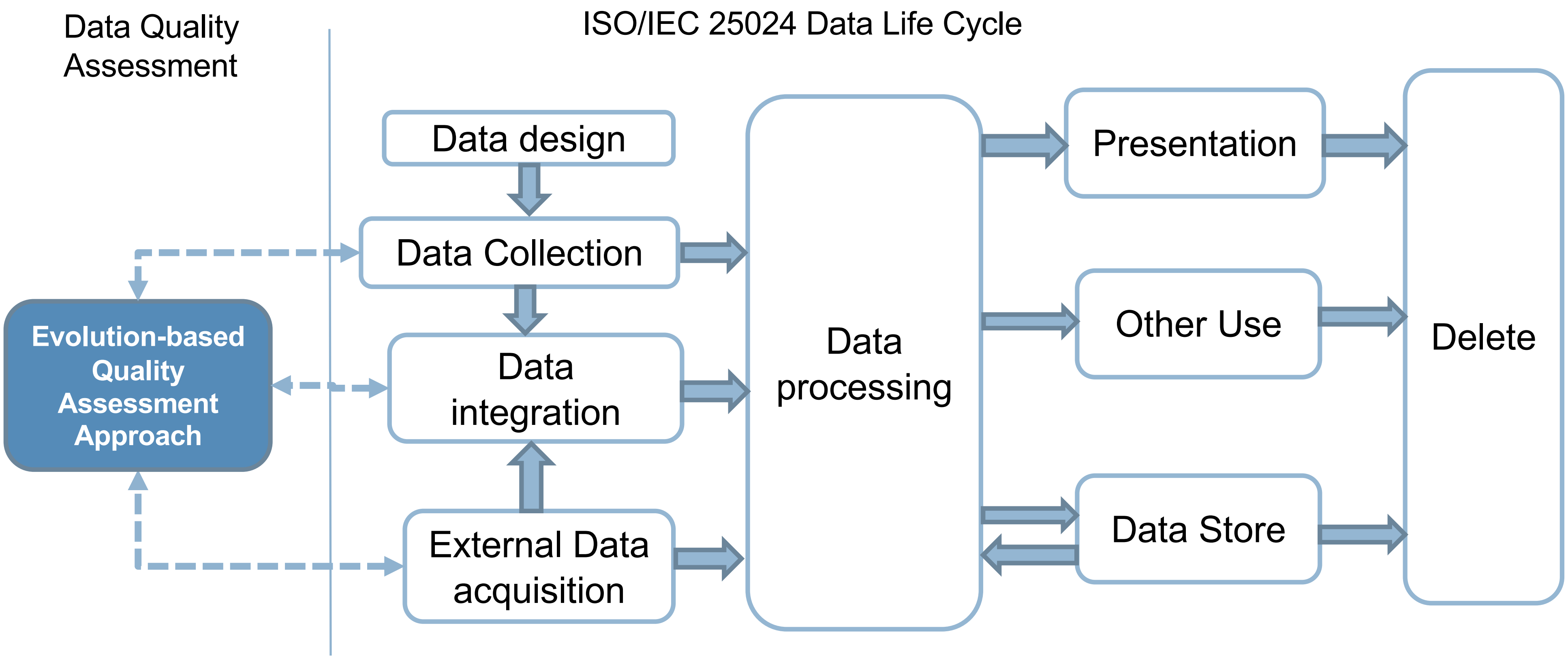
Experimental Analysis

RQ1: KB Evolution Analysis

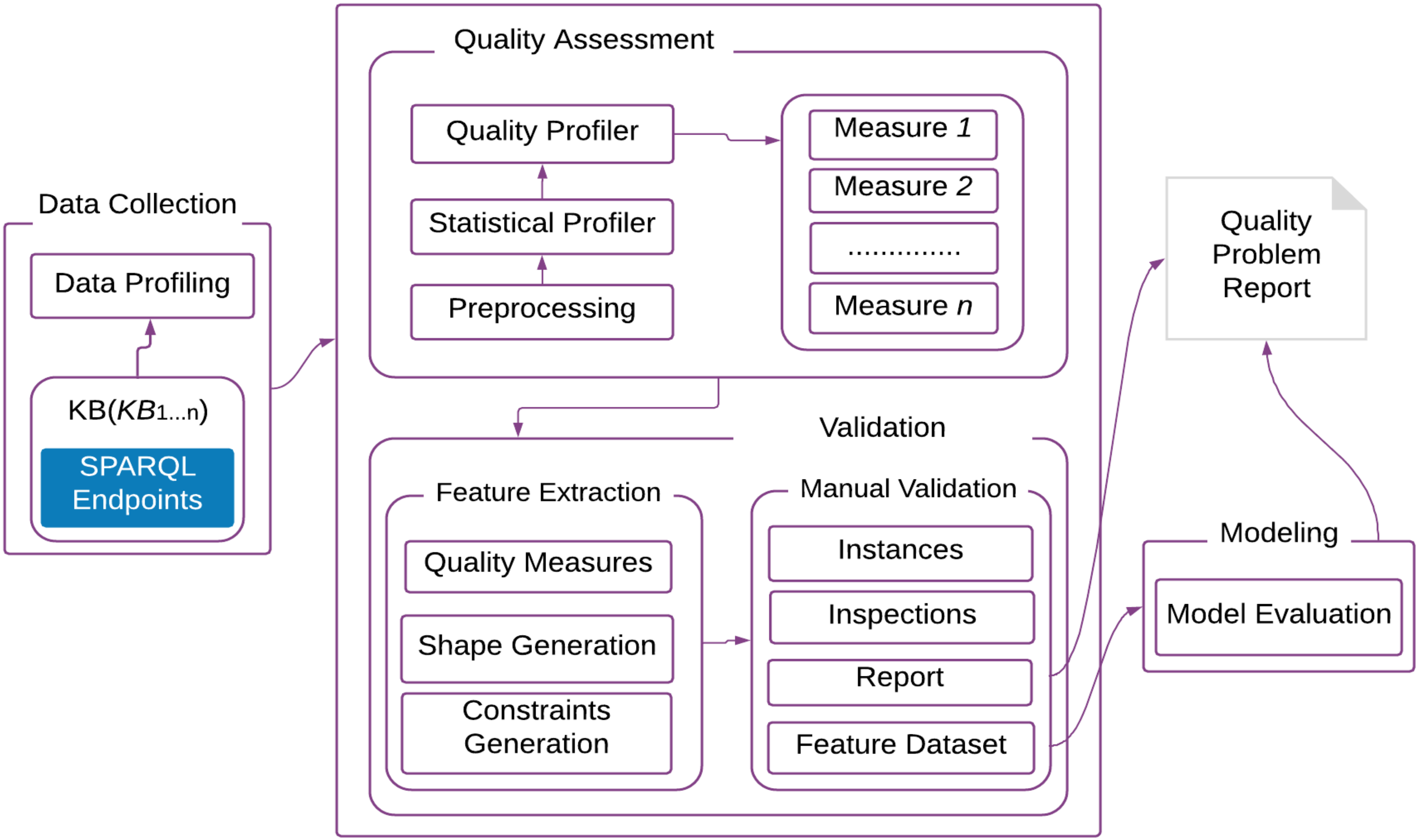
RQ2: Evolution based Quality Assessment and Validation Approach

