ANALYSIS CERTIFICATE

magister
by compilatio.net

Account :	**Bernard Merialdo**
ID :	**g3xu5sj1**
Title :	**Thesis ahmad assaf-rapporteurs.pdf**
Folder :	**Ahmad Assaf**
Comments :	*Not available*
uploaded on the :	10/07/2015 4:17 PM

Similarity document :

🟢 2%

Similarities section 3 :

🟢 1%

DETAILED INFORMATION

Title :	Thesis Ahmad Assaf-Rapporteurs.pdf
Description :	Ahmad Assaf
Analysed on :	10/07/2015 4:38 PM
Login ID :	69k8iwml
uploaded on the :	10/07/2015 4:17 PM
Upload type :	manual submission
File name :	Thesis Ahmad Assaf-Rapporteurs.pdf
File type :	pdf
Word count :	11505
Character count :	75280

TOP PROBABLE SOURCES*AMONG 1 PROBABLE SOURCE*

1.	**www.semantic-web-journal.net**/.../files/swj414.pdf	🚩 <1%

SIMILARITIES FOUND IN THIS DOCUMENT/SECTION

Matching similarities : **<1 %** ⓘ

Assumed similarities : **<1 %** ⓘ

Accidental similarities : **<1 %** ⓘ

Highly probable sources - 1	Accidental sources- 22 Sources
Less probable sources - 0	Ignored sources - 4

HIGHLY PROBABLE SOURCES

	1 Source	*Similarity*
1.	**www.semantic-web-journal.net**/.../files/swj414.pdf	<1%

LESS PROBABLE SOURCES

0 Source	*Similarity*

ACCIDENTAL SOURCES

	22 Sources	*Similarity*
1.	Document: iheyou51 - belongs to another user	<1%
2.	Source Compilatio.net gsz36	<1%
3.	Source Compilatio.net em247	<1%
4.	Source Compilatio.net bmsw14	<1%
5.	Source Compilatio.net md6k9	<1%
6.	Source Compilatio.net aeux7	<1%
7.	Source Compilatio.net 72rtpibq	<1%
8.	Source Compilatio.net cnr12	<1%
9.	Source Compilatio.net cpqr7	<1%
10.	Source Compilatio.net	<1%
11.	Source Compilatio.net dhpr6	<1%
12.	**validator.lod-cloud.net**/.../	<1%
13.	Source Compilatio.net c9mfb	<1%
14.	Source Compilatio.net ejln5	<1%
15.	**graphite.ecs.soton.ac.uk**/.../checker	<1%
16.	Source Compilatio.net aeis49	<1%
17.	Source Compilatio.net	<1%
18.	Source Compilatio.net nzpxmbtj	<1%
19.	Source Compilatio.net lvx359	<1%
20.	Source Compilatio.net x752kfdo	<1%
21.	Source Compilatio.net jpr28	<1%
22.	Source Compilatio.net	<1%

IGNORED SOURCES

4 Sources	*Similarity*

1.	**ceur-ws.org**/.../Vol-1376/LDQ2015_paper_07.pdf	✖ 6%
2.	**ceur-ws.org**/.../Vol-1362/PROFILES2015_paper1.pdf	✖ 5%
3.	**www.www2015.it**/.../companion/p159.pdf	✖ 1%
4.	**github.com**/.../ahmadassaf/opendata-checker	✖ <1%

SIMILARITIES FOUND IN THIS DOCUMENT/SECTION

Matching similarities : **<1 %**

Assumed similarities : **<1 %**

Accidental similarities : **<1 %**

TEXT EXTRACTED FROM THE DOCUMENT

25% of the datasets access information (being the dataset URL and any URL dened in its groups) have issues: generally missing or unreachable URLs. 3 datasets (1.15%) do not have a URL dened (tip, uniprotdatabases, uniprotcitations) while 45 datasets (17.3%) dened URLs are not accessible at the time of writing this paper. One dataset does not have resources information (bio2rdfchebi) while the other datasets have a total of 1068 dened resources. On the datasets resources level, we notice wrong or inconsistent values in the size and mimetype elds. However, 44 datasets have valid size eld values and 54 have valid mimetype eld values but they were not reachable, thus providing incorrect information. 15 elds (68%) of all the other access metadata are missing or have undened values. Looking closely, we notice that most of these problems can be easily xed automatically by tools that can be plugged to the data portal. For ex-

4.6. Analyzing Proling Results

59

ample, the top six missing elds are the cache last updated, cache url, urltype, webstore last updated, mimetype inner and hash which can be computed and lled automatically. However, the most important missing information which require manual entry are the dataset's name and description which are missing from 817 (76.49%) and 98 (9.17%) resources respectively. A total of 334 resources (31.27%) URLs were not reachable, thus aecting highly the availability of these datasets. CKAN resources can be of various predened types (f ile, f ile.upload, api, visualization, code, documentation). Roomba also breaks down these unreachable resources according to their types: 211 (63.17%) resources do not have valid resource type, 112 (33.53%) are les, 8 (2.39%) are metadata and one (0.029%) is example and documentation types. To have more details about the resources URL types, we created a key : objectmeta f ieldvalues group level report on the LOD cloud with resources> format:title. This aggregates the resources format information for each dataset. We observe that only 161 (62.16%) of the datasets valid URLs have SPARQL endpoints dened using the api/sparql resource format. 92.27% provided RDF example links and 56.3% provided direct links to RDF down-loadable dumps. The noisiest part of the access metadata is about license information. A total of 43 datasets (16.6%) does not have a dened license title and license id elds, where 141 (54.44%) have missing license url eld.

Figure 4.3: LOD Cloud error % by section

4.6.3

Ownership Information

Ownership information is divided into direct ownership (author and maintainer) and organization information. Four elds (66.66%) of the direct ownership information are missing or undened. The breakdown for the missing information is: 55.21% maintainer email, 51.35% maintainer, 15.06% author email, 2.32% author. Moreover, our framework performs checks to validate existing email values. 11

60

Chapter 4. Dataset Proles Generation and Validation

(0.05%) and 6 (0.05%) of the dened author email and maintainer email elds are not valid email addresses respectively. For the organization information, two eld values (16.6%) were missing or undened. 1.16% of the organization description and 10.81% of the organization image url information with two out of these URLs are unreachable.

4.6.4

Provenance Information

80% of the resources provenance information are missing or undened. However, most of the provenance information (e.g., metadata created, metadata modified) can be computed automatically by tools plugged into the data portal. The only eld requiring manual entry is the version eld which was found to be missing in 60.23% of the datasets.

4.6.5

Enriched Proles

Roomba can automatically x, when possible, the license information (title, url and id) as well as the resources MIME-type and size. 20 resources (1.87%) have incorrect mimetype dened, while 52 resources (4.82%) have incorrect size values. These values have been automatically xed based on the values dened in the HTTP response header. We have noticed that most of the issues surrounding license information are related to ambiguous entries. To resolve that, we manually created a mapping le21 standardizing the set of possible license names and urls using the open source and knowledge license information22 . As a result, we managed to normalize 123 (47.49%) of the datasets' license information. To check the impact of the corrected elds, we seeded Roomba with the enriched proles. Since Roomba uses le-based cache system, we simply replaced all the datasets json les in the \cache\datahub.io\datasets folder with those generated in \cache\datahub.io\enriched. After running Roomba again on the enriched proles, we observe that the errors percentage for missing size elds decreased by 32.02% and for mimetype elds by 50.93%. We also notice that the error percentage for missing license urls decreased by 2.32%.

