Accessibility

- *Availability*:

	Component	Description
1	Name	SPARQLendpoint Accessibility of the SPARQL endpoint and the server
2	Source references	-
3	Dimension	Accessibility: Availability
4	Tags	Dataset quality, semi-automatic, objective, LD specific
5	Description	Checking whether a working SPARQL endpoint is present. No differentiation is made based on the quality of the response for this measure (either based on time or content).
6	Value type	Boolean (true indicates a successful SPARQL response)
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	True, SPARQL-endpoint X was found and responded to a query.
11	Annotation procedure	-
12	Identifier	SM-1

	Component	Description
1	Name	Accessibility of the RDF dumps
2	Source references	-
3	Dimension	Accessibility: Availability
4	Tags	Dataset Quality, Objective, Manual
5	Description	Checking whether accessible/downloadable RDF dumps are provided.
6	Value type	Boolean (True indicates an available RDF dump
7	Value Structure	-
8	Measure function	-

9	Measure elements	-
10	Example	True, an RDF dump is available on this address (make sure the link is stable) and is downloadable.
11	Annotation procedure	-
12	Identifier	SM-2

	Component	Description
1	Name	de-referencability issues
2	Source references	decker2010weaving
3	Dimension	Accessibility: Availability
4	Tags	Dataset quality, Automatic, instance, objective
5	Description	When a URI can not be properly dereferenced through a web browser and thus returns a server or client error, this is seen as a de-referencability issue. This metric depict the percentage of URIs which are dereferencable
6	Value type	float
7	Value Structure	value between 0 and 1.

8	Measure function	pseudocode:
		for all URIs in dataset
		visit URI
		if URI is error:
		add URI to broken list:
		for all URIs in broken list:
		negative annotation
		
		de-referencability issues = broken URIs / total number of URIs

9	Measure elements	Boolean - Truth value where true indicates a properly dereferencable URI Broken list - list of URIs not properly dereferencable working list - list of dereferencable URIs
10	Example	-
11	Annotation procedure	annotate all instances which are not derefencable
12	Identifier	SM-3

	Component	Description

1	Name	No structured data available
2	Source references	decker2010weaving
3	Dimension	Accessibility: Availability
4	Tags	Dataset quality, automatic, instance, objective
5	Description	When the URI is dereferenceable, it should provide some useful structured information. The detection of dead links or URIs without any metadata or no redirection using the status code 303 See Other or no code 200 OK {should cite}
6	Value type	Percentage
7	Value Structure	-
8	Measure function	value = proper URIs / all URIs
9	Measure elements	value - Percentage Proper URIs - Number of URIs that provide supporting RDF metadata or provide a redirection to a URI, describing the same concept, that does. All URIs - Total number of URIs
10	Example	0.98 = 98 / 100
11	Annotation procedure	Flag every instance not providing structured information upon dereference.
12	Identifier	SM-4

	Component	**Description**
1	Name	dereferenced back-links
2	Source references	27
3	Dimension	Accessibility: Availability
4	Tags	data quality, objective, LD specific
5	Description	detection of all local in-links or back-links: locally available triples in which the resource URI appears as an object, in the dereferenced document returned for the given resource <citation
6	Value type	float
7	Value Structure	-
8	Measure function	backlinks = select ?s ?p ?o {

		graph <graph> {
		?s ?p ?o
		}
		filter(strstarts(str(?o), "<graph>"))
		filter(strstarts(str(?o), "<graph>"))
		}
		
		SELECT (count(?s) as ?locallyKnown)
		WHERE {
		?s ?p ?o
		filter(strstarts(str(?s), "<graph>"))
		}
		value = #backlinks / #localy known triples

9	Measure elements	backlinks = number of triples where the objects is locally known <graph> is a placeholder for the graph URI
10	Example	0.80 = 160 / 200
11	Annotation procedure	SM-5
12	Identifier	<number>

	Component	**Description**
1	Name	dereferenced forward-links
2	Source references	27
3	Dimension	Accessibility: Availability
4	Tags	objective, LD specific
5	Description	Detection of all forward links: locally known triples where the local URI is mentioned in the subject [27]
6	Value type	float
7	Value Structure	-

8	Measure function	forward-links :
		select ?s ?p ?o
		WHERE {{
		GRAPH <graph> {?s ?p ?o}
		}
		FILTER(STRSTARTS(STR(?s), "<graph>"))
		}
		Percentage = #forward-links / #triples

9	Measure elements	forward-links = number of triples where the subject is locally known

10	Example	1 = 100/100
11	Annotation procedure	
12	Identifier	SM-6

	Component	**Description**
1	Name	Content Type
2	Source references	26
3	Dimension	Accessibility: Availability
4	Tags	subjective
5	Description	Meta data attached to the http request when dereferencing a URI in a browser, can include a ‘content type’. User agents can use this to quickly identify pages containing RDF.
6	Value type	String
7	Value Structure	-
8	Measure function	SPARQL: retrieve every unique URI (small sample will likely suffice) CRAWLER: Scan web pages for ‘content-types’ in headers for found URIs. SCRIPT: measure amount of content-types indicating the page contains linked data.
9	Measure elements	-
10	Example	‘text/html’
11	Annotation procedure	Flag dataset in case of ambiguous content-types (i.e. text/html) not indicating usage of linked data.
12	Identifier	SM-7

- *Licensing*:

	Component	**Description**
1	Name	machine-readable indication of a license
2	Source references	14,27

3	Dimension	Accessibility: Licensing
4	Tags	dataset quality
5	Description	Detection of the indication of a license in the VoID description or by means of dcterms:licence
6	Value type	Boolean
7	Value Structure	-

8	Measure function	PREFIX dcat: <http://www.w3.org/ns/dcat#>
		PREFIX dcterms: <http://purl.org/dc/terms/>
		
		SELECT ?s ?o
		WHERE {
		?s a dcat:Dataset ;
		dcterms:licence ?o .
		}

9	Measure elements	dcat:Dataset dcterms:licence
10	Example	True
11	Annotation procedure	
12	Identifier	SM-8

	Component	**Description**
1	Name	Human-readable indication of a license
2	Source references	14,27
3	Dimension	Accessibility: Licensing,
4	Tags	dataset quality, semi-automatic
5	Description	Detection of a license in the documentation of the dataset or its source
6	Value type	Boolean
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	True