RQ3: Evaluation

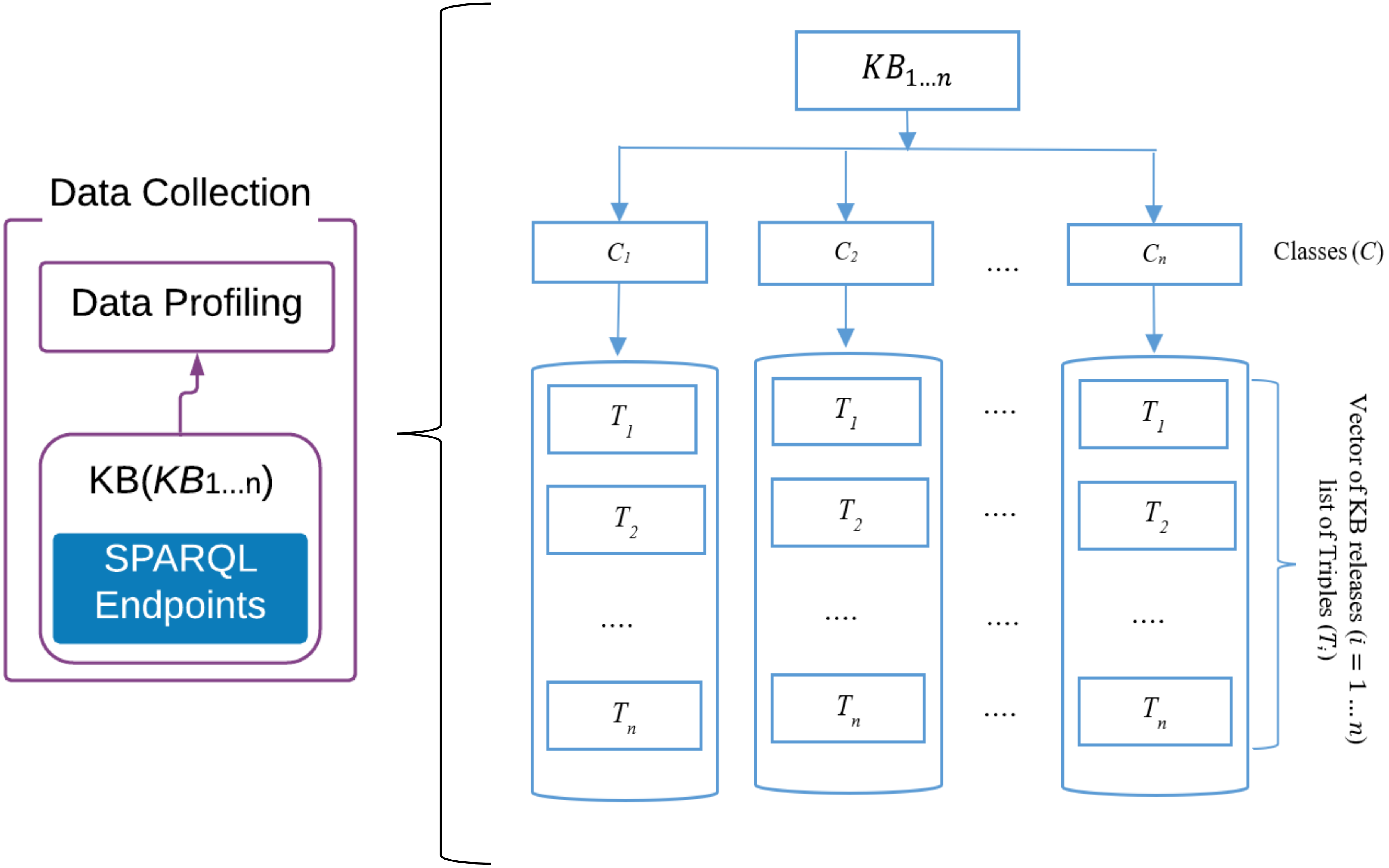
Data Life Cycle



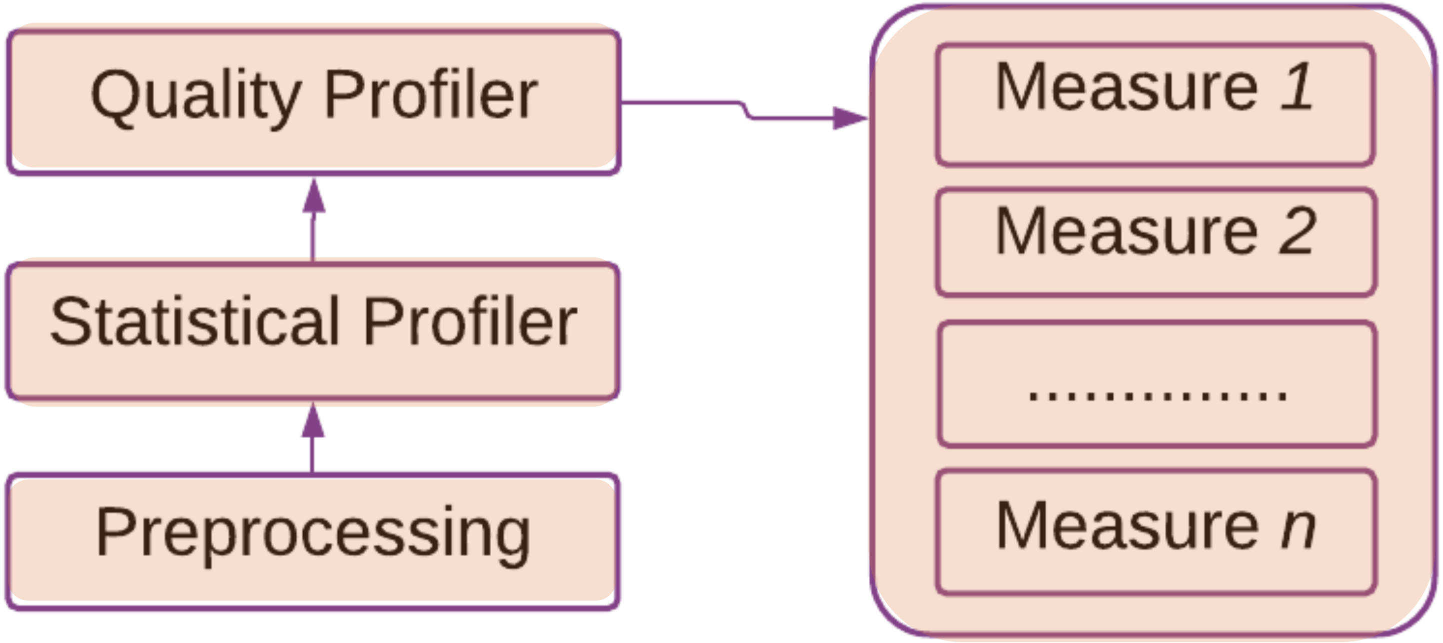
Proposed Approach



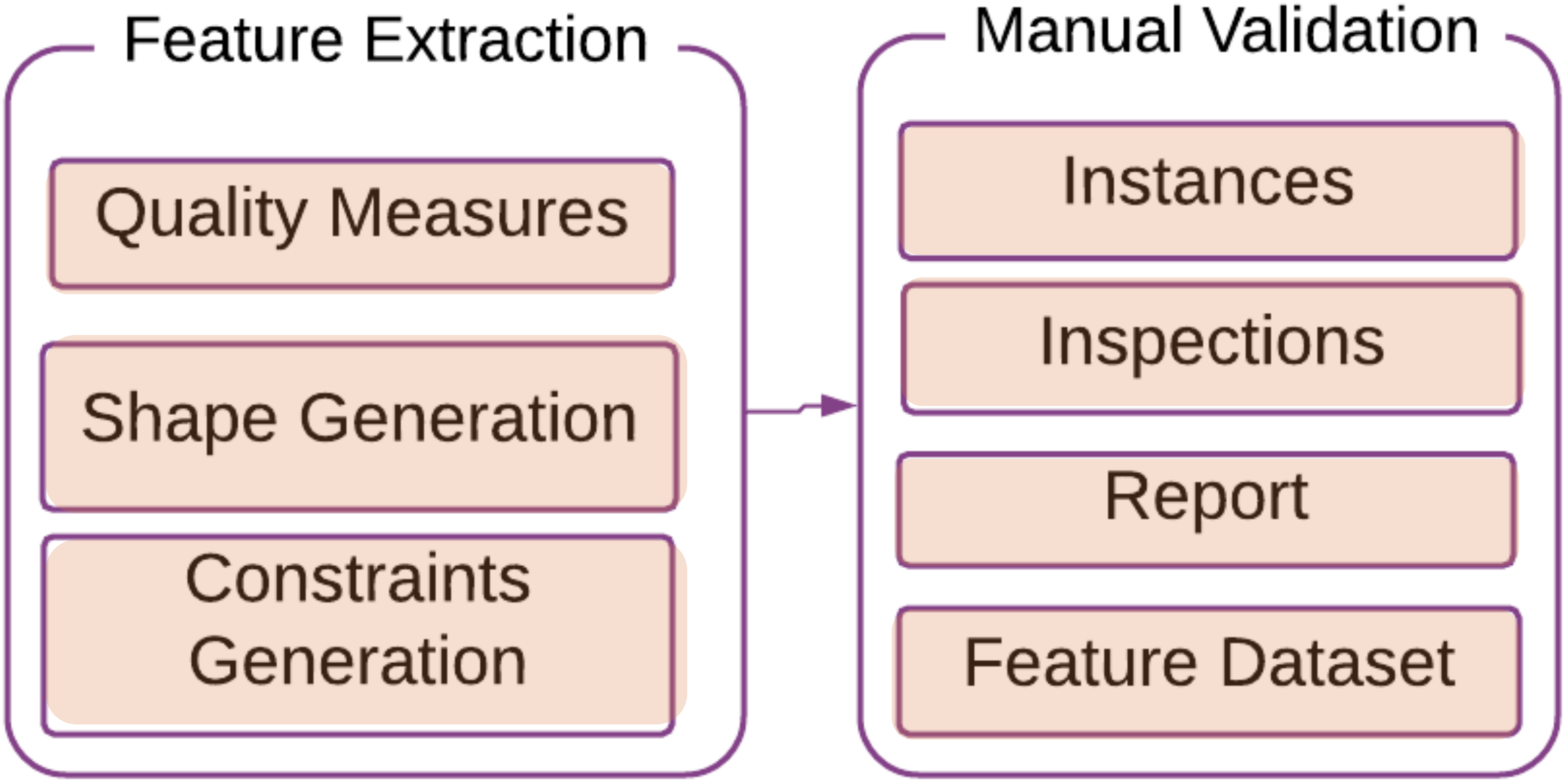
Data Collection



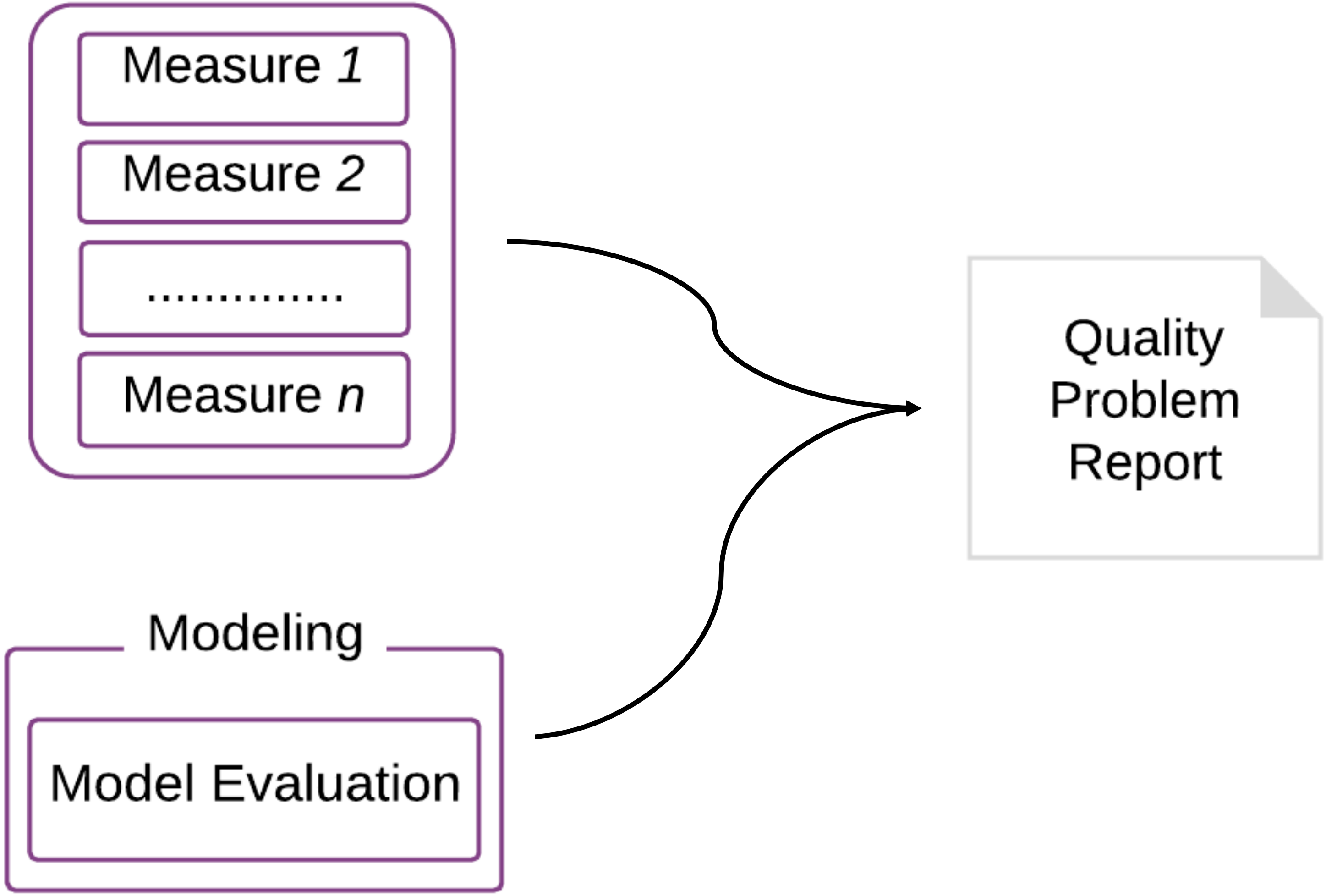
Quality Assessment



Validation Approaches



Modeling and Quality Problem Report



Measure *1*

Measure *2*

....