4.7

Summary

In this chapter, we proposed a scalable automatic approach for extracting, validating, correcting and enriching dataset proles. This approach applies several techniques

https://github.com/ahmadassaf/opendata-checker/blob/master/util/ licenseMappings.json 22 https://github.com/okfn/licenses

21

Figure 4.4: LOD Cloud error % by information type in order to check the validity of the metadata provided and to generate descriptive and statistical information for a particular dataset or for an entire data portal. It has been noticed that the issues surrounding metadata quality aect directly dataset search as data portals rely on such information to power their search index. We noted the need for tools that are able to identify various issues in this metadata and correct them automatically. We evaluated our framework manually against two prominent data portals and proved that we can automatically scale the validation of datasets metadata proles completely and correctly. We presented the results of running Roomba over the LOD cloud group hosted in the Datahub. We discovered that the general state of the examined datasets needs attention as most of them lack informative access information and their resources suer low availability. These two metrics are of high importance for enterprises looking to integrate and use external linked data. We found out that the most erroneous information for the dataset core information are ownership related since this information is missing or undened for 41% of the datasets. Datasets resources have the poorest metadata: 64% of the general metadata, all the access information and 80% of the provenance information contained missing or undened values. We also showed that the automatic correction process can eectively enhance the quality of some information. We believe there is a need to have a community eort to manually correct missing important information like ownership information (maintainer, author, and maintainer and author emails).

Chapter 5

Objective Linked Data Quality Assessment

5.1

Introduction

We are entering an era where open is the new default. Governments, universities, organizations and even individuals are publicly publishing huge amounts of open data. This openness should be accompanied with a certain level of trust or guarantees about the quality of data. The Linked Open Data is a gold mine for those trying to leverage external data sources in order to produce more informed business decisions [28]. However, the heterogeneous nature of sources reects directly on the data quality as these sources often contain inconsistent as well as misinterpreted and incomplete information. Traditional data quality is a thoroughly researched eld with several benchmarks and frameworks to grasp its dimensions [81, 19, 150]. Data quality principles typically rely on many subjective indicators that are complex to measure automatically. The quality of data in indeed realized when it is used [103], thus directly relating to the ability of satisfying users' continuous needs. Web documents that are by nature unstructured and interlinked require dierent quality metrics and assessment techniques than traditional datasets. For example, the importance and quality of Web documents can be subjectively calculated via algorithms like Page Rank [117]. Despite the fact that Linked Open Data quality is a trending and highly demanded topic, very few eorts are currently trying to standardize, track and formalize frameworks to issue scores or certicates that will help data consumers in their integration tasks. Data quality assessment is the process of evaluating if a piece of data meets the consumers need in a specic use case [22]. The dimensionality of data quality makes it dependent on the task and users requirements. For example, DBpedia [23] and YAGO [136] are knowledge bases containing data extracted from structured and semi-structured sources. They are used in a variety of applications e.g., annotation systems [110], exploratory search [108] and recommendation engines [116]. However, their data is not integrated into critical systems e.g., life critical (e.g., medical applications) or safety critical (e.g., aviation applications) as its data quality is found to

be insufficient. In this chapter, we first propose a comprehensive objective framework to evaluate the quality of Linked Data sources. The framework is based on a refinement of the data quality principles described in [6] and surveyed in [151]. Some attributes have been grouped for more detailed quality assessments while we have also extended them by adding for each attribute a set of objective indicators. These indicators are measures that provide users with quality metrics measurable by tools regardless of the use case. For example, when measuring the quality of DBpedia dataset, an objective metric would be the availability of human or machine readable license information rather than the trustworthiness of the publishers. Secondly, we survey the landscape of Linked Data quality tools to discover that they only cover a subset of the proposed objective quality indicators. As a result, we extend Roomba (see Chapter 4) with an extensible quality measurement tool. This tool helps on one hand data owners to rate the quality of their dataset and get some hints on possible improvements, and on the other hand data consumers to choose their data sources from a ranked set.

5.2

Data Quality Assessment

In [151], the authors present a comprehensive systematic review of

data quality assessment methodologies applied to LOD.

They have extracted 26 quality dimensions and a total of 110 objective and subjective quality indicators. However, some of those objective indicators are dependent on the use case thus there is no clear separation on what can be automatically measured. For example, data completeness is generally a subjective dimension. However, the authors specified that the detection of the degree on which all the real-world objects are represented, detection of number of missing values for specific property and detection of

Main source **www.semantic-web-journal.net**/.../files/swj414.pdf <1%

the degree to which instances in the dataset are interlinked are considered as objective indicators given the presence of a gold standard or the original data source to compare with.

Moreover, lots of the defined performance dimensions like low latency, high throughput or scalability of a data source were defined as objective but are still dependent on multiple subjective factors like network congestion. In addition, there were some missing objective indicators vital to the quality of LOD e.g., indication of the openness of the dataset. The ODI certificate (see Section 4.3) comes as an online and free questionnaire for data publishers focusing on certain characteristics about their data. The questions are classified into the following categories: general information (about dataset, publisher and type of release), legal information (e.g., rights to publish), licensing, privacy (e.g., whether individuals can be identified), practical information (e.g., how to reach the data), quality, reliability, technical information (e.g., format and type of data) and social information (e.g., contacts, communities, etc.). Based on the information

provided by the data publisher, a certificate is created with one of four different ratings. Although ODI is a great initiative, the issued certificates are self-certified. ODI does not verify or review submissions but retains the right to revoke a certificate at any time. At the time of writing this paper, there was only 10,555 ODI certificates issued. The dynamicity of Linked Data makes it also very difficult to update the certificates manually, especially when these changes are frequent and affect multiple categories. There is clearly a need for automatic certification which can be supplemented with some manual input for categories that cannot be processed by machines. The emerging critical need for large, distributed, heterogeneous, and complex structured datasets identified the necessity to establish industry cooperation between vendors of RDF and Graph database technologies in developing, endorsing, and publishing reliable and insightful benchmark results. The Linked Data Benchmark Council (LDBC)1 aims to bridge the gap between the industry and the new trending stack of semantic technologies and their vendors. LDBC aims at promoting graph and RDF data management systems to be an accepted industrial solution. LDBC is not focused around measuring or assessing quality. However, it focuses on creating benchmarks to measure progress in scalability, storage, indexing and query optimization techniques to become the de facto standard for publishing performance results. In [4], the authors propose a methodology for assessing Linked Data quality. It consists of three main steps: (1) requirement analysis, (2) quality assessment and (3) quality improvement. Considering the multidimensionality of data quality, the methodology requires users to provide the details of a use case or a scenario that describes the intended usage of the data. Moreover, quality issues identification is done with the help of a checklist. The user must have prior knowledge about the details of the data in order to fill this list. Tools implementing the proposed methodology should be able to generate comprehensive quality measures. However, they will require heavy manual intervention and deep knowledge on the data examined. These issues highly affect detecting quality issue on large scale. Despite all the recent efforts in providing frameworks and tools for data quality in Linked Open Data, there is still no automatic framework for the objective assessment of Linked Data quality.