11	Annotation procedure	
12	Identifier	SM-9

	Component	Description
1	Name	Permissible Usage
2	Source references	14
3	Dimension	Accessibility: Licensing
4	Tags	Dataset quality, Manual
5	Description	detection of license indicating whether reproduction, distribution, modification or redistribution is permitted.
6	Value type	boolean
7	Value Structure	True indicates the dataset can be reused. the extent to which this is the case and the included restrictions should be included in a note.
8	Measure function	-
9	Measure elements	-
10	Example	CC-0
11	Annotation procedure	
12	Identifier	SM-10

	Component	Description
1	Name	Copyright // Indication of attribution, Copyleft or ShareAlike
2	Source references	14
3	Dimension	Accessibility: Licensing
4	Tags	Dataset Quality, manual
5	Description	Detection of whether the reused work is attributed in the same way as specified by the author or licensor. share-alike and copyleft roughly imply that the work in

		question should be licensed similarly as referenced sources. The work in question can also specify share-alike and copyleft requirements.
6	Value type	boolean
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	CC Attribution-ShareAlike, www.example.com/dataset/
11	Annotation procedure	-
12	Identifier	SM-11

- *Interlinking*:

	Component	**Description**
1	Name	outward links //existence of links to external data providers
2	Source references	27
3	Dimension	Accessibility , Interlinking
4	Tags	Data Quality, Automatic, objective
5	Description	Detection of the existence and usage of external URIs and owl:sameAs links with external URIs.
6	Value type	Integer
7	Value Structure	-
8	Measure function	SPARQL: count every triple where the predicate is owl:sameAs, and/or, the object resides in another namespace (prefix)
9	Measure elements	-
10	Example	True
11	Annotation procedure	Flag every resource found by the sparql query
12	Identifier	SM-12

	Component	**Description**
1	Name	Instance enrichment by interlinking
2	Source references	

3	Dimension	Accessibility , Interlinking
4	Tags	instance quality
5	Description	Percentage of attributes gained by interlinking and owl:sameAs relations.
6	Value type	float
7	Value Structure	value between 0 and 1
8	Measure function	SPARQL: for every instance, interlinked (by means of owl:sameAs) with external data, measure the amount of gained triples. SCRIPT: measure the percentage of new unique triples, out of all triples.
9	Measure elements	-
10	Example	0.10
11	Annotation procedure	
12	Identifier	SM-13

	Component	**Description**
1	Name	Schema enrichment by interlinking
2	Source references	
3	Dimension	Accessibility , Interlinking
4	Tags	schema quality
5	Description	percentage of classes and properties reused from other sources
6	Value type	float
7	Value Structure	-

8	Measure function	SPARQL:
		Select DISTINCT ?p ?o
		FROM NAMED <graph>
		WHERE {{
		?s ?p ?o .
		}
		MINUS{
		?s ?p ?o .
		filter (strstarts(str(?p), "https://data.pdok.nl/cbs/2015/vocab/"))
		}

		MINUS{
		?s ?p ?o .
		?o a rdfs:Class .
		filter (strstarts(str(?o),
		"https://data.pdok.nl/cbs/2015/vocab/"))
		}
		}
		SCRIPT:
		measure the percentage of reused classes and properties out of all used classes and properties.

9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-14

- *Security*:

	Component	**Description**
1	Name	access to data is secure
2	Source references	67
3	Dimension	Accessibility, Security
4	Tags	dataset quality, automatic
5	Description	use of login credentials or use of SSL or SSH
6	Value type	Boolean
7	Value Structure	-
8	Measure function	SPARQL: retrieve a sample of URIs SCRIPT: check whether URIs start with HTTPS or HTTP.
9	Measure elements	-
10	Example	True, Usage of SSL
11	Annotation procedure	
12	Identifier	SM-14

	Component	Description
1	Name	Data is of proprietary nature
2	Source references	67
3	Dimension	Accessibility, Security
4	Tags	Dataset Quality
5	Description	Data owner allows access only to certain users
6	Value type	Boolean
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	True, An account is needed to access the data
11	Annotation procedure	
12	Identifier	SM-15

- *Performance*:
(Robustness)

	Component	Description
1	Name	usage of hash-URIs
2	Source references	14
3	Dimension	Accessibility, Performance
4	Tags	dataset quality, automatic
5	Description	Checking for usage of slash-URIs where large amounts of data is provided. The alternative would be 'hash-URIs, where the symbol before the last word is a hash. Hash-URIs have certain advantages when dealing with larger datasets in terms of performance.
6	Value type	Boolean
7	Value Structure	-
8	Measure function	SPARQL: retrieve a sample of URIs SCRIPT: check whether URIs uses a hash or slash.
9	Measure elements	-

10	Example	True: example.com/resource#item1
11	Annotation procedure	
12	Identifier	SM-15

-

	Component	Description
1	Name	Latency
2	Source references	14, 4
3	Dimension	Accessibility, Performance
4	Tags	dataset quality, automatic
5	Description	Latency is the delay between the submission of a request by a user, and the reception of a response from the system.
6	Value type	Float
7	Value Structure	time in seconds
8	Measure function	latency = end-time - start-time
9	Measure elements	start-time - time of submission of a request end-time - time of reception of a response
10	Example	0.76 seconds
11	Annotation procedure	-
12	Identifier	SM-16

-

	Component	Description
1	Name	high throughput
2	Source references	
3	Dimension	Accessibility, Performance
4	Tags	dataset quality
5	Description	Throughput of the sources measures the amount of HTTP requests answered per second
6	Value type	integer
7	Value Structure	-

8	Measure function	-
9	Measure elements	-
10	Example	6 http request per second
11	Annotation procedure	-
12	Identifier	SM-18