Measure *n*

Modeling

Model Evaluation

Quality Problem Report

Experimental Analysis

Evolution-based Quality Characteristics

Quality Assessment and Validation

Experimental Analysis

RQ1: KB Evolution Analysis

RQ2: Evolution based Quality Assessment and Validation Approach

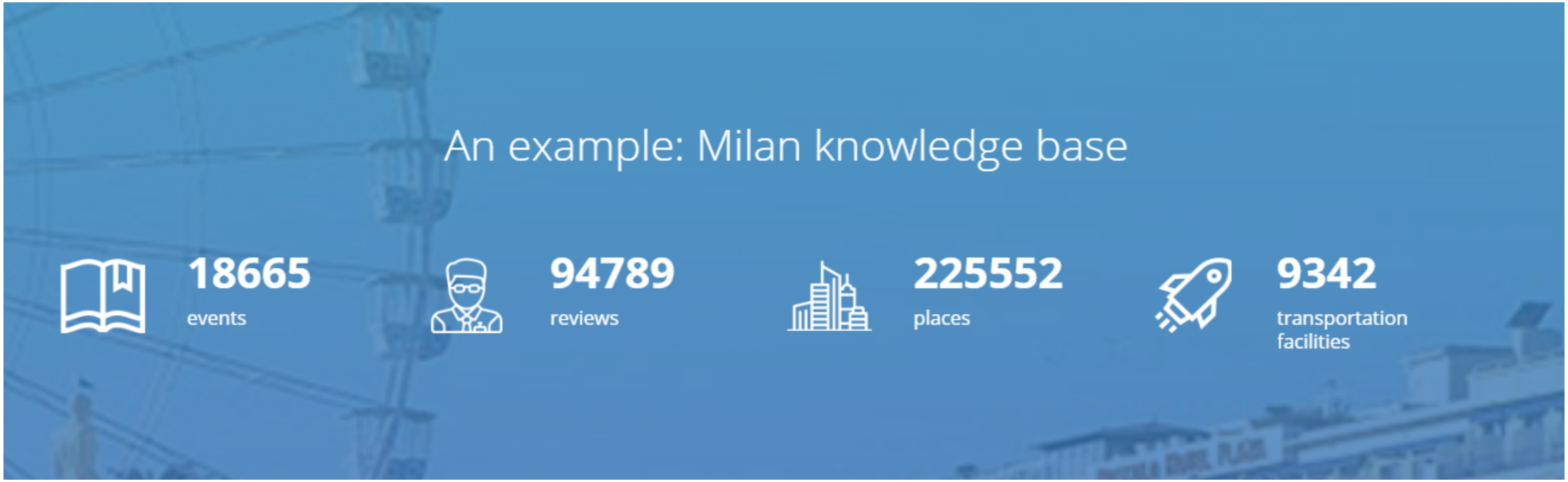
RQ3: Evaluation

Use case: 3cixty

❑ Cultural and tourist information[1].

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

❑ Nice, Milan, London, Singapore, and Madeira island.



An example: Milan knowledge base

18665 events

94789 reviews

225552 places

9342 transportation facilities

[1] Raphaël Troncy et al. 3cixty: Building comprehensive knowledge bases for city exploration. Web Semantics: Science, Services and Agents on the World Wide Web , 46-47:2 – 13, 2017.

Use case: DBpedia

❑ This knowledge base is the output of the DBpedia[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.

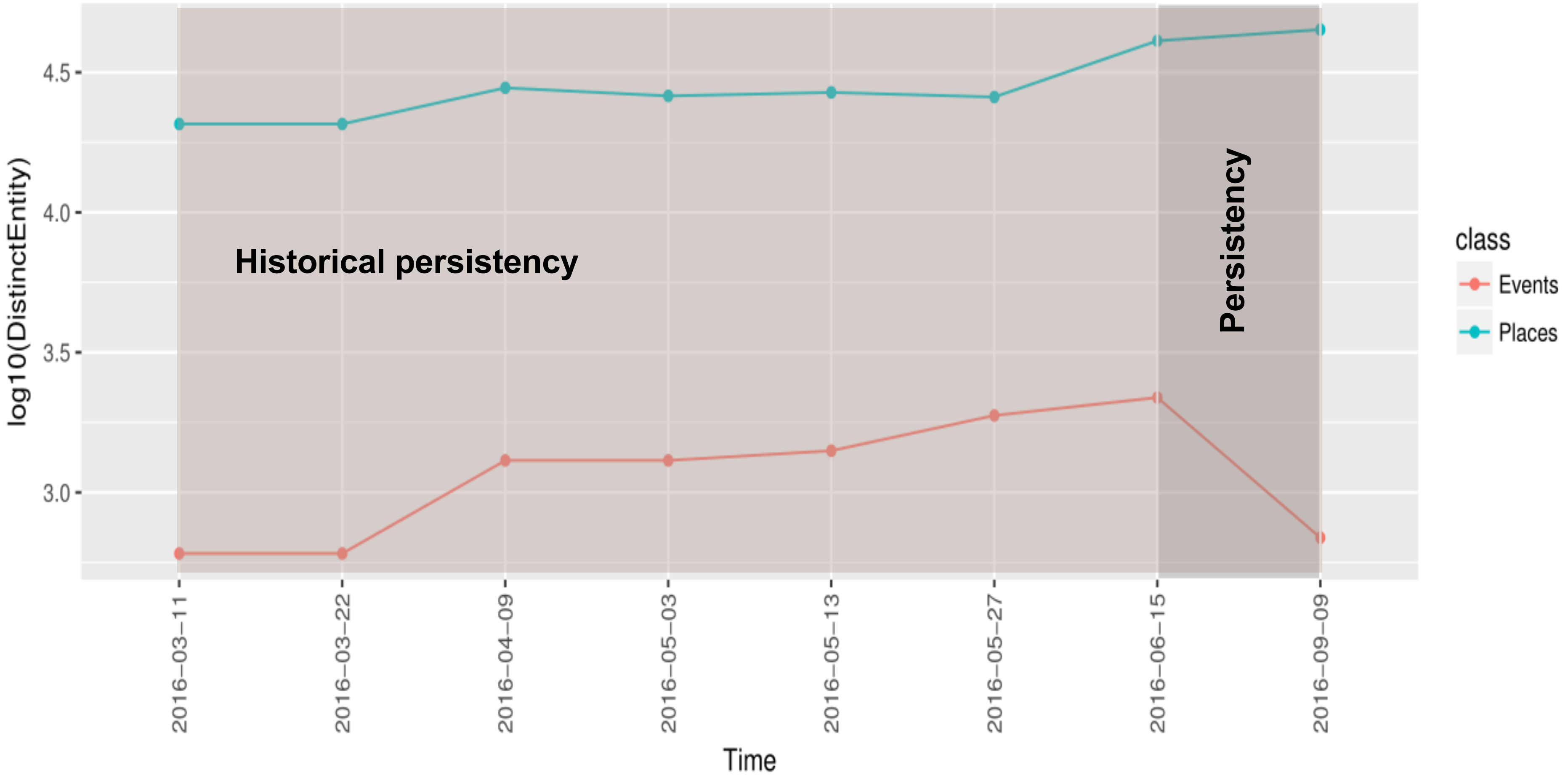
[1] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2):167–195, 2015

Experimental Settings

Knowledge Bases	Datasets		
	Classes	Properties	Releases
DBpedia	10	4477	11
3cixty	2	149	8

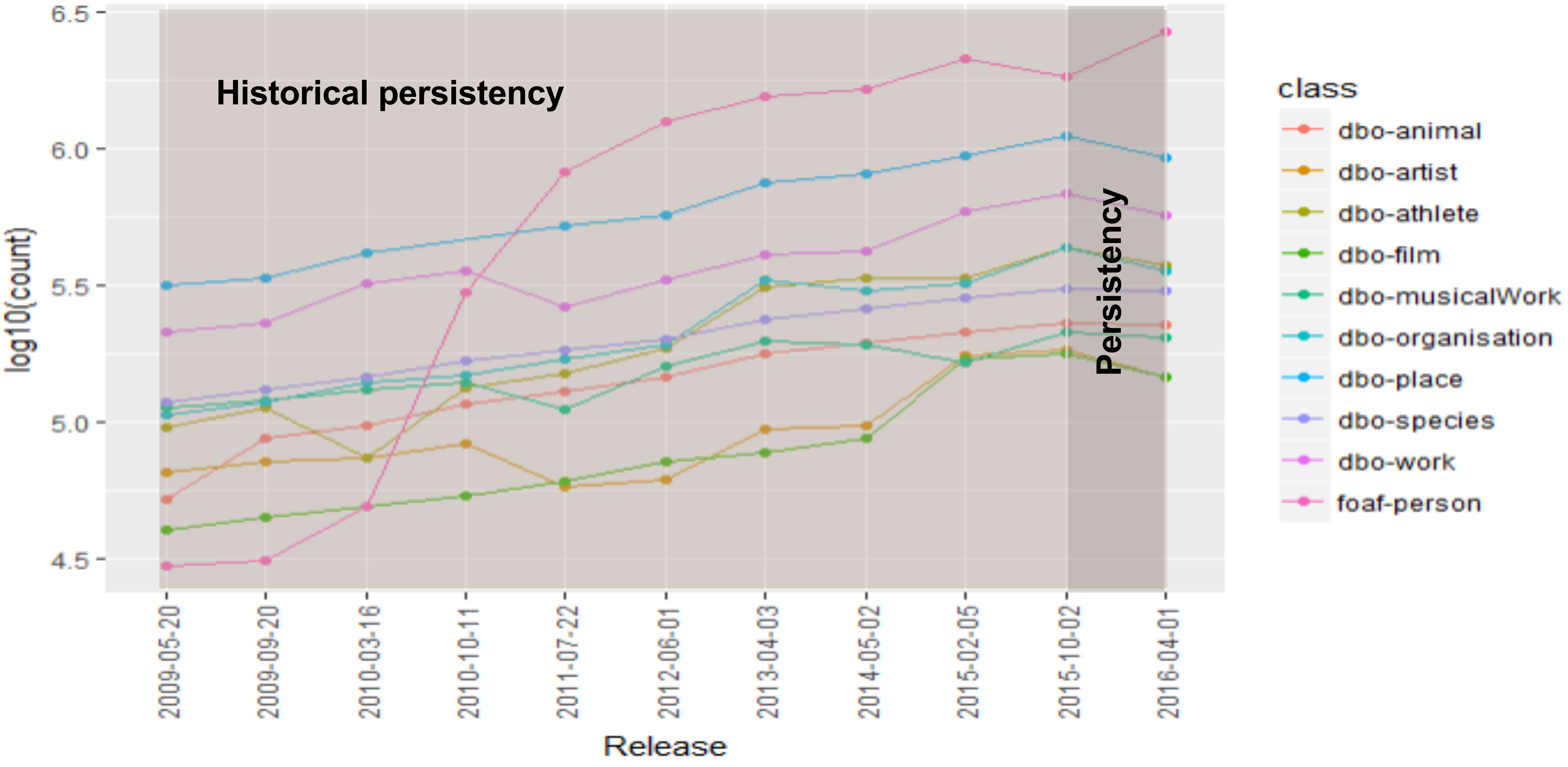
Quantitative Analysis: Persistency & Historical persistency