5.3

Objective Linked Data Quality Classification

The basic idea behind Linked Data is that its usefulness increases when it is more interlinked with other datasets. Tim Berners-Lee defined four main principles for

[1]

http://ldbc.eu/

publishing data that can ensure a certain level of uniformity reflecting directly data's usability [16]: • Make the data available on the Web: assign URIs to identify things. • Make the data machine readable: use HTTP URIs so that looking up these names is easy. • Use publishing standards: when the lookup is done provide useful information using standards like RDF. • Link your data: include links to other resources to enable users to discover more things. Building on these principles, we group the quality attributes into four main categories: • Quality of the entities : quality indicators that focus on the data at the instance level. • Quality of the dataset: quality indicators at the dataset level. • Quality of the semantic model: quality indicators that focus on the semantic models, vocabularies and ontologies. • Quality of the linking process: quality indicators that focus on the inbound and outbound links between datasets. In [6], the authors identified 24 different Linked Data quality attributes. These attributes are a mix of objective and subjective measures that may not be derived automatically. In this paper, we refine these attributes into a condensed framework of 10 objective measures. Since these measures are rather abstract, we should rely on quality indicators that reflect data quality [51] and use them to automate calculating datasets quality. The quality indicators are weighted. These weights give the flexibility to define multiple degrees of importance. For example, a dataset containing people can have more than one person with the same name thus it is not always true that two entities in a dataset should not have the same preferred label. As a result, the weight for that quality indicator will be set to zero and will not affect the overall quality score for the consistency measure. Independent indicators for entity quality are mainly subjective e.g.,

the degree to which all the real-world objects are represented,

the scope and level of details, etc. However, since entities are governed by the underlying model, we have grouped their indicators with those of the modeling quality. Table 5.1 lists the refined measures alongside their objective quality indicators. Those indicators have been gathered by:

• Transforming the objective quality indicators presented as a set of questions in [6] into more concrete quality indicator metrics. • Surveying the landscape of data quality tools and frameworks. • Examining the properties of the most prominent linked data models from the survey done in [10]. Table 5.1: Objective Linked Data quality framework

Quality Attribute Quality Category ID 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 Quality Indicator Existence of supporting structured metadata [71] Supports multiple serializations [151] Has different data access points Uses datasets description vocabularies Existence of descriptions about its size Existence of descriptions about its structure (MIME Type, Format) Existence of descriptions about its organization and categorization Existence of information about the kind and number of used vocabularies [151] Existence of dereferencable links for the dataset [71, 105, 61] Absence of disconnected graph clusters [105] Absence of omitted top concept [71] Has complete language coverage [105] Absence of unidirectional related concepts [71] Absence of missing labels [105] Absence of missing equivalent properties [82] Absence of missing inverse relationships [82] Absence of missing domain or range values in properties [82] Existence of an RDF dump that can be downloaded by users [51][71] Existence of a queryable endpoint that responds to direct queries Existence of valid dereferencable URLs (respond to HTTP request) Existence of human and machine readable license information [72] Existence of de-referenceable links to the full license information [72] Specifies permissions, copyrights and attributions [151] Existence of timestamps that can keep track of its modifications [52] Includes the correct MIME-type for the content [71] Includes the correct size for the content Absence of syntactic errors on the instance level [71] Absence of syntactic errors [138] Use the HTTP URI scheme (avoid using URNs or DOIs) [105] Contains marked top concepts [105] Absence of broader concepts for top concepts [105] Absence of missing or empty labels [2, 105] Absence of unprintable characters [2, 105] or extra white spaces in labels [137] Absence of incorrect data type for typed literals [71, 2] Absence of omitted or invalid languages tags [137, 105] Absence of terms without any associative or hierarchical relationships Continued on next page

Dataset Level

Completeness Links Level

Model Level

Availability

Dataset Level

Licensing Freshness

Dataset Level Dataset Level Dataset Level

Links Level Correctness

Model Level

Quality Attribute

Comprehensibility

Provenance

Coherence

Consistency

Security

Table 5.1 Objective Linked Data quality framework Quality Category ID Quality Indicator 37 Existence of at least one exemplary RDF file [151] 38 Existence of at least one exemplary SPARQL query [151] Dataset Level 39 Existence of general information (title, URL, description) for the dataset 40 Existence of a mailing list, message board or point of contact [51] 41 Absence of misuse of ontology annotations [105, 82] Model Level 42 Existence of annotations for concepts [82] 43 Existence of documentation for concepts [105, 82] 44 Existence of metadata that describes its authoritative information [52] Dataset Level 45 Usage of a provenance vocabulary 46 Usage of a versioning 47 Absence of misplaced or deprecated classes or properties [71] 48 Absence of relation and mappings clashes [137] 49 Absence of blank nodes [72] 50 Absence of invalid inverse-functional values [71] 51 Absence of cyclic hierarchical relations [133, 137, 105] Model Level 52 Absence of undefined classes and properties usage [71] 53 Absence of solely transitive related concepts [105] 54 Absence of redefinitions of existing vocabularies [71] 55 Absence of valueless associative relations [105] 56 Consistent usage of preferred labels per language tag [75, 105] 57 Consistent usage of naming criteria for concepts [82] 58 Absence of overlapping labels Model Level 59 Absence of disjoint labels [105] 60 Absence of atypical use of collections, containers and reification [71] 61 Absence of wrong equivalent, symmetric or transitive relationships [82] 62 Absence of membership violations for disjoint classes [71] 63 Uses login credentials to restrict access [151] Dataset Level 64 Uses SSL or SSH to provide access to their dataset [151]

5.3.1

Completeness

Data completeness can be judged in the presence of a task where the ideal set of attributes and objects are known. It is generally a subjective measure depending highly on the scenario and use-case in hand. For example, an entity is considered to be complete if it contains all the attributes needed for a given task, has complete language coverage [105] and has documentation properties [113, 105]. Dataset completeness has some objective measures which we include in our framework. A dataset is considered to be complete if it: • Contains supporting structured metadata [71]. • Provides data in multiple serializations (N3, Turtle, etc.) [151].

• Contains different data access points. These can either be a queryable endpoint (i.e. SPARQL endpoint, REST API, etc.) or a data dump file. • Uses datasets description vocabularies like DCAT2 or VOID3 . • Provides descriptions about its size e.g., void:statItem, void:numberOfTriples or void:numberOfDocuments. • Existence of descriptions about its format. • Contains information about its organization and categorization e.g., dcterms:subject. • Contains information about the kind and number of used vocabularies [151]. Links are considered to be complete if the dataset and all its resources have defined links [71, 105, 61]. Models are considered to be complete if they do not contain disconnected graph clusters [105]. Disconnected graphs are the result of incomplete data acquisition or accidental deletion of terms that leads to deprecated terms. In addition to that, models are considered to be complete if they have complete language coverage (each concept labeled in each of the languages that are also used on the other concepts) [105], do not contain omitted top concepts or unidirectional related concepts [71] and if they are not missing labels [105], equivalent properties, inverse relationships, domain or range values in properties [82].

5.3.2

Availability

A dataset is considered to be available if the publisher provides data dumps e.g., RDF dump, that can be downloaded by users [51, 71], its queryable endpoints e.g., SPARQL endpoint, are reachable and respond to direct queries and if all of its inbound and outbound links are dereferenceable.

5.3.3

Correctness

A dataset is considered to be correct if it includes the correct MIME-type and size for the content [71] and doesn’t contain syntactic errors [71]. Links are considered to be correct if they lack syntactic errors and use the HTTP URI scheme (avoid using URNs or DOIs) [105]. Models are considered to be correct if the top concepts are marked and do not have broader concepts (for example having incoming hasTopConcept or outgoing topConceptOf relationships) [105]. Moreover, if they don’t contain incorrect data type for typed literals [71][2], no omitted or invalid languages tags [137, 105], do not contain “orphan terms” (orphan terms are terms without any associative or hierarchical relationships) and that labels are not empty, do not contain unprintable characters or extra white spaces [137, 2, 105].

5.3.4 Consistency

Consistency implies lack of contradictions and conflicts. The objective indicators are mainly associated with the modeling quality. A model is considered to be consistent if it does not contain overlapping labels (two concepts having the same preferred lexical label in a given language when they belong to the same schema) [75, 105], consistent preferred labels per language tag [105, 137], atypical use of collections, containers and reification [71], wrong equivalent, symmetric or transitive relationships [82], consistent naming criteria in the model [105, 82], overlapping labels in a given language for concepts in the same scheme [105] and membership violations for disjoint classes [71, 82].