-

	Component	**Description**
1	Name	scalability of a data source
2	Source references	14
3	Dimension	Accessibility, Performance
4	Tags	
5	Description	Detection of whether the time to answer an amount of ten requests divided by ten is not longer than the time it takes to answer one request
6	Value type	boolean
7	Value Structure	-
8	Measure function	if a / 10 <= b : True
9	Measure elements	a = time taken by server to answer ten HTTP request b = time taken by server to answer one HTTP request both a and b are in seconds.
10	Example	b= 0,012 a=0.1 0.1/10 = .01 0.01 <= 0.012 True
11	Annotation procedure	
12	Identifier	SM-19

	Component	**Description**
1	Name	result size restriction
2	Source references	50
3	Dimension	Accessibility, Performance

4	Tags	dataset quality, automatic
5	Description	time-out errors can occur when the SPARQL query attempts to retrieve ‘too’ large amounts of data. Data providers can impose such restriction as well. The value for this metric is zero if no restriction is imposed.
6	Value type	xsd:dayTimeDuration
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	“PT5M”^^xsd:dayTimeDuration
11	Annotation procedure	-
12	Identifier	SM-20

Intrinsic

- *Accuracy*:

	Component	**Description**
1	Name	Detection of outliers
2	Source references	5 @article{knorr2000distance, title={Distance-based outliers: algorithms and applications}, author={Knorr, Edwin M and Ng, Raymond T and Tucakov, Vladimir}, journal={The VLDB Journal—The International Journal on Very Large Data Bases}, volume={8}, number={3-4}, pages={237--253}, year={2000}, publisher={Springer-Verlag New York, Inc.} } @article{akoglu2015graph, title={Graph based anomaly detection and description: a survey}, author={Akoglu, Leman and Tong, Hanghang and Koutra, Danai}, journal={Data Mining and Knowledge Discovery}, volume={29}, number={3}, pages={626--688}, year={2015}, publisher={Springer}

		}
3	Dimension	Intrinsic, Accuracy
4	Tags	instance quality
5	Description	by using distance-based, deviations-based and distribution-based methods
6	Value type	integer
7	Value Structure	-
8	Measure function	SCRIPT: for i in results_q2 : list_of_outliers.append(detect_outliers(select_property(results_q1, i))) return list_of_outliers

9	Measure elements	● query ='''
		CONSTRUCT {?p ex:hasValue ?o }
		WHERE {
		?s ?p ?o
		}
		FILTER(isLiteral(?o)
		
		'''
		● query2 ='''
		SELECT DISTINCT ?p
		WHERE {
		?s ?p ?o
		}
		'''
		● list_of_outliers = list of triples containing outlier values
		● detect_outlier = function to detect outliers
		● select_property(graph, property = fires a sparql query to retrieve all triples with a specific property as the subject from a graph.

10	Example	-
11	Annotation procedure	flag every instances considered an outlier
12	Identifier	SM-21

	Component	**Description**
1	Name	inaccurate values
2	Source references	15, 68

3	Dimension	Intrinsic, Accuracy
4	Tags	instance quality
5	Description	onu approach: By using functional dependencies rules between the values of two or more different predicates. a functional dependency is a constraint between different properties of an instance, in the case of linked data. the underlying concept is that if one value for property X of an instance always co-occurs with a precisely one value for property Y, then it can be said that X determines the value of Y.
6	Value type	integer
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	-
11	Annotation procedure	flag every instance considered inaccurate
12	Identifier	SM-22

	Component	**Description**
1	Name	inaccurate facts
2	Source references	39
3	Dimension	Intrinsic, Accuracy
4	Tags	instance quality
5	Description	check for inconsistencies by comparing the data to that of other sources. The metric measures the percentage of facts deemed correct.
6	Value type	integer
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	-
11	Annotation procedure	flag every triple considered inaccurate
12	Identifier	SM-23

	Component	**Description**

1	Name	literals incompatible with datatype range
2	Source references	26
3	Dimension	Intrinsic, Accuracy
4	Tags	Instance Quality
5	Description	detection of ill-typed literals which do not abide by the lexical syntax for their respective datatype.
6	Value type	integer
7	Value Structure	
8	Measure function	Reasoner can pick this up, otherwise: SPARQL: retrieve every triple where the object is a literal, where, for each specific datatype, the literal adheres to a specific regular expression.
9	Measure elements	-
10	Example	errors can include, boolean value stored as an integer misspelling of xsd types.
11	Annotation procedure	flag triples where literal are ill-typed
12	Identifier	SM-24

	Component	**Description**
1	Name	malformed datatype literals
2	Source references	26
3	Dimension	Intrinsic, Accuracy
4	Tags	instance quality
5	Description	Measures the amount of literals with an unknown datatype.
6	Value type	integer
7	Value Structure	-

8	Measure function	SPARQL:
		SELECT *
		WHERE {
		GRAPH <graphuri> {
		SELECT ?malformed ?Literal_count (?malformed / ?Literal_count as ?malformeddatatypeliterals) WHERE {
		{
		SELECT (count(?o) as ?malformed) WHERE {
		?s?p?o .

		FILTER ((datatype(?o)) = '')
		}}
		UNION {
		SELECT (count(?o) as ?Literal_count) WHERE
		{
		?s?p?o .
		FILTER (!isURI(?o))
		}
		}
		}
		}
		}

9	Measure elements	-
10	Example	-
11	Annotation procedure	flag triples where literals are not typed.
12	Identifier	SM-25

	Component	**Description**
1	Name	Attribute Accuracy
2	Source references	
3	Dimension	Intrinsic, Accuracy
4	Tags	Instance quality
5	Description	degree in which measurements are correct, or how precise the measurement was conducted. For instance with Geo spatial data, measurements can be done in meters or millimeters.
6	Value type	float
7	Value Structure	value between 0 and 1
8	Measure function	the property describing attribute accuracy has to be manually found and used in the following SPARQL query. SELECT DISTINCT ?s ?o WHERE { ?s <property> ?o} list all triples with found property
9	Measure elements	
10	Example	ex:example ex:sourceaccuracy “1.0”^^xsd:float . ex:sourceaccuracy skos:note “measured in meters”@en .
11	Annotation procedure	

12	Identifier	SM-26

- *Consistency*:

