Enabling Self-Service Data Provisioning Through Semantic Enrichment of Data

Ahmad Assaf

A doctoral dissertation submitted to:

TELECOM ParisTech

in partial fulfillment of the requirements for the degree of:

Doctor of Philosophy

Specialty : Computer Science and Multimedia

Supervisor:

Dr. Raphaël Troncy - EURECOM, France

Dr. Aline Senart - SAP, France

In the Name of God, Most Gracious, Most Merciful

Abstract

Enterprises use a wide range of heterogeneous information systems in their business activities such as Enterprise Resource Planning (ERP), Customer Relationships Management (CRM) and Supply Chain Management (SCM) systems. In addition to the large amounts of heterogeneous data produced by those systems, external data is an important resource that can be leveraged to enable taking quick and rational business decisions. Classic Business Intelligence (BI) and even the newer Agile Visualization tools focus much of their selling features on attractive and unique visualizations. Preparing data for those visualizations still remains the far more challenging task in most BI projects large and small. Self-service data provisioning aims at tackling this problem by providing intuitive datasets discovery, acquisition and integration techniques to the end user.

The goal of this thesis is to provide a framework that enables self-service data provisioning in the enterprise. This framework empowers business users to search, inspect and reuse data through semantically enriched datasets profiles.

Publicly available datasets contain knowledge from various domains such as encyclopedic, government, geographic, entertainment, publications and so on. The increasing diversity of these datasets makes it difficult to annotate them with a fixed number of pre-defined tags. Moreover, manually entered tags are subjective and may not capture their essence and breadth. We propose a mechanism to automatically attach meta information to data objects by leveraging knowledge bases like DBpedia and Freebase.

In many knowledge bases, data entities are described with numerous properties. However, not all properties have the same importance. Some properties are considered as keys for performing instance matching tasks while other properties are generally chosen for quickly providing a summary of the key facts attached to an entity. Business users may want to enrich their reports with these data entities. To facilitate this, we propose a mechanism to select what properties should be used when augmenting extra columns into an existing dataset or annotating instances with semantic tags.

Linked Open Data (LOD) has emerged as one of the largest collections of interlinked datasets on the web. In order to benefit from this mine of data, one needs to access to descriptive information about each dataset (or metadata). This metadata enables dataset discovery, understanding, integration and maintenance. Data portals, which are considered to be datasets' access points, offer metadata represented in different and heterogeneous models. We first propose a harmonized dataset model

based on a systematic literature survey that enables complete metadata coverage to enable data discovery, exploration and reuse.. Second, we discovered that rich metadata information is currently very limited to a few data portals where they are usually provided manually, thus being often incomplete and inconsistent in terms of quality. We propose a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. This approach applies several techniques 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.

Traditional data quality is a thoroughly researched field with several benchmarks and frameworks to grasp its dimensions. Ensuring data quality in Linked Open Data is much more complex. It consists of structured information supported by models, ontologies and vocabularies and contains queryable endpoints and links. We propose an objective assessment framework for Linked Data quality based on quality metrics that can be automatically measured. We further present an extensible quality measurement tool that helps on one hand data owners to rate the quality of their datasets and get some hints on possible improvements, and on the other hand data consumers to choose their data sources from a ranked set.

Finally, the Internet has created a paradigm shift in how we consume and disseminate information. Data nowadays is spread over heterogeneous silos of archived and live data. People willingly share data on social media by posting news, views, presentations, pictures and videos. We propose a service that brings relevant, live and archived information to the user. The key advantage is an instantaneous access to complementary information without the need to search for it. Information appears when it is relevant enabling the user to focus on what is really important.