5.3.5 Freshness

Freshness is a measure for the recency of data. The basic assumption is that old information is more likely to be outdated and unreliable [52]. Dataset freshness can be identified if the dataset contains timestamps that can keep track of its modifications. Data freshness could be considered as a subjective measure. However, our concern is the existence of temporal information allowing dataset consumers to subjectively decide its freshness for their scenario.

5.3.6 Provenance

Provenance can be achieved at the dataset level by including metadata that describes its authoritative information (author, maintainer, creation date, etc.), versioning information and verifying if the dataset uses a provenance vocabulary like PROV [95].

5.3.7 Licensing

Licensing is a quality attribute that is measured on the dataset level. It includes the availability of machine readable license information [72], human readable license information in the documentation of the dataset or its source [72] and the indication of permissions, copyrights and attributions specified by the author [151].

5.3.8 Comprehensibility

Dataset comprehensibility is identified if the publisher provides general information about the dataset (e.g., title, description, URI). In addition, if he indicates at least one exemplary RDF file and SPARQL query and provides an active communication channel (mailing list, message board or e-mail) [51]. A model is considered to be comprehensible if there is no misuse of ontology annotations and that all the concepts are documented and annotated [105, 82].

5.3.9 Coherence

Coherence is the ability to interpret data as expected by the publisher or vocabulary maintainer [71]. The objective coherence measures are mainly associated with the modeling quality. A model is considered to be coherent when it does not contain undefined classes and properties [71], blank nodes [72], deprecated classes or properties [71], relations and

[2] http://www.w3.org/TR/vocab-dcat/
[3] http://www.w3.org/TR/void/

mappings clashes [137], invalid inverse-functional values [71], cyclic hierarchical relations [133, 137, 105], solely transitive related concepts [105], redefinitions of existing vocabularies [71] and valueless associative relations [105].

5.3.10 Security

Security is a quality attribute that is measured on the dataset level. It is identified if the publishers use login credentials, SSL or SSH to provide access to their dataset, or if they only grant access to specific users [151].

5.4 Linked Data Quality Tools

In this section, we present the results of our survey on the Linked Data quality tools. There exists a number of data quality frameworks and tools that are either standalone or implemented as modules in data integration tools. These approaches can be classified into automatic, semi-automatic, manual or crowdsourced approaches.

5.4.1 Information Quality

RDF is the standard to model information in the Semantic Web. Linked Data publishers can pick from a plethora of tools that can automatically check their RDF files for quality problems4 . Syntactic RDF checkers are able to detect errors in RDF documents like the W3C RDF Validator5 , RDF:about validator and Converter6 and The Validating RDF Parser (VRP)7 . The RDF Triple-Checker8 is an online tool that helps find typos and common errors in RDF data.

Vapour9 [18] is a validation service to check whether semantic Web data is correctly published according to the current best practices [16].

4 5

http://www.w3.org/2001/sw/wiki/SWValidators http://www.w3.org/RDF/Validator/ 6 http://rdfabout.com/demo/validator/ 7 http://139.91.183.30:9090/RDF/VRP/index.html 8 http://graphite.ecs.soton.ac.uk/checker/ 9 http://validator.linkeddata.org/vapour

ProLOD [26], ProLOD++ [1], Aether [106] and LODStats [13] are not purely quality assessment tools. They are Linked Data profiling tools providing clustering and labeling capabilities, schema discovery and statistics about data types and patterns. The statistics are about properties distribution, link-to-literal ratio, number of entities and RDF triples, average properties per entity and average error.

5.4.2 Modeling Quality

Reusing existing ontologies is a common practice that Linked Data publishers are always trying to adopt. However, ontologies and vocabularies development is often a long error-prone process especially when many contributors are working consecutively or collaboratively [138]. This can introduce deficiencies such as redundant concepts or conflicting relationships [63]. Getting to choose the right ontology or vocabulary is vital to ensure modeling correctness and consistency. 5.4.2.1 Semi-automatic Approaches

DL-Learner [96] uses supervised machine learning techniques to learn concepts from user-provided examples. CROCUS [34] applies a cluster-based approach for instancelevel error detection. It validates identified errors by non-expert users and iterate to reach higher quality ontologies that can be safely used in industrial environments. 5.4.2.2 Automatic Approaches

qSKOS10 [105] scans SKOS vocabularies to provide reports on vocabulary resources and relations that are problematic. PoolParty checker11 is an online service based on qSKOS. Skosify [137] supports OWL and RDFS ontologies by converting them into well-structured SKOS vocabularies. It includes automatic correction abilities for quality issues that have been observed by reviewing vocabularies on the Web. The OOPS! pitfall scanner [123] evaluates OWL ontologies against a rules catalog and provides the user with a set of guidelines to solve them. ASKOSI12 retrieves vocabularies from different sources, stores and displays the usage frequency of the different concepts used by different applications. It promotes reusing existing information systems by providing better management and presentation tools. Some errors in RDF will only appear after reasoning (incorrect inferences). In [132, 140] the authors perform quality checking on OWL ontologies using integrity constraints involving the Unique Name Assumption (UNA) and the Closed World Assumption (CWA). Pellet13 provides reasoning services for OWL ontologies. It in10 11

https://github.com/cmader/qSKOS http://www.poolparty.biz/ 12 http://www.w3.org/2001/sw/wiki/ASKOSI 13 http://clarkparsia.com/pellet corporates a number of heuristics to detect and repair quality issues among disjoint properties, negative property assertions and reexive, irreexive, symmetric, and anti-symmetric properties. Eyeball14 provides quality inspection for RDF models (including OWL). It provides checks for a variety of problems including the usage of unknown predicates, classes, poorly formed namespaces, literal syntax validation, type consistency and other heuristics. RDF:Alerts15 provides validation for many issues highlighted in [71] like misplaced, undened or deprecated classes or properties.

5.4.3

Dataset Quality

Considering the large amount of available datasets in the Linked Open Data, users have a hard time trying to identify appropriate datasets that suit certain tasks. The most adopted approaches are based on link assessment. Provenance-based approaches and entity-based approaches are also used to compute not only dataset rankings, but also rankings on the entity level. 5.4.3.1 Manual Ranking Approaches

Sieve [109] is a framework for expressing quality assessment and fusion methods. It is implemented as a component of the Linked Data Integration Framework (LDIF)16 . Sieve leverages the LDIF provenance metadata as quality indicators to produce quality assessment scores. However, despite its nice features, it is only targeted to perform data fusion based on user-congurable conict resolution tasks. Moreover, since Sieve main input is provenance metadata, it is only limited to domains that can provide such metadata associated with their data. SWIQA [58] is a framework providing policies or formulas controlling information quality assessment. It is composed of three layers: data acquisition, query and ontology layers. It uses query templates based on the SPARQL Inferencing Notation (SPIN)17 to express quality requirements. The queries are built to compute weighted and unweighted quality scores. At the end of the assessment, it uses vocabulary elements to annotate important values of properties and classes, assigning inferred quality scores to ontology elements and classifying the identied data quality problems. 5.4.3.2 Crowd-sourcing Approaches

There are several quality issues that can be dicult to spot and x automatically. In [2] the authors highlight the fact that the RDFication process of some data can