	Component	Description
1	Name	entities as members of disjoint classes
2	Source references	14 26
3	Dimension	Intrinsic, Consistency
4	Tags	semi-automatic, instance quality
5	Description	entities (inferred to be) part of two disjoint classes. This metric measures the percentage of entities where this is the case
6	Value type	integer
7	Value Structure	-

8	Measure function	Reasoner could pick this up.
		SPARQL:
		SELECT *
		WHERE {
		?s a ?class1,?class2 .
		?class1 owl:disjointWith ?class2 .
		UNION {
		?s a ?class1,?class2 .
		?class2 owl:disjointWith ?class1 .
		}
		}

9	Measure elements	-
10	Example	entity of type foaf:Person and foaf:Organization
11	Annotation procedure	flag entities which are members of disjoint classes
12	Identifier	SM-27

	Component	Description
1	Name	undefined classes and properties
2	Source references	26
3	Dimension	Intrinsic, Consistency
4	Tags	schema quality
5	Description	Detection of classes and properties without a formal

		definition.
6	Value type	Integer
7	Value Structure	

8	Measure function	SPARQL:
		SELECT ?class
		WHERE {
		?s rdf:type ?class
		FILTER NOT EXIST {?class rdf:type rdfs:Class}
		}
		
		SELECT ?property
		WHERE {
		?s ?property ?o
		FILTER NOT EXIST {?property rdf:type rdfs:Property}
		}

9	Measure elements	-
10	Example	20% of the used classes was not formally defined.
11	Annotation procedure	
12	Identifier	SM-27

	Component	**Description**
1	Name	misplaced classes or properties
2	Source references	26
3	Dimension	Intrinsic, Consistency
4	Tags	schema quality
5	Description	detecting triples where the predicate is filled with an URI defined as a class or where the object position is filled by a property while the predicate is a rdf:type. These errors can be found with the help of a reasoner.
6	Value type	measure
7	Value Structure	percentage, list

8	Measure function	SELECT * WHERE {
		?s ?p ?o .
		?p a rdfs:Class.
		UNION {
		?s rdf:type ?o .
		?o a rdf:property .
		}

		}
9	Measure elements	-
10	Example	ex:Class ex:hasProperty ex:literal . ex:hasProperty rdf:type ex:Class .
11	Annotation procedure	-
12	Identifier	SM-28

	Component	**Description**
1	Name	property type misuse of owl:datatypeProperty or owl:objectProperty
2	Source references	26
3	Dimension	Intrinsic, Consistency
4	Tags	schema quality
5	Description	Properties of a resource can refer to either a literal value, or to a URI/blank-node, an owl:DatatypeProperty and a n owl:ObjectProperty, respectively.
6	Value type	integer
7	Value Structure	-

8	Measure function	SELECT *
		WHERE {
		{
		?p a owl:datatypeProperty .
		?o a xsd:anyURI .
		?s ?y ?z .
		}
		UNION {{
		?s a owl:ObjectProperty ;
		?p ?o.}
		FILTER (datatype(?z) != xsd:anyURI)}
		}

9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-29

	Component	**Description**
1	Name	Use of members of owl:DeprecatedClass or owl:DeprecatedProperty

2	Source references	26
3	Dimension	Intrinsic, Consistency
4	Tags	schema quality
5	Description	usage of deprecated classes and properties is measured by retrieving resources of type owl:DeprecatedClass and owl:DeprecatedProperty.
6	Value type	boolean
7	Value Structure	-

8	Measure function	select ?x ?y ?z {
		?x ?y ?z .
		{ { ?y a owl:DeprecatedProperty } union { ?x a owl:DeprecatedClass } union { ?z a owl:DeprecatedClass } }
		}

9	Measure elements	-
10	Example	
11	Annotation procedure	
12	Identifier	SM-30

	Component	**Description**
1	Name	void values provide a blacklist for void values
2	Source references	26
3	Dimension	Intrinsic, Consistency
4	Tags	schema quality
5	Description	usage owl:Inverse-functionalProperty implies a resource can be uniquely identified by the value of the property. When this usage results in faulty values (void values), it will no longer uniquely identify a resource, but each resource with a similar void value, resulting in an explosion of owl:sameAs links.
6	Value type	integer
7	Value Structure	
8	Measure function	SPARQL: count every instance with an inverse-functional property which is not unique

9	Measure elements	
10	Example	
11	Annotation procedure	flag every instance with an inverse-functional property which is not unique
12	Identifier	SM-31

	Component	**Description**
1	Name	ontology hijacking
2	Source references	26
3	Dimension	Intrinsic, Consistency
4	Tags	schema quality, automatic
5	Description	detection of the redefinition by third parties of classes/properties such that reasoning over data using those external terms is affected.
6	Value type	measure
7	Value Structure	boolean, affected resource(s)

8	Measure function	SELECT * WHERE {
		?s ?p ?o .
		FILTER NOT EXIST {
		FILTER (STRSTARTS(STR(?s), "<graph namespace>"))
		}
		}

9	Measure elements	-
10	Example	ex1:class a ex:class
11	Annotation procedure	flag all resources hijacked
12	Identifier	SM-32

	Component	**Description**
1	Name	property consistency
2	Source references	50
3	Dimension	Intrinsic, Consistency
4	Tags	Schema quality
5	Description	Measures the number of instances with identical values for different attributes. When two or more properties of an

		instance contains the same information, but with different naming, it could be redundant.
6	Value type	float
7	Value Structure	-

8	Measure function	for each instance :
		check all attributes
		if any attributes contain identical literals/URIs
		count the redundancy and list triples.
		property conciseness = a / b
		
		SELECT * WHERE {
		?s ?p ?o;
		?p2 ?o2.
		FILTER (?o = ?o2)
		}

9	Measure elements	a = number of redundant attributes b = total number of attributes
10	Example	ex:instance ex:islocate ex:aHouse . ex:instance ex:inhabits ex:aHouse . Both properties point to the same IRI, hence ex:hasOtherProperty only provides already known information.
11	Annotation procedure	-
12	Identifier	SM-33