Contents

Acknowledgements . . . ii
Abstract . . . ii
Contents . . . v
List of Figures . . . ix
List of Tables . . . xi
List of Listings . . . xii
List of Publications . . . xiv
Acronyms . . . xv

1 Introduction **1**
1.1 Context and Motivation . . . 1
1.2 Use Case Scenario . . . 2
1.3 Research Challenges . . . 3
1.3.1 Dataset Integration and Enrichment . . . 3
1.3.2 Dataset Maintenance & Discovery . . . 4
1.3.3 Dataset Quality: . . . 5
1.4 Thesis Contributions . . . 5
1.4.1 Contributions on Dataset Integration and Enrichment . . . 5
1.4.2 Contributions on Dataset Maintenance & Discovery . . . 7
1.4.3 Contributions on Dataset Quality Control . . . 7
1.5 Thesis Outline . . . 7

I Towards A Complete Dataset Profile **10**

2 Background **13**
2.1 Semantic Web . . . 13
2.1.1 Resource Description Framework (RDF) . . . 14
2.1.2 RDF Schema . . . 16
2.1.3 Web Ontology Language . . . 17
2.1.4 SPARQL Query Language . . . 17
2.1.5 Linked Data . . . 17
2.2 Open Data . . . 19
2.2.1 Open Licenses . . . 21
2.3 Data Profiling . . . 22
2.4 Data Portals . . . 23

3 Dataset Profiles and Models **25**
3.1 Data Portals and Dataset Models . . . 25
3.1.1 DCAT . . . 26
3.1.2 DCAT-AP . . . 26
3.1.3 ADMS . . . 26
3.1.4 VoID . . . 27
3.1.5 CKAN . . . 27
3.1.6 DKAN . . . 28
3.1.7 Socrata . . . 29
3.1.8 Junar . . . 29
3.1.9 INSPIRE metadata . . . 29
3.1.10 Schema.org . . . 29
3.1.11 Common Core Metadaa Schema (CCMS) . . . 30
3.2 Metadata Classification . . . 30
3.3 Towards A Harmonized Model . . . 33
3.4 Summary . . . 39

4 Dataset Profiles Generation and Validation **40**
4.1 Introduction . . . 40
4.2 Motivation . . . 41
4.3 Related Work . . . 42
4.4 Profiling Data Portals . . . 44
4.4.1 Data Portal Identification . . . 45
4.4.2 Metadata Extraction . . . 46
4.4.3 Instance and Resource Extraction . . . 47
4.4.4 Profile Validation . . . 48
4.4.5 Profile and Report Generation . . . 50
4.5 Experiments and Evaluation . . . 51
4.5.1 Experimental Setup . . . 52
4.5.2 Profiling Correctness . . . 53
4.5.3 Profiling Completeness . . . 54
4.6 Analyzing Profiling Results . . . 54
4.6.1 General information . . . 55
4.6.2 Access information . . . 56
4.6.3 Ownership information . . . 56
4.6.4 Provenance information . . . 57
4.6.5 Enriched Profiles . . . 57
4.7 Summary . . . 58

5 Objective Linked Data Quality Assessment 60
5.1 Introduction . . . 60
5.2 Data Quality Assessment . . . 61
5.3 Objective Linked Data Quality Classification . . . 63
5.3.1 Completeness . . . 66
5.3.2 Availability . . . 67
5.3.3 Correctness . . . 67
5.3.4 Consistency . . . 67
5.3.5 Freshness . . . 68
5.3.6 Provenance . . . 68
5.3.7 Licensing . . . 68
5.3.8 Comprehensibility . . . 68
5.3.9 Coherence . . . 68
5.3.10 Security . . . 69
5.4 Linked Data Quality Tools . . . 69
5.4.1 Information Quality . . . 69
5.4.2 Modeling Quality . . . 69
5.4.3 Dataset Quality . . . 71
5.4.4 Queryable End-point Quality . . . 75
5.5 An Extensible Objective Quality Assessment Framework . . . 75
5.5.1 Quality Score Calculation . . . 76
5.5.2 Evaluation . . . 77
5.5.3 Experiments and Analysis . . . 78
5.6 Roomba vs. state of the art . . . 80
5.7 Summary . . . 80

II Data Integration in the Enterprise 83

6 Background 86
6.1 Data Integration . . . 86
6.1.1 Data Warehousing (DW) . . . 86
6.2 Business Intelligence . . . 87
6.2.1 Multidimensional Model . . . 87
6.2.2 SAP BI Application Suite . . . 88
6.3 SAP High Performance Analytic Appliance (HANA) . . . 88
6.3.1 HANA XS-Engine . . . 90
6.3.2 Active Information Store (AIS) . . . 90
6.4 Social Web . . . 91