14 15

http://jena.sourceforge.net/Eyeball/ http://swse.deri.org/RDFAlerts/ 16 http://ldif.wbsg.de/ 17 http://spinrdf.org/

be more challenging than others, leading to errors in the Linked Data provisioning process that needs manual intervention. This can be more visible in datasets that have been semi-automatically translated to RDF from their primary source (the best example for this case is DBpedia [23]). The authors introduce a methodology to adjust crowdsourcing input from two types of audience: 1) Linked Data experts, researchers and enthusiasts through a contest to nd and classify erroneous RDF triples and 2) Crowdsourcing through the Amazon Mechanical Turk18 . TripleCheckMate [88] is a crowdsourcing tool used by the authors to run out their assessment supported by a semi-automatic quality verication metrics. The tool allows users to select resources, identify and classify possible issues according to a predened taxonomy of quality problems. It measures inter-rater agreements, meaning that the resources dened are checked multiple times. These features turn out to be extremely useful to analyze the performance of users and allow better identication of potential quality problems. TripleCheckMate is used to identify accuracy issues in the object extraction (completeness of the extraction value for object values and data types), relevancy of the extracted information, representational consistency and interlinking with other datasets. 5.4.3.3 Semi-automatic Approaches

Luzzu [44] is a generic Linked Data quality assessment framework. It can be easily extended through a declarative interface to integrate domain specic quality measures. The framework consists of three stages closely corresponding to the methodology in [4]. They believe that data quality cannot be tackled in isolation. As a result, they require domain experts to identify quality assessment metrics in a schema layer. Luzzu is ontology driven. The core vocabulary for the schema layer is the Dataset Quality Ontology (daQ) [43]. Any additional quality metrics added to the framework should extend it. RDFUnit19 is a tool centered around the denition of data quality integrity constraints [87]. The input is a dened set of test cases (which can be generated manually or automatically) presented in SPARQL query templates. One of the main advantages for this approach is the ability to discover quality problems beyond conventional quality heuristics by encoding domain specic semantics in the test cases. LiQuate [128] is based on probabilistic models to analyze the quality of data and links. It consists of two main components: A Bayesian Network builder and an ambiguity detector. They rely on data experts to represent probabilistic rules. LiQuate identies redundancies (redundant label names for a given resource), incompleteness (incomplete links among a given set of resources) and inconsistencies (inconsistent

18 19

https://www.mturk.com/ http://github.com/AKSW/RDFUnit

links). Quality Assessment of Data Sources (Flemming's Data Quality Assessment Tool)20 calculates data quality scores based on manual user input. The user should assign weights to the predened quality metrics and answer a series of questions regarding the dataset. These include, for example, the use of obsolete classes and properties by dening the number of described entities that are assigned disjoint classes, the usage of stable URIs and whether the publisher provides a mailing list for the dataset. The main disadvantage for using this tool is the manual intervention which requires deep knowledge in the dataset examined. Moreover, the tool lacks support for several quality concerns like completeness or consistency. LODGRefine [146] is the Open Refine21 of Linked Data. It does not act as a quality assessment tool, but it is powerful in cleaning and refining raw instance data. LODGRefine can help detect duplicates, empty values, spot inconsistencies, extract Named Entities, discover patterns and more. LODGRefine helps in improving the quality of the dataset by improving the quality of the data at the instance level. 5.4.3.4 Automatic Ranking Approaches

The Project Open Data Dashboard22 tracks and measures how US government websites implement the Open Data principles to understand the progress and current status of their public data listings. A validator analyzes machine readable files e.g., JSON files for automated metrics like the resolved URLs, HTTP status and content-type. However, deep schema information about the metadata is missing like description, license information or tags. Similarly on the LOD cloud, the Data Hub LOD Validator23 gives an overview of Linked Data sources cataloged on the Data Hub. It offers a step-by-step validator guidance to check a dataset

completeness level for inclusion in the LOD cloud.

The results are divided into four different compliance levels from basic to reviewed and included in the LOD cloud. Although it is an excellent tool to monitor LOD compliance, it still lacks the ability to give detailed insights about the completeness of the metadata and overview on the state of the whole LOD cloud group and is very specific to the LOD cloud group rules and regulations. Link-based Approaches The basic idea behind link assessment tools is to provide rankings for datasets based on the cardinality and types of the relationships with other datasets. Tradi20 21

http://linkeddata.informatik.hu-berlin.de/LDSrcAss/datenquelle.php http://openrefine.org/ 22 http://labs.data.gov/dashboard/ 23 http://validator.lod-cloud.net/

tional link analysis has proven to be an effective way to measure the quality of Web documents search. Algorithms like PageRank [117] and HITS [85] became successful based on the assumption that a certain Web document is considered to have higher importance or rank if it has more incoming links that other Web documents [30][33]. However, the basic assumption that links are equivalent does not suit the heterogeneous nature of links in the Linked Open Data. Thus, the previous approaches fall short to provide reliable rankings as the types of the links can have a direct impact on the ranking computation [141]. The first adaption of PageRank for Semantic Web resources was the Ontology Rank algorithm implemented in the Swoogle search engine [46]. They use a rational random surfing model that takes into account the different types of links between discovered sets and compute rankings based on three levels of granularity: documents, terms and RDF graphs. ReConRank [70] rankings are computed at query time based on two levels of granularity: resources and context graphs. DING [141] adapted the PageRank to rank datasets based on their interconnections. DING can also automatically assign weights to different link types based on the nature of the predicate involved in the link. Broken links are a major threat to Linked Data. They occur when resources are removed, moved or updated. DSNotify24 [67] is a framework that informs data consumers about the various types of events that occur on data sources. Their approach is based on an indexing infrastructure that extracts feature vectors and stores them to an index. A monitoring module detects events on sources and write them to a central event log which pushes notifications to registered applications. LinkQA [61] is a fully automated approach which takes a set of RDF triples as an input and analyzes it to extract topological measures (links quality). However, the authors depend only on five metrics to determine the quality of data (i.e.degree, clustering coefficient, centrality, sameAs chains and descriptive richness through sameAs). Provenance-based Approaches Provenance-based assessment methods are an important step towards transparency of data quality in the Semantic Web. In [66]25 the authors use a provenance model as an assessment method to evaluate the timeliness of Web data. Their model identifies types of “provenance elements” and the relationships between them. Provenance elements are classified into three types: actors, executions and artifacts. The assessment procedure is divided into three steps: 1) Creating provenance graph based on the defined model 2) Annotating the graph with impact values 3) Calculating the information quality score. In [52] the authors describe a set of provenance-based assessment metrics to support quality assessment and repair in Linked Open Data. They rely on both data and metadata and use indicators like the source reputation,

24 25

http://www.cibiv.at/˜niko/dsnotify/ http://trdf.sourceforge.net

freshness and plausibility. In [65] the authors introduce the notion of naming authority which connects an identifier with the source to establish a connection to its provenance. They construct a naming authority graph that acts as input to derive PageRank scores for the data sources. Entity-based Approaches Sindice [143] uses a set of techniques to rank Web data. They use a combination of query dependent and query independent rankings implemented in the Semantic Information Retrieval Engine (SIREn)26 to produce a final entity rank. Their query dependent approach rates individual entities by aggregating the the score of the matching terms with a term frequency - inverse subject frequency (tf-isf) algorithm. Their query independent ranking is done using hierarchical links analysis algorithms [45]. The combination of these two approaches is used to generate a global weighted rank based on the dataset, entities and links ranks.

5.4.4

Queryable End-point Quality

The availability of Linked Data is highly dependent on the performance qualities of its queryable end-points. The standard query language for Semantic Web resources is SPARQL. As a result, we focus on tools measuring the quality of SPARQL endpoints. In [32]27 the authors present their findings to measure the discoverability of SPARQL endpoints by analyzing how they are located and the metadata used to describe them. In addition to that, they also analyze endpoints interoperability by identifying features of SPARQL 1.0 and SPARQL 1.1 that are supported. The authors tackled the endpoints efficiency by testing the time taken to answer generic, content-agnostic SPARQL queries over HTTP.