	Component	**Description**
1	Name	ambiguous instances
2	Source references	42
3	Dimension	Intrinsic, Consistency
4	Tags	Instance Quality
5	Description	Instances mapping to more than one real-world object, resulting in multiple possible interpretations
6	Value type	integer
7	Value Structure	
8	Measure function	SCRIPT: 1 - (a / b)
9	Measure elements	a = # of ambiguous instances

		b = # number of instances
10	Example	-
11	Annotation procedure	flag every instance considered ambiguous
12	Identifier	SM-34

- *Conciseness*:

	Component	Description
1	Name	duplicate instance
2	Source references	
3	Dimension	Intrinsic, Conciseness
4	Tags	instance quality
5	Description	measures the amount of duplicate instances in a dataset
6	Value type	integer
7	Value Structure	-
8	Measure function	SPARQL: sparql query to see whether two instances share all properties
9	Measure elements	-
10	Example	-
11	Annotation procedure	flag every instance considered a duplicate, including instances which have a duplicate
12	Identifier	SM-35

Contextual

- *Completeness*:

	Component	Description
1	Name	schema similarity Schema completeness
2	Source references	4,15,50
3	Dimension	Contextual, Completeness
4	Tags	Schema quality

5	Description	no. of classes and properties represented / total no. of classes and properties. for the purpose of interlinking, a more interesting form would take into account the intersection of the schema of both datasets as well as the parts only present in one of the datasets and their respective size.
6	Value type	Measure
7	Value Structure	(d1, intersection, d2)
8	Measure function	amount=count(dataset, feature) Percentage = amount(dataset1)/amount(dataset2) overlap =amount(intersection(dataset1, dataset2))/amount(dataset2)
9	Measure elements	amount count - a function to obtain the number of occurrences of, for this research, entities or properties. percentage percentage measures the relative size of both datasets. overlap intersection - The intersection either has to be done manually, where an expert determines the overlap in properties and/or entities (while entities might not be feasible in larger datasets) or by means of matching techniques.
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-36

-

	Component	**Description**
1	Name	Schema completeness
2	Source references	
3	Dimension	Contextual, Completeness
4	Tags	schema quality, semi-automatic
5	Description	Usage of external (and established) vocabularies improves the machine-readability and reusability of a schema. They also provide an overview of properties and classes relevant for a given domain and scope. This metric measures the extent to which the vocabulary is exploited by taking the percentage of distinct classes and properties of an external vocabulary reused in the dataset.
6	Value type	float

7	Value Structure	value between 0 and 1
8	Measure function	SPARQL: for every reused vocabulary/ontology, count the amount of distinct properties/classes from that vocabulary/ontology. SCRIPT: measure the percentage of reused properties/classes out of the total amount of properties/classes defined in the respective vocabulary/ontology
9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-37

	Component	**Description**
1	Name	property completeness
2	Source references	4, 14
3	Dimension	
4	Tags	Instance quality
5	Description	Property completeness entails every entity having the properties described in the schema known.
6	Value type	float
7	Value Structure	-
8	Measure function	for each entity c = a / b total += c total / d
9	Measure elements	a= number of properties for the respective entity b= number of properties specified in the schema for the respective entity d = total number of entities
10	Example	96% This implies that 96% of the expected number of property values were found.
11	Annotation procedure	flag every instance considered not complete
12	Identifier	SM-38

	Component	Description
1	Name	population completeness
2	Source references	4, 15, 26, 50
3	Dimension	Contextual, Completeness
4	Tags	Dataset quality
5	Description	This metric measures the percentage of real-world objects represented in the dataset. attaching a number to the amount of real-world object is a difficult task. In some cases it can be assumed that the primary dataset, approximately, contains the total amount of real-world objects. in this case the total amount of objects in the measure function is known. In other cases this number can be obtained by consulting other trusted data sources.
6	Value type	float
7	Value Structure	-
8	Measure function	c = a / b
9	Measure elements	a = objects represented in the dataset b = real world objects in total
10	Example	0.25 The secondary dataset contained 25% of the total amount of real world object known. the total amount was found by consulting the primary dataset.
11	Annotation procedure	
12	Identifier	SM-39

	Component	Description
1	Name	interlinking completeness/clustering coefficient
2	Source references	22
3	Dimension	Contextual, Completeness
4	Tags	dataset quality
5	Description	Interlinking completeness measures the amount of instances that are interlinked in respect to the total amount of instances. by measuring this clustering coefficient one can assess the local cohesiveness of the dataset.
6	Value type	float
7	Value Structure	-
8	Measure function	c=a/b

9	Measure elements	a=#of interlinks b=total # of instances
10	Example	.5
11	Annotation procedure	-
12	Identifier	SM-40

- *Amount-of-data*:

	Component	**Description**
1	Name	Amount of %%s%%
2	Source references	10
3	Dimension	contextual, Amount-of-Data
4	Tags	Dataset quality/Schema quality, semi-automatic, objective, (Linked Data specific)
5	Description	The amount of triples in the dataset. Similar metrics Amount of Classes and Amount Of properties are also contained in this table.
6	Value type	Integer
7	Value Structure	-

8	Measure function	- There are several ways of obtaining the amount of triples, performing a line count on a nquads file, retrieving the void:triples property attached to the dataset or by means of a COUNT SPARQL query.
		
		● void:
		
		PREFIX void: <http://rdfs.org/ns/void#>
		SELECT ?y ?x
		WHERE {
		?y a void:Dataset ;
		void:triples ?x .
		}
		
		● count:
		
		SELECT (COUNT(?s) as ?count)
		#replace ?s with ?p or ?o for properties and classes
		#respectively.
		WHERE {
		?s ?p ?o .
		}

9	Measure elements	-

10	Example	SELECT (COUNT(*) as ?count)
		WHERE {
		  ?s ?p ?o .
		}
		result: 1000

11	Annotation procedure	
12	Identifier	SM-41

	Component	**Description**
1	Name	coverage
2	Source references	
3	Dimension	Contextual, Amount of data
4	Tags	Semi-automatic, Contextual, Data quality, Instance quality
5	Description	Coverage measures the scope of the dataset as the number of entities represented and the descriptiveness as the number of properties known for each entity.
6	Value type	Measure
7	Value Structure	Measure(datasets, amount, percentage, overlap)
8	Measure function	Coverage = Scope | Descriptiveness Scope = Some form of a count of prov:entities or other features. descriptiveness - for each of the found entities the number of attached properties are counted. The coverage then entails the number of entities and the average number of properties
9	Measure elements	Scope - number of entities Descriptiveness - average number of properties per entity
10	Example	The dataset contains 1000 entities with 6 properties on average. Coverage = 1000 | 6
11	Annotation procedure	-
12	Identifier	SM-42