3cixty Knowledge Base



Quantitative Analysis: Persistency & Historical persistency

DBpedia Knowledge Base



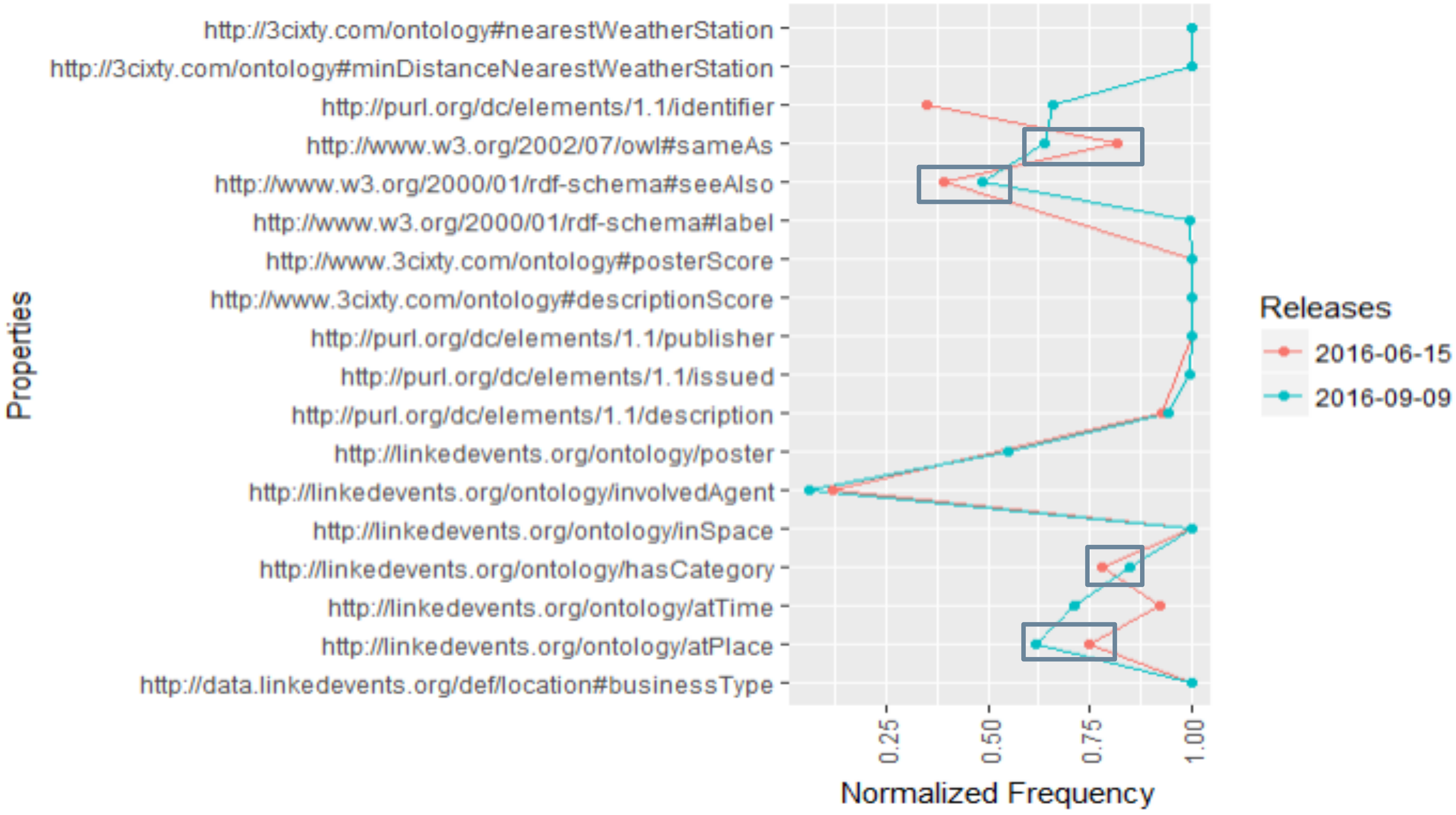
Quantitative Analysis: Consistency

DBpedia Knowledge Base

Class	Total	Inconsistent	Consistent
dbo:Animal	162	123	39
dbo:Artist	429	329	100
dbo:Athelete	436	298	138
dbo:Film	450	298	152
dbo:MusicalWork	325	280	45
dbo:Organisation	1,014	644	370
dbo:Place	1,090	589	501
dbo:Species	99	57	42
dbo:Work	935	689	276
foaf:Person	381	158	223

Quantitative Analysis: Completeness

3cixty Knowledge Base: *lode:Events*



Quantitative Analysis: Completeness

DBpedia Knowledge Base

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

Qualitative Analysis: Manual Validation

❑ Precision for evaluating the effectiveness of our approach

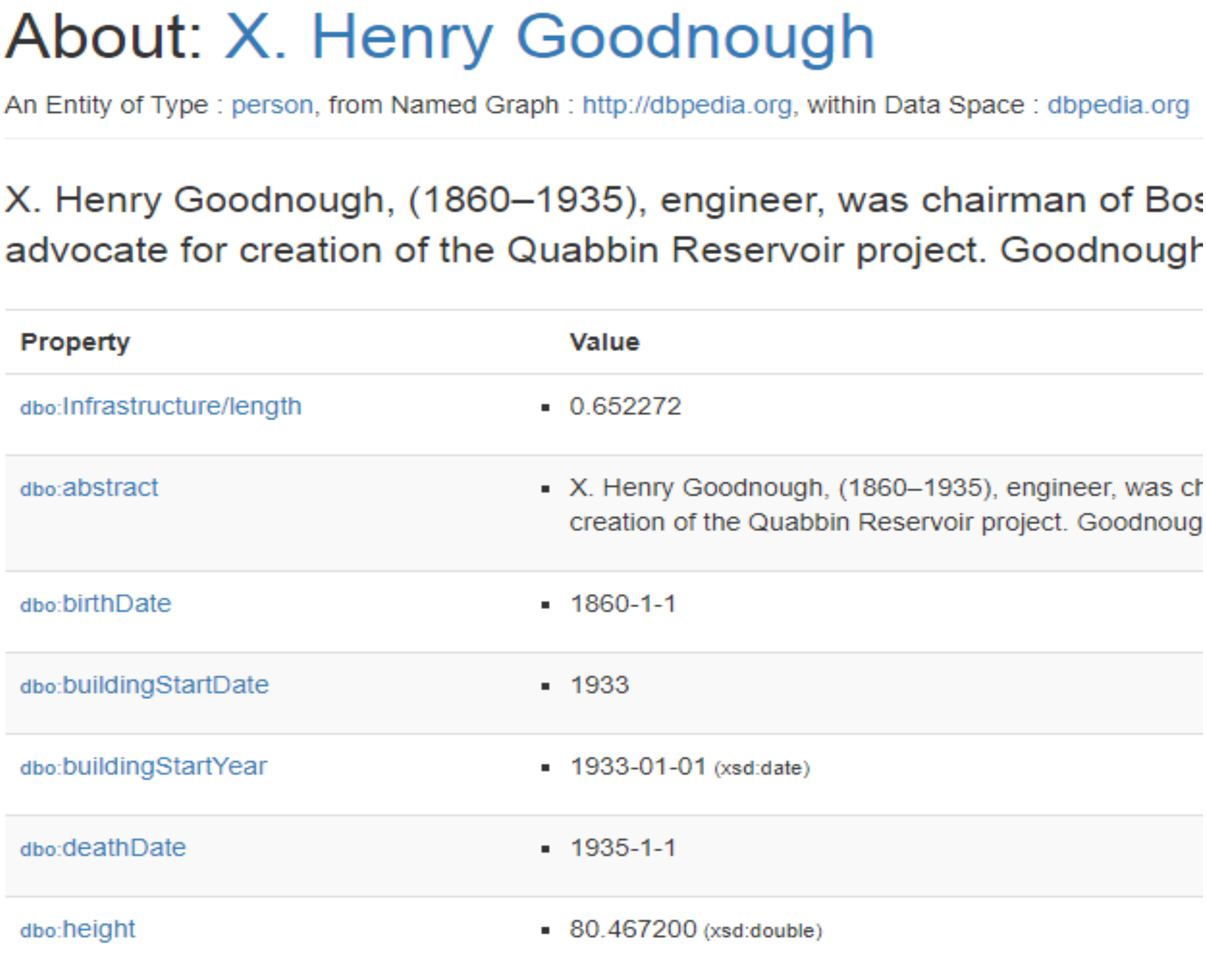
❑ Precision is defined as the proportion of accurate results of a quality measure over the total results