7 Data Integration in the Enterprise **93**
7.1 Enterprise Knowledge Bases . . . 94
7.1.1 Entity Disambiguation with DPbedia in SAP HANA . . . 95
7.1.2 Important Properties for Entities . . . 96
7.2 Enhancing Schema Matching . . . 100
7.2.1 Related Work . . . 100
7.2.2 Proposition . . . 101
7.2.3 Activity Flow . . . 102
7.2.4 Data Reconciliation . . . 103
7.2.5 Matching Unnamed and Untyped Columns . . . 103
7.2.6 Column Labeling . . . 105
7.2.7 Handling Non-String Values . . . 106
7.2.8 Experiments . . . 106
7.3 Summary . . . 112

8 Semantic Social News Aggregation **114**
8.1 Introduction . . . 114
8.2 Underlying Mechanism . . . 115
8.2.1 Document Handler . . . 115
8.2.2 Query Layer . . . 117
8.2.3 Data Parser . . . 119
8.3 Front-End . . . 121
8.4 Summary . . . 121

9 Conclusions and Future Perspectives **123**
9.1 Achievements . . . 123
9.2 Perspectives . . . 123

A HDL - A Harmonized Dataset Model **124**

B Installation and cutomization instructions **129**
B.1 Installation and cutomization instructions for Roomba . . . 129
B.2 Customizing Roomba . . . 129
B.2.1 Localization . . . 130
B.3 Installation and cutomization instructions for Knowledge Graph Scraper 130
B.3.1 Customization . . . 131

C Roomba Results **133**
C.1 Roomba profiling report for the LOD Cloud . . . 133
C.2 Roomba quality profiling report for the LOD Cloud . . . 136

D DBpedia ranked properties in Fresnel vocabulary **137**

E SAP HANA Semantic Services API **141**
E.1 XSJS API Implementation 141
E.2 API Documentation 143
E.2.1 Entity Enrichment 143
E.2.2 Entity Disambiguation (`disambiguate.xsjs`) 144
E.2.3 Schema Matching (`matchTables.xsjs`) 145

F Source code for mappings **147**
F.1 Open Licenses Mappings 147
F.2 Semantic Social News Aggregation Mappings 149

Bibliography **153**

List of Figures

1.1 Processing pipeline for enabling self-service data provisioning 6

2.1 Example of RDF representation of an address 15
2.2 The LOD cloud as of May, 2007 18
2.3 Open Data ecosystem 19
2.4 Heat map of Open Government Data adoption according to the Open Data Barometer 2015 21
2.5 Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak colored by licensing types. http://www.cosasbuenas.es/blog/how-o-is-lod-2015 22

3.1 CKAN data model 28
3.2 Information sections and groups across data models 33

4.1 Processing pipeline for validating and generating dataset profiles . . . 45
4.2 LOD Cloud error % by section 57
4.3 LOD Cloud error % by information type 58

5.1 Average Error % per quality indicator for LOD group 78

6.1 Example of a hierarchical geography dimension 88
6.2 SAP's High Performance Analytic Appliance (HANA) in SAP BI suite 89
6.3 AIS data model 91

7.1 Google knowledge graph panel for the city of Nice, France 97
7.2 RUBIX Activity Workflow 102
7.3 Screenshot showing the results mapping view in RUBIX 103

8.1 SNARC's Document Handler 115
8.2 SNARC's Query Layer 118
8.3 SNARC's Data Parser 120
8.4 SNARC's User Interface - The Google Chrome Extension 121

List of Tables

3.1 Harmonized Dataset Models Mappings 36

4.1 Summary of the experiments details 53
4.2 Datasets chosen for the correctness evaluation 53
4.3 Groups chosen for the correctness evaluation 53
4.4 Top metadata fields error % by type 55

5.1 Objective Linked Data Quality Framework 64
5.2 Objective Quality Assessment Methods for CKAN-based Data Portals 76
5.3 Datasets chosen for the correctness evaluation 77
5.4 Functional Comparison of Automatic Linked Data quality Tools . . . 81