- *Relevancy*:

	Component	Description
1	Name	usage of meta-information attributes
2	Source references	4
3	Dimension	Contextual, Relevancy
4	Tags	dataset quality, manual
5	Description	Measures the number of meta-data properties assigned to the dataset.
6	Value type	integer
7	Value Structure	-
8	Measure function	SPARQL: count the number of properties assigned to the instance describing the dataset.
9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-43

	Component	Description
1	Name	Similarity of meta-data
2	Source references	4
3	Dimension	Contextual, Relevance
4	Tags	Dataset Quality semi-automatic
5	Description	by measuring the cosine similarity of ‘descriptive’ metadata (i.e. Description, theme, keywords/tags) between the two datasets, one can roughly assess whether one is relevant for the other.
6	Value type	float
7	Value Structure	-

8	Measure function	$sim(d,q) = \cos(\vec{v}(d), \vec{v}(q)) = \frac{\vec{v}(d) \cdot \vec{v}(q)}{\|\vec{v}(d)\| \cdot \|\vec{v}(q)\|} = \frac{\sum_{i=1}^{\|V\|} d_i \cdot q_i}{\sqrt{\sum_{i=1}^{\|V\|} d_i^2} \cdot \sqrt{\sum_{i=1}^{\|V\|} q_i^2}}$
9	Measure elements	$\sum_{i=1}^{\|V\|} d_i \cdot q_i$ - the sum of the product of the frequency of each term i in each dataset. $\sqrt{\sum_{i=1}^{\|V\|} d_i^2} \cdot \sqrt{\sum_{i=1}^{\|V\|} q_i^2}$
10	Example	Metadata properties, for each dataset, as a whole are considered a document. D1 Vector space model represents each document and each query as a vector of numbers. D2 A document vector contains term frequencies of all terms in that document. Q document vector $cos(d_1, q) = \frac{1+2}{\sqrt{2^2+2^2+10} \cdot \sqrt{2}} = \frac{3}{\sqrt{18}\sqrt{2}}$ $cos(d_2, q) = \frac{2+1}{\sqrt{2^2+10} \cdot \sqrt{2}} = \frac{3}{\sqrt{14}\sqrt{2}}$
11	Annotation procedure	-
12	Identifier	SM-44

Representation

- *Representational-conciseness*:

	Component	Description
1	Name	Keeping URIs short
2	Source references	27
3	Dimension	Representational, Representational-conciseness
4	Tags	Dataset Quality, semi-automatic
5	Description	URIs should be readable and concise. Making sure there is no redundant information is present in the URI makes it more easily understood.
6	Value type	boolean
7	Value Structure	-

8	Measure function	-
9	Measure elements	-
10	Example	The following URI is concise and contains no redundant information and would score 'True' for this metric. https://example.com/ontology#chair The following URI contains alot of redundant informations such as the publication date. This URI would score "false for this metric. https://example.com/resource/paper/17-05-2001#the-semantic-web/
11	Annotation procedure	flag instances where its URI is considered not short.
12	Identifier	SM-45

	Component	**Description**
1	Name	Prolix RDF features no use of prolix RDF features
2	Source references	27
3	Dimension	Representational, representational-conciseness
4	Tags	schema quality
5	Description	measure the usage of too lengthy rdf features such "rdf reification, rdf containers and rdf collections. it is advised not to use these since it make the use of SPARQL more cumbersome and since the semantics behind these features are unclear.
6	Value type	integer
7	Value Structure	#number of prolix rdf features
8	Measure function	COUNT SPARQL query for the following properties and classes. rdfs:member rdf:_n (n 2 N), df:subject, rdf:predicate, rdf:object the class rdf:List. rdf:Statement; rdf:Alt, rdf:Bag, rdf:Seq

		rdfs:Container rdf:first, rdf:rest,
9	Measure elements	-
10	Example	12 occurences of prolix rdf features
11	Annotation procedure	flag instances where its properties use prolic rdf features.
12	Identifier	SM-46

- *Representational-consistency*:

	Component	**Description**
1	Name	re-use existing vocabularies
2	Source references	14
3	Dimension	Representation, Representational-consistency
4	Tags	schema quality
5	Description	The usage of established vocabularies boosts the understandability (and uniformity) of the data schemas and thereby the ease of reuse.
6	Value type	Measure
7	Value Structure	The measure contains a list of every import.
8	Measure function	By looking up the URLs preceded by the PREFIX statement, one can view every import used while describing data
9	Measure elements	-
10	Example	@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> . @prefix owl: <http://www.w3.org/2002/07/owl#> . @prefix dcterms: <http://purl.org/dc/terms/> .
11	Annotation procedure	
12	Identifier	SM-47

	Component	**Description**
1	Name	Vocabulary similarity
2	Source references	-

3	Dimension	Representation, Representational-consistency
4	Tags	subjective schema quality
5	Description	Interlinking datasets that use different notations for identical information can convolute the process. If both datasets use a vocabulary, it can provide junctions where the two datasets can be interlinked.
6	Value type	Measure
7	Value Structure	The measure contains a list of every import shared by both datasets.
8	Measure function	By looking up the URLs preceded by the PREFIX statement, one can view every import used while describing data. By performing an intersection
9	Measure elements	-
10	Example	@prefix skos: <http://www.w3.org/2004/02/skos/core#> . @prefix dcat: <http://www.w3.org/ns/dcat#> . @prefix foaf: <http://xmlns.com/foaf/0.1/> .
11	Annotation procedure	
12	Identifier	SM-48

- *Understandability*:

	Component	**Description**
1	Name	human-readable labelling of classes, properties and entities
2	Source references	13
3	Dimension	representational, understandability
4	Tags	data quality
5	Description	percentage of resources containing a human readable label. For instance by means of an rdfs:label or rdfs:comment. Other vocabularies also offer labelling functions such as skos:preflabel or skos:note.
6	Value type	float
7	Value Structure	value between - and 1
8	Measure function	SPARQL: count all instances containing a label. SCRIPT: measure percentage of instances with a label out of all instances.