❑ 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

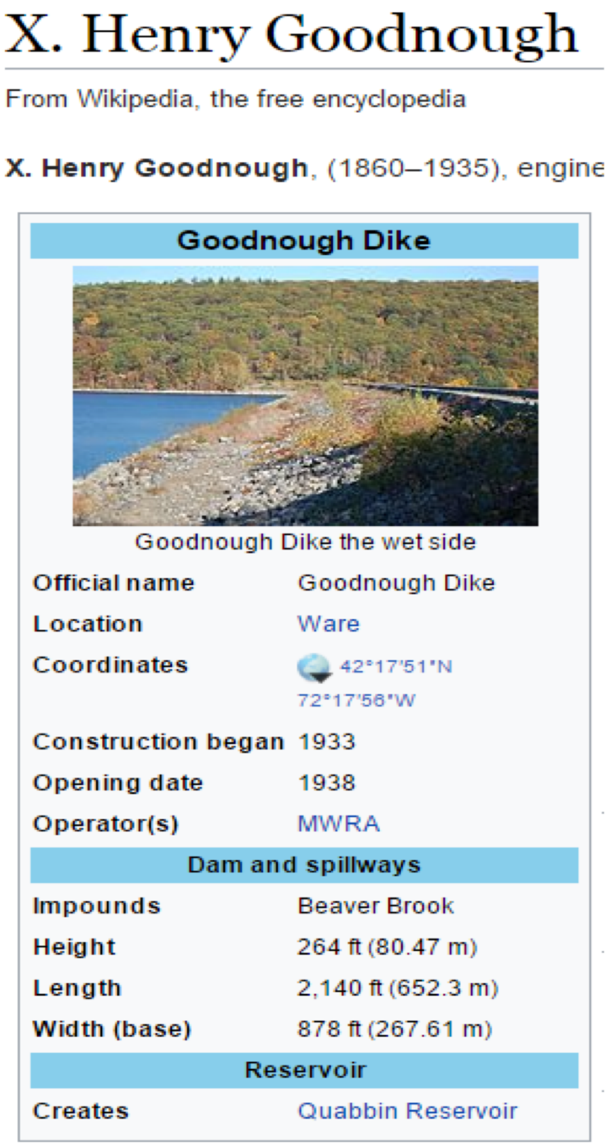


About: X. Henry Goodnough

An Entity of Type : person, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

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

Property	Value
dbo:Infrastructure/length	▪ 0.652272
dbo:abstract	▪ X. Henry Goodnough, (1860–1935), engineer, was ch creation of the Quabbin Reservoir project. Goodnoug
dbo:birthDate	▪ 1860-1-1
dbo:buildingStartDate	▪ 1933
dbo:buildingStartYear	▪ 1933-01-01 (xsd:date)
dbo:deathDate	▪ 1935-1-1
dbo:height	▪ 80.467200 (xsd:double)



X. Henry Goodnough

From Wikipedia, the free encyclopedia

X. Henry Goodnough, (1860–1935), engine

Goodnough Dike

Goodnough Dike the wet side

Official name	Goodnough Dike
Location	Ware
Coordinates	42°17′51″N 72°17′56″W
Construction began	1933
Opening date	1938
Operator(s)	MWRA
Dam and spillways	
Impounds	Beaver Brook
Height	264 ft (80.47 m)
Length	2,140 ft (652.3 m)
Width (base)	878 ft (267.61 m)
Reservoir	
Creates	Quabbin Reservoir

Qualitative Analysis: Manual Validation

KB	Quality Characteristics	Level	Experiment
3cixty Nice	Persistency & Historical Persistency	Class	*lode:Event*
	Completeness	Property	*lode:Event 8* properties
	Consistency	Property	*lode:Event* 10 properties
DBpedia	Persistency & Historical Persistency	Class	*dbo:Species* and *dbo:Film*
	Completeness	Property	*foaf:Person* and *dbo:Place* class 50 properties
	Consistency	Property	*foaf:Person* class 158 properties and *dbo:Place* class 114 properties

Qualitative Analysis: Manual Validation

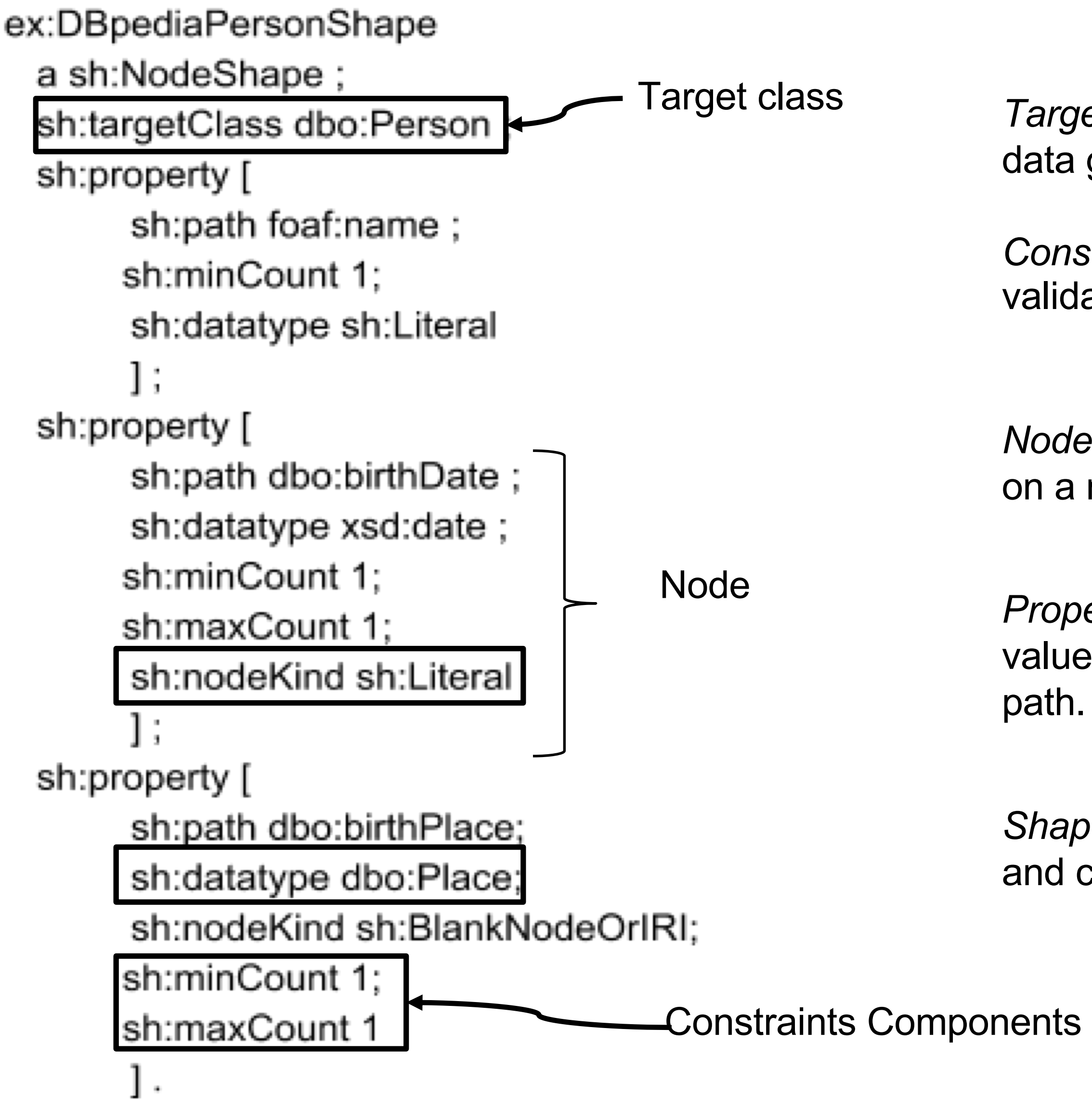
Quality Characteristics	**3cixty**	**DBpedia**
Persistency & Historical Persistency	*Error in reconciled algorithm*	*Fixed in the current version*
Consistency	*No real issues were found in the properties. Scheme remains consistent for all the KB releases*	*We found issues in the properties due to erroneous conceptualization*
Completeness	*Error in reconciled algorithm.* *Precision 95%*	*Error due to erroneous conceptualization and missing resources.* *Precision 94%*

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**

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

Node

Constraints Components

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

Constraints Type	Parameters
Cardinality	minCount, maxCount
Types of Values	Node Kind
Range of Values	minInclusive, maxInclusive, minExclusive, maxExclusive
String Based	minLength, maxLength, pattern,
Property pair	lessThan, lessThanOrEquals, disjoint, equal
Others	class, datatype, in, hasValue, ignoredProperties

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

Cardinality	Key	Description
Minimal cardinality	MIN0	Minimum Cardinality = 0
	MIN1+	Minimum Cardinality > 1
Maximum cardinality	MAX1	Maximum Cardinality = 1
	MAX1+	Maximum Cardinality >1

Feature Extraction: Cardinality Constraints

SPARQL Query:

```
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)
```

Raw cardinalities: 0, 0, 0, 0, 0, 1, 0, 1, 2, 3, 0, 0, 0, 2, 1, 4 …

Cardinalities of class property dbo:Sport / dbo:union

Cardinality	Instance Count	Percentage
0	1662	0.84883
1	279	0.14249
2	10	0.00511
3	5	0.00255
4	2	0.00102

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

MIN0	Minimum Cardinality = 0
MAX1+	Maximum Cardinality >1

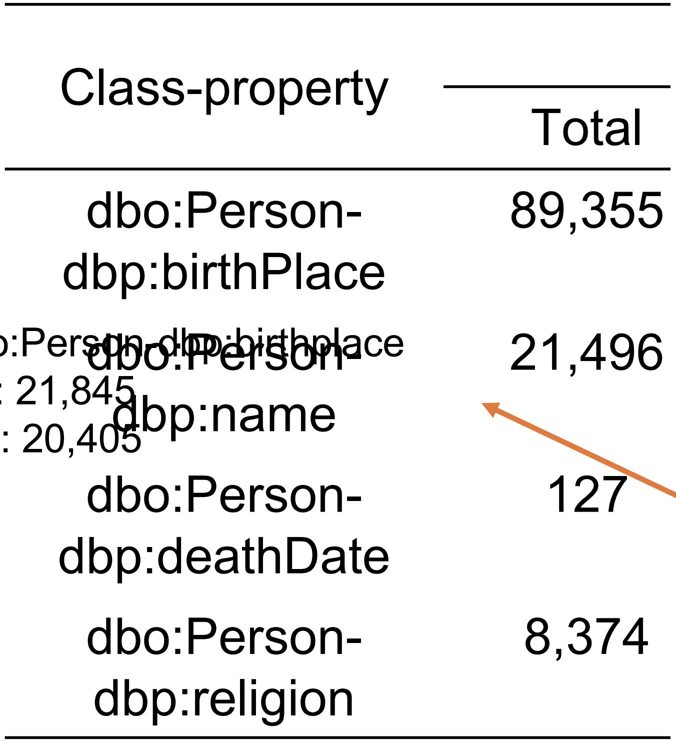
Range Constraints

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).