7.1 Tables structure for DBpedia in HANA column store 95
7.2 Results of combining text search score with the number of incoming associations for query "apple" 96
7.3 Results of the enhanced equation 7.2 and its affect on the overall score for query "apple" 96
7.4 Agreement on properties between users and the Knowledge Graph Panel 99
7.5 Similarity Scores Using the AMC Default Matching Algorithms 108
7.6 Similarity Scores Using the AMC Default Matching Algorithms + Cosine Similarity Method 109
7.7 Similarity Scores Using the AMC Default Matching Algorithms+ the PPMCC Similarity Method 110
7.8 Similarity Scores Using the AMC Default Matching Algorithms + Spearman Similarity Method 111
7.9 Similarity Scores Using the Combination of Cosine, PPMCC and AMC's defaults 111

Listings

3.1 Excerpt of the report for aggregating *extras* field values on the LOD Cloud . . . 35
3.2 Result for aggregating *resource_type* field values on the LOD Cloud . . 35
4.1 Excerpt of a dataset profile in CKAN standard model . . . 47
4.2 License mapping file sample . . . 49
4.3 Excerpt of the DBpedia validation report . . . 50
5.1 Excerpt of the LOD cloud group quality report . . . 79
A.1 The suggested harmonized dataset model (HDL) . . . 124
B.1 Roomba's customization via the options file . . . 129
B.2 Roomba's localization file example . . . 130
B.3 Roomba's localization file example . . . 131
B.4 Roomba's localization file example . . . 132
C.1 The result of running Roomba on the LOD Cloud group. Note that some URLs are cut for display purposes . . . 133
C.2 The result of running Roomba quality profiler on the LOD Cloud group 136
D.1 An excerpt of the Fresnel vocabulary for top properties mappings of DBpedia 3.9 . . . 137
E.1 Entity description API call return values sample . . . 143
E.2 API call paramteres for entity enrichment . . . 143
E.3 Entity enrichment API call return values sample . . . 143
E.4 API call paramteres for entity disambiguation . . . 144
E.5 Entity disambiguation API call return values sample . . . 145
E.6 API call paramteres for schema matching . . . 145
E.7 Schema matching API call return values sample . . . 145
F.1 The mappings of the Open Licenses for the LOD Cloud on the Datahub 147
F.2 The mappings of YouTube categories with DMOZ and Alchemy API . 149
F.3 The mappings of the StackExchange services with DMOZ and Alchemy API . . . 151

List of Publications

Journal

1. **Ahmad Assaf**, Raphaël Troncy and Aline Senart: **Towards An Objective Assessment Framework for Linked Data Quality**. International Journal On Semantic Web and Information Systems, *under review*, 2015.

Conferences

1. **Ahmad Assaf**, Raphaël Troncy and Aline Senart: **Automatic Validation, Correction and Generation of Dataset Metadata - Enhancing Dataset Search and Spam Detection**. In 24th International World Wide Web Conference, Demo Track, May 2015, Florence, Italy.

2. **Ahmad Assaf**, Ghislain Atemezing, Raphaël Troncy and Elena Cabrio: **What are the important properties of an entity? Comparing users and knowledge graph point of view**. In 11th Extended Semantic Web Conference (ESWC 2014), Demo Track, May 2014, Heraklion, Crete.

3. **Ahmad Assaf**, Aline Senart and Raphaël Troncy: **SNARC - An Approach for Aggregating and Recommending Contextualized Social Content**. In 10th Extended Semantic Web Conference (ESWC 2013), Sattelite Events, May 2013, Montpellier, France. **1^{st} Prize Winner of the AI Mashup Challenge**

Workshops

1. **Ahmad Assaf**, Raphaël Troncy and Aline Senart: **What's up LOD Cloud - Observing The State of Linked Open Data Cloud Metadata**. In 2nd Workshop on Linked Data Quality (LDQ), May 2015, Portoroz, Slovenia. **Best paper award**

2. **Ahmad Assaf**, Raphaël Troncy and Aline Senart: **HDL - Towards A Harmonized Dataset Model for Open Data Portals**. In 2nd International Workshop on Dataset PROFIling & fEderated Search for Linked Data (PROFILES), May 2015, Portoroz, Slovenia.