9	Measure elements	sparql results = integer percentage = sparql results / total entities
10	Example	100%
11	Annotation procedure	annotate instances without a label.
12	Identifier	SM-49

	Component	**Description**
1	Name	indication of one or more exemplary URI
2	Source references	14
3	Dimension	representational, understandability
4	Tags	dataset quality, manual
5	Description	detecting whether the URI patterns are provided
6	Value type	boolean
7	Value Structure	boolean value + found exemplary URIs
8	Measure function	-
9	Measure elements	-
10	Example	URI = <<publisher>/<type>#<identifier>>
11	Annotation procedure	
12	Identifier	SM-50

	Component	**Description**
1	Name	indication of an exemplary SPARQL query
2	Source references	14
3	Dimension	representational, understandability
4	Tags	dataset quality
5	Description	As with the exemplary URIs, exemplary SPARQL queries provide insight as to how the dataset can be queried.
6	Value type	Boolean
7	Value Structure	boolean + exemplary queries
8	Measure function	-

9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-51

	Component	**Description**
1	Name	prefix indication indication of the vocabularies used in the dataset
2	Source references	
3	Dimension	representational, understandability
4	Tags	dataset quality, manual
5	Description	Metric X checked the usage of other vocabularies. With the dataset, an indicaion of these external vocabularies should be provided to enhance understanding of the dataset.
6	Value type	Boolean
7	Value Structure	Boolean + URL to indication
8	Measure function	-
9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-52

	Component	**Description**
1	Name	community provision of message boards and mailing lists
2	Source references	14
3	Dimension	representational, understandability
4	Tags	dataset quality, manual
5	Description	Providing message boards and mailing list can be used to actively improve, and help users understand and query the dataset. This metric measure the presence and use of such mechanism. A neglected message board might indicate that the dataset is no longer updated. The value of this metric is True only when message boards and/or mailing list are provided and used.

6	Value type	boolean
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-53

- *Interpretability*:

	Component	**Description**
1	Name	interpretability of terms
2	Source references	4
3	Dimension	Representational, interpretability
4	Tags	data quality, subjective, semi-automatic
5	Description	use of various schema languages to provide definitions for terms. providing definitions for used terms helps users to interpret the information.
6	Value type	boolean
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	True, a glossary of terms is provided at <URL>
11	Annotation procedure	-
12	Identifier	SM-54

	Component	**Description**
1	Name	interpretability of data
2	Source references	
3	Dimension	Representational, interpretability
4	Tags	data quality, subjective
5	Description	Detect the use of appropriate language, symbols, units

		and clear definitions. For instance, the use of abbreviations can make the dataset highly understandable, but its interpretation is limited to those who know the abbreviations.
6	Value type	boolean
7	Value Structure	-
8	Measure function	SPARQL: retrieve used languages, symbols, units and a sample of definitions.
9	Measure elements	-
10	Example	8 - dateTime property missed day and month values even though they would be beneficiary
11	Annotation procedure	
12	Identifier	SM-55

	Component	**Description**
1	Name	blank nodes misinterpretation of missing values
2	Source references	
3	Dimension	Representational, interpretability
4	Tags	data Quality
5	Description	Blank nodes are local to a given node and can only be locally referenced (not globally unique). This also impedes interlinking as no external rdf can link to a blank node.
6	Value type	percentage
7	Value Structure	-
8	Measure function	percentage = blank nodes / total URIs
9	Measure elements	-
10	Example	-
11	Annotation procedure	Flag every blank node
12	Identifier	SM-56

- *Versatility*:

	Component	**Description**

1	Name	Provision of the data in different serialization formats
2	Source references	14
3	Dimension	Representational, versatility
4	Tags	Dataset Quality, manual
5	Description	serialization formats such as turtle (which is a human-friendly serialization of rdf) and JSON-LD (an adaption of JSON) can both be used to publish a dataset and each is preferred for different use-cases. Having multiple serialization formats available lowers the threshold for users to use the dataset.
6	Value type	boolean
7	Value Structure	boolean + serialization formats.
8	Measure function	-
9	Measure elements	-
10	Example	True - RDF/XML <URL> - Turtle <URL> - JSON-LD <URL>
11	Annotation procedure	
12	Identifier	SM-57

	Component	**Description**
1	Name	Dataset language
2	Source references	1, 14, 37
3	Dimension	Representational, versatility
4	Tags	dataset quality
5	Description	Checking whether data is available in different languages. For some purposes one would require the secondary dataset to be in the same language as the primary dataset or at least another language that is sufficiently understood by the target users. this metric measures the amount of languages the dataset is published in.
6	Value type	Measure
7	Value Structure	boolean, languages
8	Measure function	-
9	Measure elements	-

10	Example	True, [English, Dutch]
11	Annotation procedure	
12	Identifier	SM-58

	Component	Description
1	Name	application of content negotiation
2	Source references	14
3	Dimension	Representational, versatility
4	Tags	dataset quality, manual
5	Description	checking whether data can be retrieved in accepted formats and languages by adding a corresponding accept-header to an HTTP request
6	Value type	Boolean
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	True
11	Annotation procedure	-
12	Identifier	SM-59

Dataset Dynamicity

- *Currency*:

	Component	Description
1	Name	currency of documents/statements
2	Source references	59
3	Dimension	Dataset Dynamicity, currency
4	Tags	dataset quality, semi-automatic
5	Description	This metric measures how ‘current’ the data is. This contributes to the credibility of the data and is measured by taking into account the publication time and the current time.
6	Value type	float

7	Value Structure	value between 0 and 1 where 1 indicates perfectly current data
8	Measure function	1 - (time since last modification / (observation time - publishing time))
9	Measure elements	time since last modification = Metric <identifier> observation time = current time publication time = time of publication
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-60