	IRI	Blank Node	Literal	Type
	X	X	X	Any
	X	X		BlankNodeOrIRI
➡	**X**			**IRI**
		X		BlankNode
➡			**X**	**Literal**
	X		X	IRIOrLiteral
		X	X	BlankNodeOrLiteral

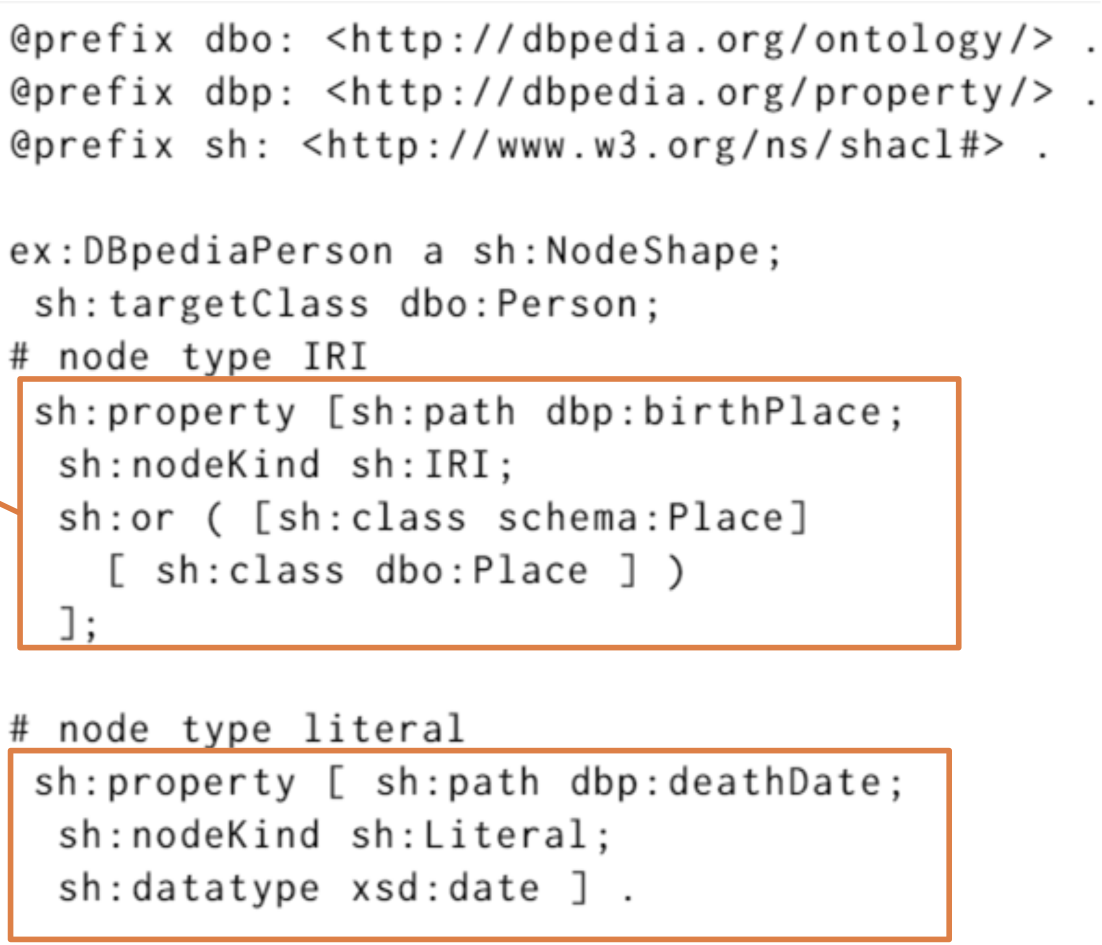
Feature Extraction: Range Constraints

Object node type information: IRI or LIT ?



Class-property	Total
dbo:Person-dbp:birthPlace	89,355
dbo:Person-dbp:name	21,496
dbo:Person-dbp:deathDate	127
dbo:Person-dbp:religion	8,374