5.5

An Objective Quality Assessment Framework

Looking at the list of objective quality indicators, we found out that a large amount of those indicators can be examined automatically from attached datasets metadata found in data portals. As a result, we have chosen to extend Roomba as it performs the preprocessing steps needed to objectively measure datasets quality. In our framework, we have presented 30 objective quality indicators related to dataset and links quality. The remainder 34 indicators are related to the entities and models quality and cannot be checked through the attached metadata. Excluding security related quality indicators as LOD cloud group members should not restrict access to their datasets, the Roomba quality extension is able to assess and score 23

26 27

http://siren.sindice.com/ http://labs.mondeca.com/sparqlEndpointsStatus/

of them (82%). We have extended Roomba with 7 submodules that will check various dataset quality indicators shown in Table 5.2. Some indicators have to be examined against a finite set. For example, to measure the quality indicator no.3 (having different data access points), we need to have a defined set of access points in order to calculate a quality score. Since Roomba runs on CKAN-based data portals, we built our quality extension to calculate the scores against the CKAN standard model (see Section 3.1).

Quality Indicator 1 2 3 4 5 6 7 9 18 19 20 21 22 24 25 26 28,29 37 39 40 44 46

Assessment Method Check if there is a valid metadata file by issuing a package show request to the CKAN API Check if the format field for the dataset resources is defined and valid Check the resource type field with the following possible values file, file.upload, api, visualization, code, documentation Check the resources format field for meta/void value Check the resources size or the triples extras fields Check the format and mimetype fields for resources Check if the dataset has a topic tag and if it is part of a valid group in CKAN Check if the dataset and all its resources have has a valid URI Check if there is a dereferenceable resource with a description containing string dump Check if there is a dereferencable resource with resource type of type api Check if all the links assigned to the dataset and its resources are dereferenceable Check if the dataset contains valid license id and license title Check if the license url is dereferenecable Check if the dataset and its resources contain the following metadata fields metadata created, metadata modified, revision timestamp, cache last updated Check if the content-type extracted from the a valid HTTP request is equal to the corresponding mimetype field. Check if the content-length extracted from the a valid HTTP request is equal to the corresponding size field. Check that all the links are valid HTTP scheme URIs Check if there is at least one resource with a format value corresponding to one of example/rdf+xml, example/turtle, example/ntriples, example/x-quads, example/rdfa, example/x-trig Check if the dataset and its tags and resources contain general metadata id, name, type, title, description, URL, display name, format Check if the dataset contain valid author email or maintainer email fields Check if the dataset and its resources contain provenance metadata maintainer, owner org, organization, author, maintainer email, author email Check if the dataset contain and its resources contain versioning information version, revision id

Table 5.2: Objective Quality Assessment Methods for CKAN-based Data Portals

5.5.1

Quality Score Calculation

A CKAN portal contains a set of datasets D = {D1 , ...Dn }. We denote the set of resources Ri = {r1 , ..., rk }, groups Gi = {g1 , ..., gk } and tags Ti = {t1 , ..., tk } for Di D(i = 1, ..., n) by R = {R1 , ..., Rn }, G = {G1 , ..., Gn } and T = {T1 , ..., tn } respectively. Our quality framework contains a set of measures M = {M1 , ..., Mn }. We denote the set of quality indicators Qi = {q1 , ..., qk } for Mi M(i = 1, ..., n) by Q = {Q1 , ..., Qn }. Each quality indicator has a weight, context and a score Qi < weight, context, score >. In Roomba, all the weights are equal and set to 1. However, they can be adjusted manually to rank the quality indicators. Each Qi of Mi (for i = 1,...n) is applied to one or more of the resources, tags or groups. The indicator context is defined where Qi R G T. The quality indicator score is based on a ratio between the number of violations V and the total number of instances where the rule applies T multiplied by the specified weight for that indicator. In some cases, the quality indicator score is a boolean value (0 or 1). For example, checking if there is a valid metadata file (QI.1) or checking if the license url is dereferenceable (QI.22). Q weightedscore = (V /T) Q < weight > (5.1)

Q weightedscore is an error ratio. A quality measure score should reflect the alignment of the dataset with respect to the quality indicators. The quality measure score M is calculated by dividing the weighted quality indicator scores sum by the total number of instances in its context, as the following formula shows:

n

M = 1 ((

i=1

Qi weightedscore)/ | Qi context |)

(5.2)

5.5.2

Evaluation

In our evaluation, we focused on two aspects: i)quality profiling correctness which manually assesses the validity of the errors generated in the report, and ii)quality profiling completeness which assesses if Roomba covers all the quality indicators in Table 5.2. Profiling Correctness To measure profile correctness, we need to make sure that the issues reported by Roomba are valid. On the dataset level, we chose five datasets from the LOD Cloud detailed in Table 5.3. After running Roomba and examining the results on the selected datasets and groups, we found out that our framework provides 100% correct results on the in-

Dataset ID	Resources	Tags
dbpedia	10	21
event-media	9	15
geolinkeddata	4	13
nytimes-linked-open-data	5	14
yovisto	6	20

Table 5.3: Datasets chosen for the correctness evaluation dividual dataset level. Roomba's aggregation have been evaluated in [11], thus we can infer that the quality profiler at the group and portal level also produces correct profiles. Profiling Completeness We analyzed the completeness of our framework by manually constructing a synthetic set of profiles28 . These profiles cover the indicators in Table 5.2. After running our framework at each of these profiles, we measured the completeness and correctness of the results. We found out that our framework covers indeed all the quality problems discussed.

Figure 5.1: Average Error % per quality indicator for LOD group

5.5.3

Experiments and Analysis

In this section, we provide the experiments done using the proposed framework. Listing 5.1 shows an excerpt of the generated quality report (see appendix B for full report). All the experiments are reproducible by Roomba and their results are available on its Github repository. We have run the framework on the LOD cloud containing 259 datasets at the time of writing this paper. We ran the instance and

[28] https://github.com/ahmadassaf/opendata-checker/tree/master/test

resource extractor in order to cache the metadata files for these datasets locally and ran the quality assessment process which took around two hours on a 2.6 Ghz Intel Core i7 processor with 16GB of DDR3 memory machine. We found out that licensing, availability and comprehensibility had the worst quality measures scores: 19.59%, 26.22% and 31.62% respectively. On the other hand, the LOD cloud datasets have good quality scores for freshness, correctness and provenance as most of the datasets have an average of 75% for each one of those measures. Figure 5.1 shows the average errors percentage in quality indicators grouped by the corresponding measures. The error percentage is the inverse quality. For example, 86.3% of the datasets resources do not have information about its size, which means that only 13.7% of the datasets are considered in good quality for this indicator. After examining the results, we notice that the worst quality indicators scores are for the comprehensibility measure where 99.61% of the datasets did not have valid exemplary RDF file (QI.37) and did not define valid point of contact (QI.40). Moreover, we noticed that 96.41% of the datasets queryable endpoints (SPARQL endpoints) failed to respond to direct queries (QI.19). After careful examination, we found that the cause was incorrect assignment for metadata fields. Data publishers specified the resource format field as an api instead of the specifying the resource type field.

```
==================================================================== DatasetQu
ality Report ====================================================================
completeness quality Score : 50.22% availability quality Score : 26.22% licensing quality Score :
```

```
19.59% freshness quality Score : 79.49% correctness quality Score : 72.06% comprehensibility quality Score : 31.62% provenance quality Score : 74.07% Average total quality Score : 50.47% ================================================================== Quality Indicators Average Error % ================================================================== Quality Indicator : Supports multiple serializations : 11.35% Quality Indicator : Has different data access points : 19.31% Quality Indicator : Uses datasets description vocabularies : 88.80% Quality Indicator : Existence of descriptions about its size : 86.30% Quality Indicator : Existence of descriptions about its structure : 83.67%
```

Listing 5.1: Excerpt of the LOD cloud group quality report To drill down more on the availability issues, we generated a metadata profile assessment report using Roomba's metadata profiler. We found out that 25% of the datasets access information (being the dataset URL and any URL defined in its

groups) has issues related to them (missing or unreachable URLs). Three datasets (1.15%) did not have a URL defined while 45 datasets (17.3%) defined URLs were not accessible at the time writing this paper. Out of the 1068 defined resources 31.27% were not reachable. All these issues resulted in a 26.22% average availability score. This can highly affect the usability of those datasets especially in an enterprise context.