	Component	**Description**
1	Name	time since last modification
2	Source references	50
3	Dimension	Dataset Dynamicity, currency
4	Tags	dataset quality, semi-automatic
5	Description	Measured time since last modification
6	Value type	dayTimeDuration
7	Value Structure	days / hours:minutes
8	Measure function	observation time - last modified time
9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-61

	Component	**Description**
1	Name	Difference between currency of datasets
2	Source references	
3	Dimension	Dataset Dynamicity, currency
4	Tags	dataset quality, semi-automatic
5	Description	Measure the difference between the primary and secondary dataset in terms of currency.
6	Value type	daytimeduration

7	Value Structure	Values closer to 0 indicate higher similarity between the currency of the datasets, negative values indicate that the secondary dataset is more current and positive values indicate a more current primary dataset.
8	Measure function	PD(metic X) - SD(metric X)
9	Measure elements	PD(metric X) measures metric X for the primary dataset SD(metric X) measures metric X for the secondary dataset
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-62

	Component	**Description**
1	Name	exclusion of outdated data
2	Source references	14, 58 @book{croft2010search, title={Search engines}, author={Croft, W Bruce and Metzler, Donald and Strohmann, Trevor}, year={2010}, publisher={Pearson Education} }
3	Dimension	Dataset Dynamicity, Currency
4	Tags	instance quality, semi-automatic
5	Description	With the usage of temporal metadata, one can (if available) detect outdated information based on the information retrieval concepts of age and freshness. By predicting the age of a resource, one can estimate when it should be updated. When a resource is overdue for an update, it can be considered outdated.
6	Value type	float
7	Value Structure	value between 0 and 1
8	Measure function	1 - (outdated data / total amount of data)
9	Measure elements	-
10	Example	-
11	Annotation procedure	SM-63

- *Volatility*:

	Component	Description
1	Name	frequency of change
2	Source references	14
3	Dimension	Dataset Dynamicity, Volatility
4	Tags	Dataset quality, semi-automatic
5	Description	refer to the change frequency attribute in a Semantic Sitemap for value of the frequency or updates of a data source
6	Value type	String
7	Value Structure	• always • hourly • daily • weekly • monthly • yearly • never
8	Measure function	-
9	Measure elements	-
10	Example	<changefreq>monthly</changefreq>
11	Annotation procedure	-
12	Identifier	SM-64

	Component	Description
1	Name	validity time interval
2	Source references	15
3	Dimension	Dataset Dynamicity, Volatility
4	Tags	
5	Description	Given that an expiry time is known, a validity interval can be computed based on the time of input and the expiry date. If no expiry time is provided, a measure used in Metric <Exclusion of outdated data> can be used to calculate when the dataset is due for an update. However, this alternative would not measure validity as strictly and

		instead would measure ‘likeliness of being up to date’ and hence valid.
6	Value type	time interval
7	Value Structure	
8	Measure function	expiry time − input time of the semantic web source OR compute an interval based on the frequency of change (metric SM-64)
9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-65

- *Timeliness*:

	Component	**Description**
1	Name	timeliness of datasource timeliness between the semantic source web and original source
2	Source references	
3	Dimension	Dataset Dynamicity, Timeliness
4	Tags	
5	Description	this metric refers to the delay of an expected change of the real-world value and its adjustment in its corresponding data value.
6	Value type	measure
7	Value Structure	boolean, delay
8	Measure function	delay = (last modified time - Expected real-world change time) boolean = true when delay is a positive value
9	Measure elements	
10	Example	True, +18 hours
11	Annotation procedure	
12	Identifier	SM-66

	Component	**Description**

1	Name	timeliness between the semantic source web and original source
2	Source references	
3	Dimension	Dataset Dynamicity, Timeliness
4	Tags	
5	Description	This metric checks if there is a difference in currency between a dataset and its distributions.
6	Value type	boolean
7	Value Structure	-
8	Measure function	-
9	Measure elements	-
10	Example	-
11	Annotation procedure	-
12	Identifier	SM-67

	Component	**Description**
1	Name	timeliness of a resource
2	Source references	
3	Dimension	Dataset Dynamicity, Timeliness
4	Tags	instance quality, semi-automatic
5	Description	a positive difference between current and expiry time of the resource implies data source to be outdated
6	Value type	integer
7	Value Structure	
8	Measure function	SPARQL: count instances where current time is past its expiry time.
9	Measure elements	-
10	Example	-
11	Annotation procedure	flag every instance which is expired
12	Identifier	SM-68

	Component	Description
1	Name	Timeliness between the ideal freshness and the data source freshness
2	Source references	49
3	Dimension	Dataset Dynamicity, Timeliness
4	Tags	
5	Description	Ideally, both datasets would be equally fresh, especially with highly volatile data. Differences in freshness could result in inconsistencies even though, at time of publication, each dataset was flawless. This metric measures the difference between the ideal freshness (freshness of the primary dataset) and the freshness of the secondary dataset.
6	Value type	float
7	Value Structure	value between 0 and 1
8	Measure function	1 - (freshness(SD) / ideal Freshness
9	Measure elements	freshness = observation time - last modification time ideal freshness = freshness(PD)
10	Example	1 (implying the datasets are equally fresh.
11	Annotation procedure	
12	Identifier	SM-69

	Component	Description
1	Name	self-reference
2	Source references	
3	Dimension	representational conciseness
4	Tags	Data quality, automatic
5	Description	Measures the amount of triples where the object is identical to the subject and therefore create loops in the data.
6	Value type	
7	Value Structure	SELECT ?s WHERE { GRAPH <%s> { ?s ?p ?s } }
8	Measure function	SPARQL: count all triples where the object is identical to the subject.

9	Measure elements	-
10	Example	-
11	Annotation procedure	flag all self-referencing triples.
12	Identifier	SM-70

template:

	Component	**Description**
1	Name	
2	Source references	
3	Dimension	
4	Tags	
5	Description	
6	Value type	
7	Value Structure	
8	Measure function	
9	Measure elements	
10	Example	
11	Annotation procedure	
12	Identifier	<number>