dbo:Person-dbp:birthplace
IRI: 21,845
LIT: 20,405



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

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 ] .
```

dbo:Person-dbp:deathDate
IRI: 111
LIT: 32,449

Experimental Settings

Knowledge Bases	**Dataset**		
	Classes	Properties	Release
DBpedia	*dbo:Place*	200	2016-04
	foaf:Person	174	
	dbo:Organization	219	
3cixty	*lode:Events*	215	2016-09-09

Model Evaluation

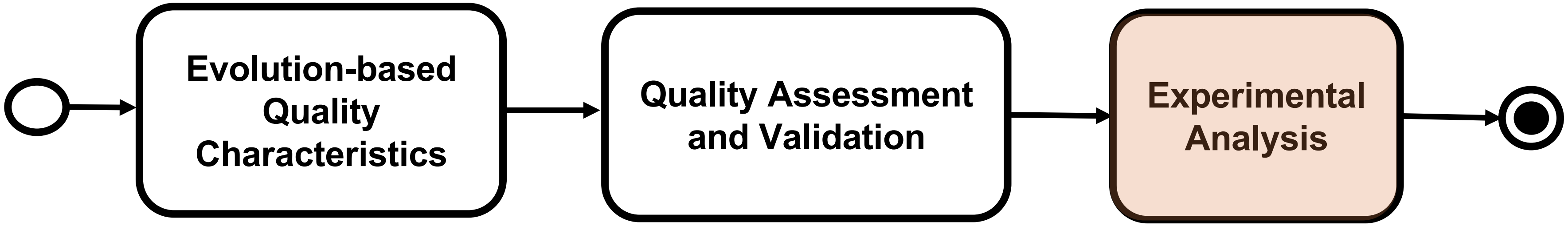
Integrity Constraints performance measures for 3cixty

Learning Algorithm	Minimum Cardinality F1 Score	Maximum Cardinality F1 Score	Range F1 Score
Random Forest	**0.91**	**0.93**	**0.91**
Multilayer Perceptron	0.81	0.81	0.90
Least Squares SVM	0.74	0.84	0.86
Naive Bayes	0.70	0.77	0.82
K-Nearest Neighbor	0.68	0.76	0.80

Integrity Constraints performance measures for DBpedia

Learning Algorithm	Minimum Cardinality F1 Score	Maximum Cardinality F1 Score	Range F1 Score
Random Forest	**0.97**	**0.98**	**0.95**
Least Squares SVM	0.97	0.90	0.89
Multilayer Perceptron	0.95	0.88	0.84
K-Nearest Neighbor	0.94	0.87	0.83
Naive Bayes	0.88	0.83	0.84

Summary of findings


Evolution-based Quality Characteristics
Quality Assessment and Validation
Experimental Analysis

RQ1: KB Evolution Analysis

RQ2: Evolution based Quality Assessment and Validation Approach

RQ3: Evaluation

Evolution Analysis to Drive Quality Assessment

❑ Causes of quality issues

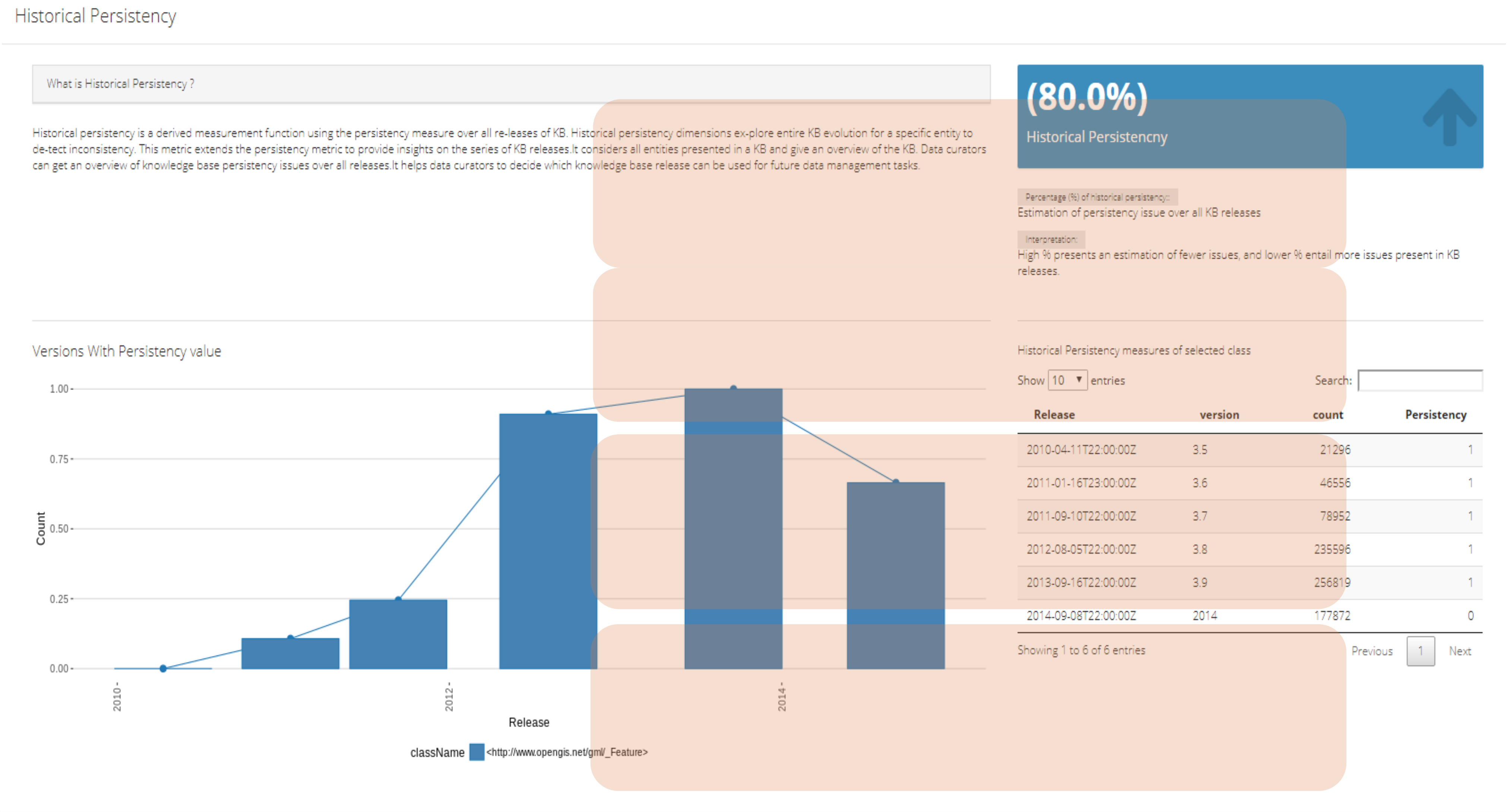
→ Errors in the data source extraction process
→ Erroneous schema presentation
→ Errors in literal values

❑ Performance

Knowledge Bases	Dataset			Performance
	Classes	Properties	Releases	Completeness Precision
DBpedia	10	4477	11	95%
3cixty	2	149	8	94%

Summary of findings

KBQ



Historical Persistency

What is Historical Persistency ?

Historical persistency is a derived measurement function using the persistency measure over all re-leases of KB. Historical persistency dimensions ex-plore entire KB evolution for a specific entity to de-tect inconsistency. This metric extends 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.It helps data curators to decide which knowledge base release can be used for future data management tasks.

(80.0%)

Historical Persistencny

Percentage (%) of historical persistency::

Estimation of persistency issue over all KB releases

Interpretation:

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

Versions With Persistency value

Historical Persistency measures of selected class

Show 10 ▾ entries Search:

Release	version	count	Persistency
2010-04-11T22:00:00Z	3.5	21296	1
2011-01-16T23:00:00Z	3.6	46556	1
2011-09-10T22:00:00Z	3.7	78952	1
2012-08-05T22:00:00Z	3.8	235596	1
2013-09-16T22:00:00Z	3.9	256819	1
2014-09-08T22:00:00Z	2014	177872	0

Showing 1 to 6 of 6 entries

Previous 1 Next

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

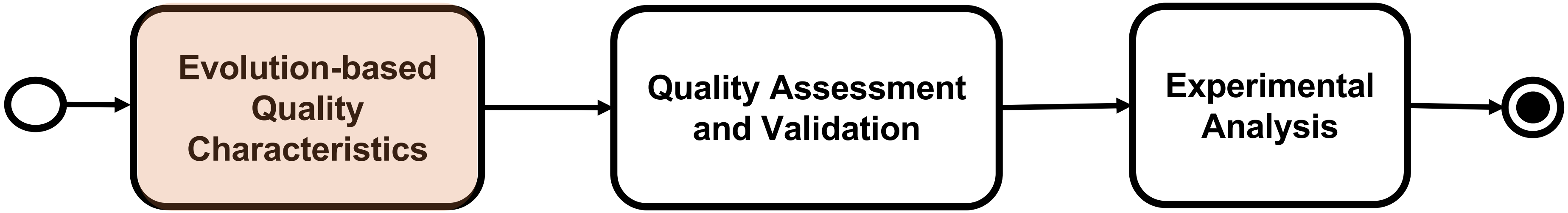
Limitations

- **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

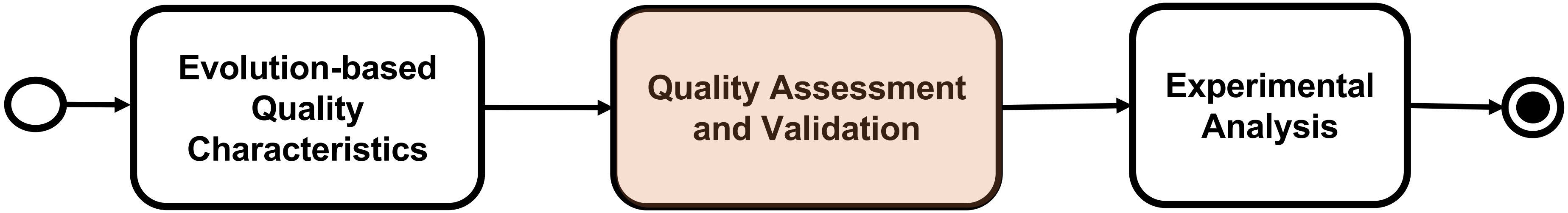
RQ1 How can we identify quality issues with respect to KB evolution?


Evolution-based Quality Characteristics
Quality Assessment and Validation
Experimental Analysis

- ❑ Proposed evolution-based measures to detect quality issues
- ❑ Introduced four evolution-based quality characteristics using summary statistics

Answers for research questions

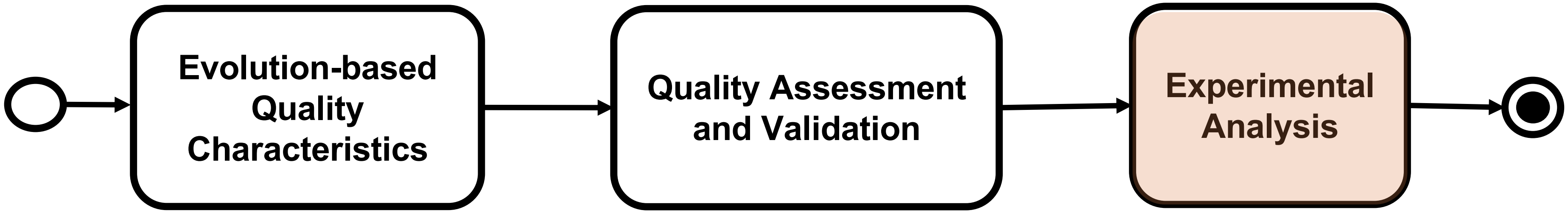
RQ2 Which quality assessment approach can be defined on top of the evolution based quality characteristics?


Evolution-based Quality Characteristics
Quality Assessment and Validation
Experimental Analysis

- ❑ 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?


Evolution-based Quality Characteristics
Quality Assessment and Validation
Experimental Analysis

- ❑ 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

❑ Journal article

- Mohammad Rashid, Giuseppe Rizzo, Marco Torchiano, Nandana Mihindukulasooriya, and Oscar Corcho, "A Quality Assessment Approach for Evolving Knowledge Bases.”, Special issue on Benchmarking Linked Data, Semantic Web Journal (2017).