5.6

Roomba Quality Extension vs. state of the art

Looking at Section 5.4 we notice that there is a plethora of tools (syntactic checkers or statistical profilers) that automatically check the quality of information at the entities level. Moreover, various tools can automatically check the models against the objective quality indicators mentioned. OOPS! covers all of them with additional support for the other common modeling pitfalls in [82]. PoolParty covers also a wide set of those indicators but it targets SKOS vocabularies only. However, we notice a lack in automatic tools to check the dataset quality especially in its completeness, licensing and provenance measures. Table summarizes the automatic dataset quality approaches that have implemented tools (full circle denotes full quality indicator assessment, while half circle denoted partial assessment). As can be seen in Table 5.4 Roomba covers most of the quality indicators with its focus on completeness, correctness provenance and licensing. Roomba is not able to check the existence of information about the kind and number of used vocabularies (QI.8), license permissions, copyrights and attributes (QI.23), exemplary SPARQL query (QI.38), usage of provenance vocabulary (QI.45) and is not able to check the dataset for syntactic errors (QI.27). These shortcomings are mainly due to the limitations in the CKAN dataset model. However, syntactic checkers and additional modules to examine vocabularies usage could be easily integrated in Roomba to fix QI.27, QI.8 and QI.45. Roomba's metadata quality profiler can fix QI.23 as we have manually created a mapping file standardizing the set of possible license names and their information29 . We have also used the open source and knowledge license information30 to normalize license information and add extra metadata like the domain, maintainer and open data conformance. The quality report is currently generated in a JSON format. Leveraging quality vocabularies like the Data Quality Vocabulary (DQV) [42] allows us to expose this data in a machine readable format so that they can be automatically consumed by various applications.

https://github.com/ahmadassaf/opendata-checker/blob/master/util/ licenseMappings.json 30 https://github.com/okfn/licenses

29

Tool\Indicator LOV Data.gov Roomba 1 2 3 4 G G G G G G G G

5 6 7 8 9 18 19 20 21 22 23 24 25 26 27 28 29 37 38 39 40 44 45 46 63 64 Q Q G G G Q G Q G G Q G G G Q G G Q G G G G G G G G G G G G G G G G G G G

Table 5.4: Functional Comparison of Automatic Linked Data quality Tools

5.7

Summary

In this section, we have presented a comprehensive objective quality framework applied to the Linked Open Data. We have built upon previous efforts with focus on objective data quality measures. We have identified a total of 64 quality indicators that were mapped when suitable to four main categories (entity, dataset, links, models). We have also surveyed more than 30 different tools that measure different quality aspects of Linked Open Data. We identified several gaps in the current tools and identified the need for a comprehensive evaluation and assessment framework and specifically for measuring quality on the dataset level. As a result, we presented an extension of Roomba that covers 82% of the suggested datasets objective quality indicators. Based on our experiments running Roomba on the LOD cloud, we discovered that the general state of the datasets needs attention as most of them have low completeness, provenance, licensing and comprehensibility quality scores.

Conclusion of Part I

In this part, we presented the various parts required to automatically assess and build harmonized dataset profiles. First, we surveyed the landscape of various models and vocabularies that described datasets on the web. We have identified four main sections that should be included in the model and classified the information to be included into eight types. We proposed HDL, a harmonized dataset model, that takes the best out of these models and extends them to ensure complete metadata coverage to enable data discovery, exploration and reuse. Second, we detail the gaps in the current tools for automatic validation and generation of dataset profiles. Afterwards, we propose Roomba to tackle these gaps and show the results of running it on various data portals. Last, we cover the quality dimension from HDL. We propose an objective assessment framework by identifying quality indicators that can be automatically measured by tools. We further survey the landscape of quality tools and discover various shortcomings. As a result, we extend Roomba and cover 82% of the proposed quality indicators. Going back to our scenario, our data portal administrator Paul will be able to use HDL as a basis to extend and present the datasets he controls. Moreover, he can use HDL and the proposed mappings as a basis to extend Roomba to support various dataset models like DKAN or Socrata. Roomba with its quality extension helps Paul to have a detailed overview on the health and quality of the datasets. He can use it to automatically fix some issues, and notify the datasets owners of the other issues to be manually fixed. He will be able to identify spam datasets resulting in higher data quality. Dan on the other will be able to have access to cleaner, richer set of datasets. He will be able to examine detailed attributes of the datasets. This will help Dan to make more infomred decisions on which dataset to use in his report.

Part II

Towards Enriched Enterprise Data

Overview of Part II

In Part II, we focus on building tools and frameworks to enable data integration and semantic enrichment of enterprise data. We highlight the various challenges and tackle them in an incremental manner. In Chapter 6, we overview the background of our work in Data Integration and semantic enrichment. We first introduce basic concepts in Business Intelligence and various relevant tools in the SAP ecosystem. We finally overview relevant social media outlets that can expose relevant information useful for the decision making process. In Chapter 7, we identify the need for an enterprise knowledge base. We detail the challenges and design decisions to import DBpedia into SAP HANA. We further present a set of tools that enable entity disambiguation, entity properties rankings and semantic enrichment on top of DBpedia. We also enhance an in-house schema matching tool called AMC with a set of matchers that show that using Linked Data to map cell values with instances and column headers with types improves significantly the quality of the matching results and therefore should lead to more informed business decisions. In Chapter 8, we note that aggregating relevant social news is not an easy task. We present a semantic social news aggregation framework called SNARC. SNARC is a service that uses semantic web technology and combines services available on the web to aggregate social news. SNARC brings live and archived information to the user that is directly related to his active page. The key advantage is an instantaneous access to complementary information without the need to dig for it. Information appears when it is relevant enabling the user to focus on what is really important.

Chapter 6

Background

6.1

Data Integration

Data Integration (DI) is the process of providing the user with a unified view of data residing at different sources [98]. Data Integration is a challenging task since these sources are in many real-world applications, mutually inconsistent. Various approaches and methodologies have been proposed to solve the DI problem in the enterprise: • XML as a hierarchical data format can be used as a uniform standard uniform for data representation. However, extending XML to provide complex mappings and source descriptions is difficult. • SOA can be seen as a holistic approach for distributed systems communication and architecture. In its core, SOA aims at minimizing impedance in the architecture paving the way for easier communication between data sources. However, in [54], the authors argue that SOA is well-suited for transactional processing rather than an approach for data integration. • Ontologies can be used as a rich format to describe queries and data mappings between schemas and sources. However, developing ontologies require specific skills and it is difficult to provide a complete model that captures the dynamics of the enterprise. • Linked Data paradigm is a slightly different approach from the ontology-based by exploiting Semantic Web technologies like RDF to represent enterprise taxonomies. The LD approach allows terms to be easily reused and extended. Data integrated from various resources should be loaded into a central repository often referred to as a Data Warehouse (DW). A Data Warehouse is a large repository where integrated data from different resources reside for the purpose of analysis. Feeding data into the warehouse is done using the Extract-Transform-Load (ETL) process: First the data is extracted from the operational source systems (ERP, CRM, etc.) and then the transformation process is applied in order to unify the data into the warehouse format. Finally the loading is applied to import the data to the warehouse.

6.2

Business Intelligence

Business Intelligence (BI) is the set of techniques and tools for transforming raw data into meaningful and useful information to be used in the decision making process [129]. BI consists of various number of components including Data Integration, Data Quality and Data Warehousing among others.

6.2.1

Multidimensional Model

The traditional relational model is efficient in performing ”online” transactions. However, it has clear shortcomings when the objective is to analyze large scale data. The multidimensional model is designed specifically to support data analysis by presenting data as facts with associated numerical values. The multidimensional model has the following fundamental concepts: • Dimensions: Textual data used for labeling, selection, filtering and grouping of data at various levels of details. A dimension is organized into a containmentlike hierarchy composed of number of levels, each of which represents a specific level of details. The instances of the dimensions are typically called dimension values or members; each value or member belongs to a particular level. Figure 6.1 shows an example of a hierarchical geography dimension.

Figure 6.1: Example of a hierarchical geography dimension

• Measures: A measure has two components, a numerical property and a formula (usually an aggregation function such as sum or average). Measures generally represent the properties of a chosen fact. • Facts: Facts are the objects that present the subject of the analysis. They are mostly defined by their combination of dimension values. a fact has a certain granularity which is determined by the levels from which its dimension values are drawn.

• Cubes: A cube is a multidimensional data structure for capturing and analyzing data. It generalizes the tabular spreadsheet such as there can be any number of dimensions (in contrast to only two in the tabular spreadsheets). • Pivot Tables: A pivot table is a two dimensional table of data with associated subtotals and totals. It may also allow the user to use hierarchies to drill down or roll up. It can be also nested into several dimensions on one axis or pivoted such as the dimensions can be rotated (swapping x and y). 6.2.1.1 Relational Representation

There are two principal ways of representing dimensions: • Star Schema: A star schema has one table for each dimension. This table has a key column and one column for each level of the dimension. Furthermore, a star schema has a Fact Table that hold a row for each multidimensional fact and has a column for each dimension. The primary key in the dimension tables is typically a surrogate key (ID). This results in better storage, prevention of key-reuse problems and more efficient query processing. • Snowflake Schema: Very similar to the star schema. However, it contains several dimension tables for each dimension. This results in removing the redundancy found in star schemas. As a result, querying the schema is now harder since several joins must be applied resulting in longer processing time to compute the results. 6.2.1.2 Analysis and Querying

Querying multidimensional is done by special systems that aggregate measure values over a range of dimension values. One of the widely used systems is the Online Analytical Processing (OLAP). OLAP systems provide fast answers to queries that aggregate large amounts of data to find overall trends; the results are presented in a multidimensional model. As opposed to the well known Online Analytical Transaction Processing (OLTP) the focus is on data analysis rather than transactions. OLAP systems generally never delete nor update its data; only additions of new data takes place periodically, thus OLAP systems are optimized for retrieving and summarizing large amounts of data. The support for analysis and querying on cubes is done using these operations: • Rolling up: Rolling up causes the data view to go up to a higher cross grained view

• Drilling down and Drilling Out: The opposite of rolling up, the data view becomes more fine grained and detailed. Drilling out occurs when a drill down is done by including an additional dimension. After a drill out the measure values are spread out among more cells. • Slicing and Dicing: A slice happens when an analyst wishes to consider a subset of the cube, so he selects a specific value for a dimension. It is possible to slice the result further in what is called a dice. Slicing generally refer to filtering out data, and dicing refers to grouping out the filtered data. • Drill across: This is done when we do operation on more than one cube that share one or more conformed dimensions. The data from these cubes is combines by these shared dimensions, this in relational terms corresponds to a Full Outer-join • Pivot: Allows an analyst to rotate the cube in space to see its various faces.

6.2.2

SAP BI Application Suite

The SAP BI application suite can be divided into the following main areas: • Analysis Solutions: Empower business analysts with the ability to analyze multidimensional data and quickly answer sophisticated business questions. • Discovery Solutions: Provide an interface to access, transform and visualize data in a self-serviced way. • Predictive Solutions: Provide intuitive and easy-to-use environment to design and visualize complex predictive models. • Dashboard Solutions: Allow creation of rich visualizations that allow users to interact in real time with their data. • Reporting Solutions: Offers powerful interfaces that enable not only analysts, but also non-technical users to ask spontaneous and iterative business questions about their data. The output is a static reports representing snapshots of the data.

6.3

SAP High Performance Analytic Appliance (HANA)

SAP High Performance Analytic Appliance (HANA)1 is an in-memory data platform that is deployable as an on-premise appliance, or in the cloud. It is a revolutionary

[1] http://hana.sap.com/

platform that is best suited for performing real-time analytics, and developing and deploying real-time applications. At the core of this real-time data platform is the SAP HANA database (see Figure 6.2) which leverages the cheap price of memory chips and does the computation operations all in the memory instead of disk. For BI and Real-time analytics HANA specializes on: • Data Warehousing: Provides real-time data warehousing which allows businesses to rapidly access their Enterprise Data Warehouse (EDW). • Operational Reporting: Provides real-time insights and Business Intelligence from transaction systems such as ERP. • Predictive and text analysis on Big Data: Provides the ability to perform predictive and text analysis on large volumes of data in real-time. With its text search/analysis capabilities SAP HANA also provides a robust way to leverage unstructured data.

Figure 6.2: SAP's High Performance Analytic Appliance (HANA) with the SAP BI suite HANA has both columns and rows stores for data storage, the user specifies on which data store he wishes to put his data. Row stores are fit for traditional transaction systems (traditional databases) when transactions are done on row level. However BI queries or analytical queries are done on subsets of columns as the database does not need to access all the elements in a row in order to fetch the required data. HANA has mainly three views on data:

• Attribute Views: Used to model dimensions and perform all types of joins. In most cases used to model master data like entities (like Product, Location, Business Partner). For example, our analyst Dan have accident details scattered in more than one table. However, he needs to model an accident as one entity. To do that, he needs to create an attribute view that aggregates data from different tables into one single entity which is Accident. • Analytical Views: Used for calculation and aggregation. Adds transactional tables and measures (key figures), calculates aggregates (e.g., Number of Products sold per year), joins Attribute Views. It is defined at least on one fact table. In most cases used for exposing transactional data by joining the fact table with Attribute Views. • Calculation Views: Performs complex views calculations that are not possible with other views.

6.3.1 HANA XS-Engine

Consuming data from HANA needs a lot of pre-configuration. To ease this process, the XS-Engine was created to act as lightweight application server within HANA DB. It is a presentation logic on client side that encapsulates control flow logic and calculation logic while providing REST and ODATA interfaces.

6.3.2 Active Information Store (AIS)

The Active Information Store (AIS) is a graph engine built on top of HANA. AIS provides storage and query services on graphs. AIS offers a flexible data representation model (see Figure 6.3) that contains:

Figure 6.3: AIS data model

• Info Items: They are the vertices in the graph. They represent a unique single identifiable data instance. Info Items can have a set of properties that describe them. Each Info Item is identified by its URI and must belong to at least one workspace. A Workspace establishes a scope for visibility and access control. • Associations: They are the edges in the graph. Associations can further have attributes which describe them. • Attributes: Typed properties used to describe Info Items and Associations.

6.4 Social Web