- Mohammad Rashid, Giuseppe Rizzo, Marco Torchiano, Nandana Mihindukulasooriya, and Oscar Corcho, "Completeness and Consistency Analysis for Evolving Knowledge Bases.", Journal of Web Semantics (2018) [Under review with minor revisions].

❑ Conference Proceedings

- 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

- Nandana Mihindukulasooriya, Mohammad Rashid, Giuseppe Rizzo, Raúl García-Castro, Oscar Corcho, and Marco Torchiano, "RDF Shape Induction using Knowledge Base Profiling", In Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC ’18, pages 1952–1959, New York, NY, USA, 2018. ACM

Publications

- Mohammad Rashid, Giuseppe Rizzo, Nandana Mihindukulasooriya, Marco Torchiano, and Oscar Corcho, "KBQ - A Tool for Knowledge Base Quality Assessment Using Temporal Analysis", In Proceedings of Workshops and Tutorials of the 9th International Conference on Knowledge Capture (KCAP2017), Volume 2065 of CEUR Workshop Proceedings, Austin, Texas, 2017. CEUR-WS. Org.

- Rashid, Mohammad, Torchiano Marco, "A systematic literature review of open data quality in practice", In Proceedings of 2nd Open Data Research Symposium (ODRS), Madrid, Spain, 2016

❑ Other papers published during the PhD

- Rashid, Mohammad, Luca Ardito, Marco Torchiano, "Energy Consumption Analysis of Algorithms Implementations" In Proceedings of 9th International Symposium on Empirical Software Engineering and Measurement (ESEM), China, 2015

- Rashid, Mohammad, Luca Ardito, Marco Torchiano, "Energy Consumption Analysis of Image Encoding and Decoding Algorithms", In Proceedings of 4th International Workshop on Green and Sustainable Software (GREENS), 2015.

POLITECNICO DI TORINO
1859 · 1906

SOftEng
http://softeng.polito.it

ISMB
Istituto Superiore Mario Boella

Ontology Engineering Group

Thank You
Grazie

State of the art

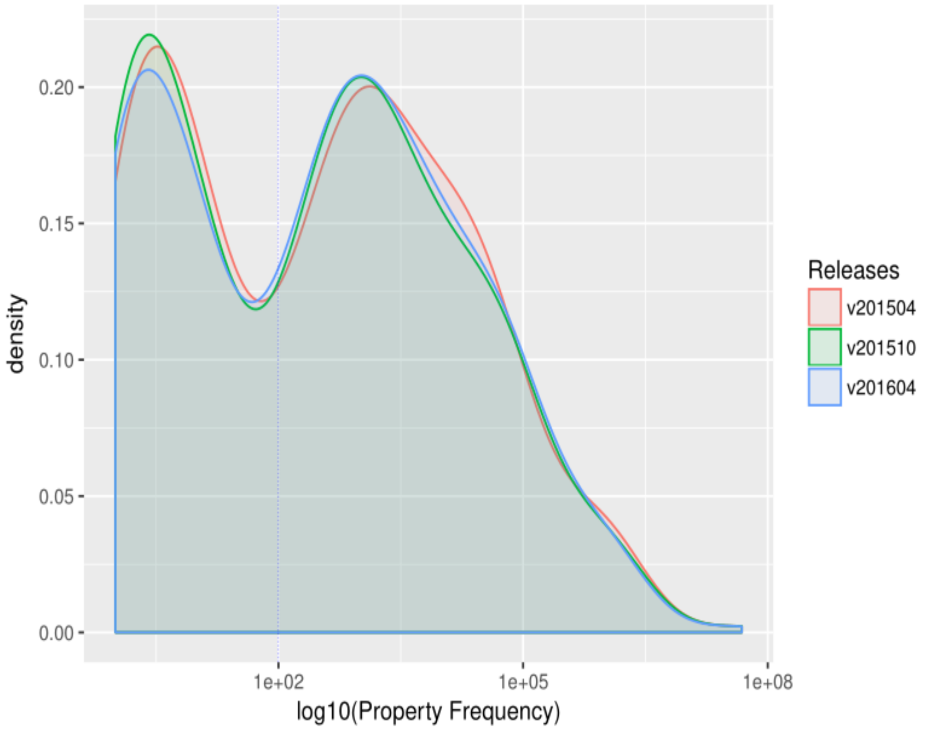
- ❑ Linked Data Dynamics[1]
- ❑ Knowledge Base Quality Assessment
  - ○ Comprehensive Surveys[2][3]
  - ○ Frameworks[4]
- ❑ Knowledge Base Validation
  - ○ Open World Assumption[5]
  - ○ Closed World Assumption[6]

1. Jürgen Umbrich, Boris Villazón-Terrazas, and Michael Hausenblas. Dataset dynamics compendium: A comparative study. In Proceedings of the First International Workshop on Consuming Linked Data (COLD2010) at the 9th International Semantic Web Conference (ISWC2010), Volume 665 of CEUR. Workshop Proceedings, Shanghai, China, 2010. CEUR-WS.
2. Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality Assessment for linked Data: A Survey. Semantic Web, 7(1):63–93, 2016.
3. Mohamed Ben Ellefi, Zohra Bellahsene, J Breslin, Elena Demidova, Stefan Dietze, Julian Szymanski, and Konstantin Todorov. RDF Dataset Profiling – a Survey of Features, Methods, Vocabularies and Applications. Semantic Web, pages 1–29, 2018.
4. Jeremy Debattista, Sören Auer, and Christoph Lange. Luzzu - A Methodology and Framework for Linked Data Quality Assessment. Journal of Data and Information Quality (JDIQ), 8(1):4:1–4:32, October 2016.
5. Jiao Tao, Evren Sirin, Jie Bao, and Deborah L McGuinness. Extending OWL with Integrity Constraints. In Haarslevand Volker, Toman David, and Weddell Grant, editors, International Workshop on Description Logics (DL), Volume 573 of CEURWorkshop Proceedings,Waterloo, Ontario, Canada, 2010. CEURWS.org.
6. Peter F. Patel-Schneider. Using Description Logics for RDF Constraint Checking and Closed-world Recognition. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI’15, pages 247–253. AAAI Press, 2015.

Consistency

Threshold Value Analysis

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:

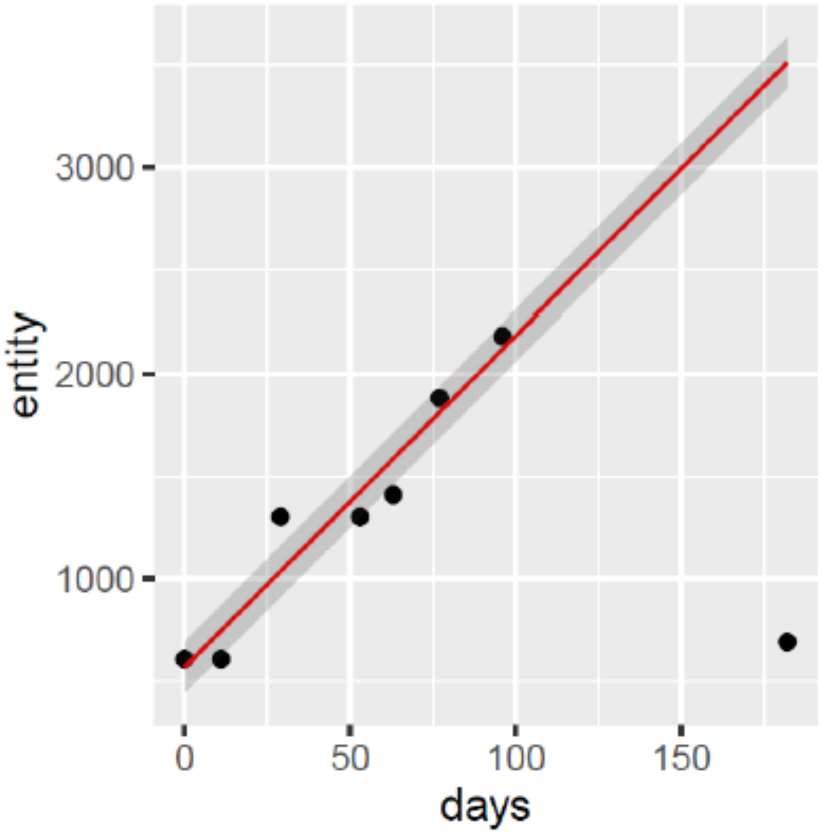
$$ND(C) = \frac{residual_n(C)}{mean(|residual_n(C)|)}$$

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

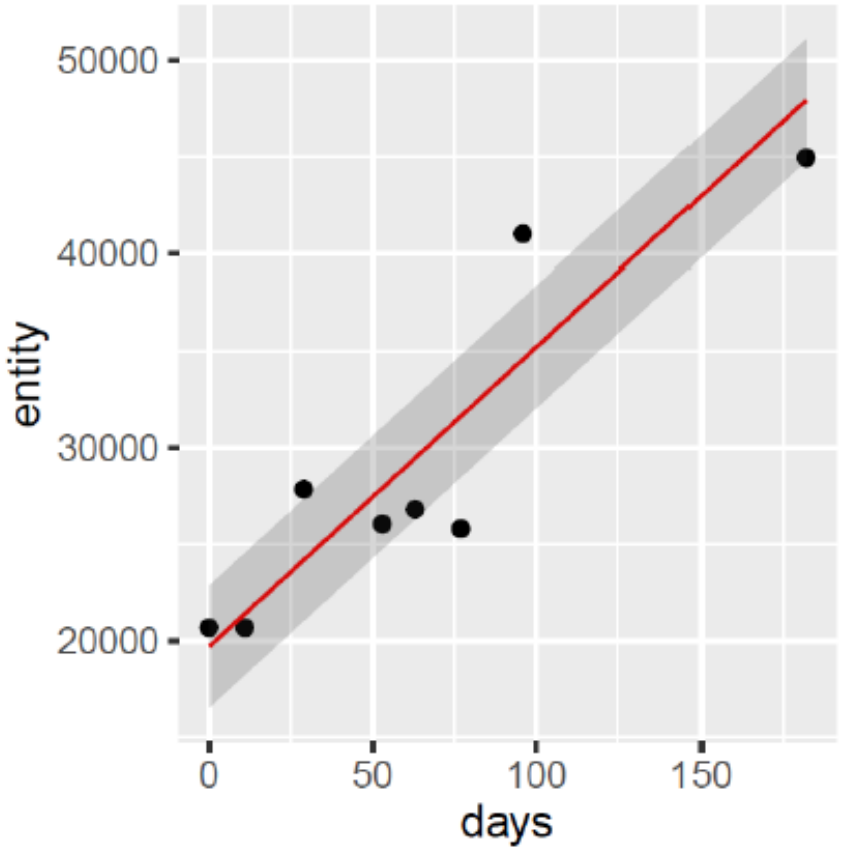
$$KB_{growth}(C) = \begin{cases} 1\ if\ ND(C)\ \geq 1 \\ 0\ if\ ND(C) < 1 \end{cases}$$

Lifespan Analysis of Evolving KBs

3cixty Knowledge Base



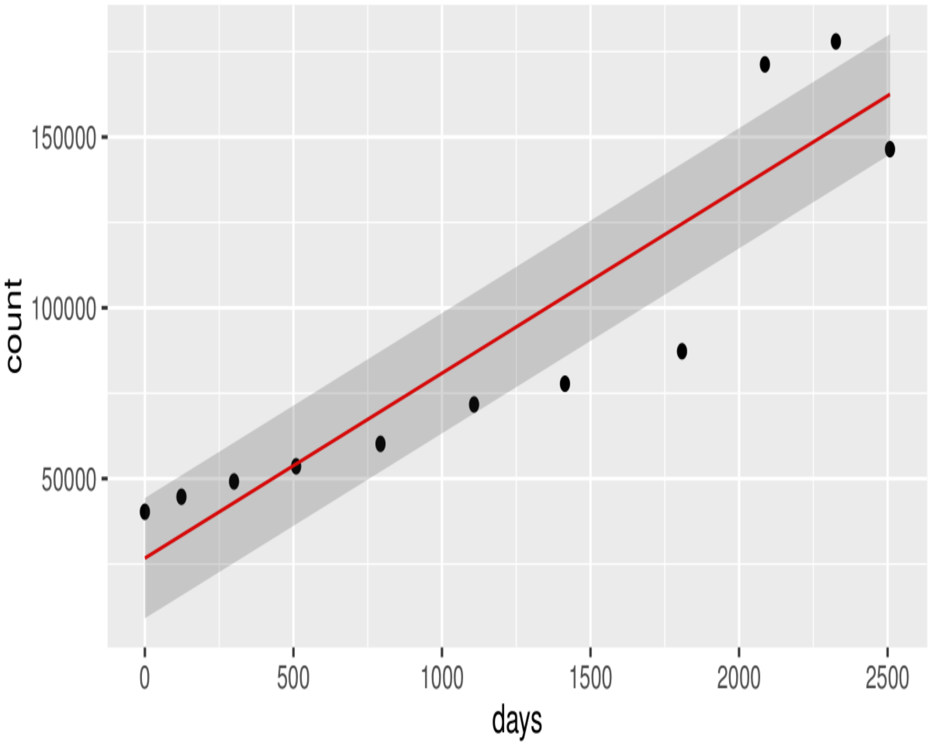
lode:Event entity type



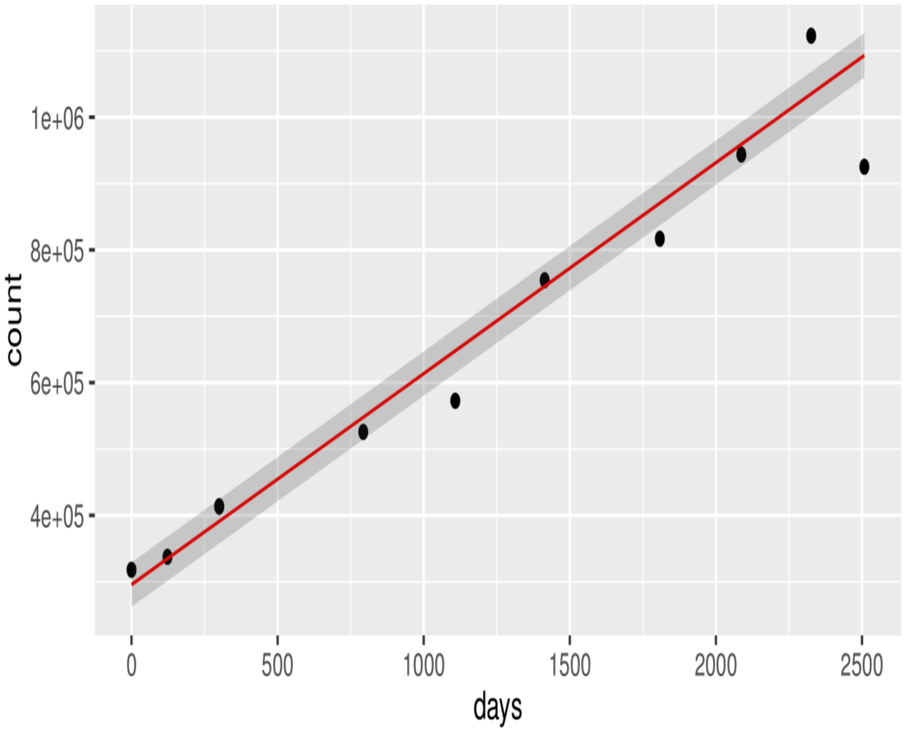
dul:Places entity type

Lifespan Analysis of Evolving KBs

DBpedia Knowledge Base



foaf:film entity type



dbo:Places entity type