3. **Ahmad Assaf**, Raphaël Troncy and Aline Senart: **An Extensible Framework to Validate and Build Dataset Profiles**. In 2nd International Workshop on Dataset PROFIling & fEderated Search for Linked Data (PROFILES), May 2015, Portoroz, Slovenia. **Best paper award**

4. **Ahmad Assaf**, Aline Senart and Raphaël Troncy: **Data Quality Principles in the Semantic Web**. International Workshop on Data Quality Management and Semantic Technologies, July 2012, Palermo, Italy.

5. **Ahmad Assaf**, Eldad Louw, Aline Senart, Corentin Follenfant Raphaël Troncy and David Trastour: **RUBIX: a Framework for Improving Data Integration with Linked Data**. In 1st International Workshop on Open Data (WOD), June 2012, Nantes, France.

Glossary

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first appears in the text.

AIS	Active Information Store
AMC	Auto Mapping Core
API	Application Programming Interface
BI	Business Intelligence
CCMS	Common Core Metadaa Schema
CRM	Customer Relationships Management
CSV	Comma Separated Values
DI	Data Integration
DMS	Data Management Systems
DW	Data Warehousing
ERP	Enterprise Resource Planning
ETL	Extract-Transform-Load
FOAF	Friend of a friend
GA	Genetic Algorithm
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IR	Information Retrieval
JSON	JavaScript Object Notation
KB	Knowledge Base
LD	Linked Data
LDA	Latent Dirichlet Allocation
LOD	Linked Open Data
ML	Machine Learning
NE	Named Entity
NER	Named Entity recognition
NERD	Named Entity Recognition and Disambiguation
NLP	Natural Language Processing
OBD	Open Business Data
OD	Open Data
OGD	Open Government Data
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing

OWL	Web Ontology Language
POD	Project Open Data
PPMCC	Pearson Product-Moment Correlation Coefficient
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
SaaS	Software-as-a-Service
SAP HANA	SAP High Performance Analytic Appliance
SCM	Supply Chain Management
SKOS	Simple Knowledge Organization System
SOA	Service Oriented Architecture
SPARQL	Protocol and RDF Query Language
URI	Universal Resource Identifier
URL	Universal Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Chapter 1

Introduction

"More data usually beats better algorithms"
Anand Rajaraman

Business Intelligence (BI) has always been about creating new insight for business by converting data into meaning that can be shared between people to drive change in the organization. One key aspect of creating meaning is to have a common shared understanding of information also known as Semantics.

Classic BI and even the newer Agile Visualization tools focus much of their selling features on attractive and unique visualizations. Preparing data for those visualizations however still remains the far more challenging task in most BI projects large and small. The ultimate goal of BI is to facilitate efficient decisions while eliminating some of the IT headache. Traditionally, BI approaches have been controlled by a centralized version of truth with a wall between IT and the business. Self-service data provisioning aims at removing this wall by providing intuitive dataset discovery, acquisition and integration techniques intuitively to the end user.

1.1 Context and Motivation

Enterprises use a wide range of heterogeneous information systems in their business activities such as Enterprise Resource Planning (ERP), Customer Relationships Management (CRM) and Supply Chain Management (SCM) systems. An enterprise distributed IT landscape contains multiple systems using different technologies and data standards [109]. In addition to this heterogeneity, the amount of information in enterprise databases and on-line data stores expands exponentially each year. Enterprise Big Data isn't big in volume only, but in the associated file formats. The information is also often stored in unstructured and unknown formats.

Data integration is challenging as it requires combining data residing at different sources, and providing the user with a unified view of these data [96]. In large enterprises, it is a time and resource costly task. Various approaches have been introduced to solve this integration challenge. These approaches were primarily based on XML as the data representation syntax, Web Services to provide the data exchange protocols and Service Oriented Architecture (SOA) as a holistic approach for distributed

systems architecture and communication. However, it was found that these technologies are no sufficient to solve the integration problems in large enterprises [53, 54]. Recently, ontology-based data integration approaches have been suggested where ontologies are used to describe the data, queries and mappings between them [147]. A slightly different approach is the use of the Linked Data paradigm [22] for integrating enterprise data. Enterprises like Google and Microsoft are not only using the Linked Data integration paradigm for their information systems, but are also aiming at building enterprise knowledge bases (like the Google Knowledge Graph powered in part by Freebase[1]) that act as a crystallization point for their structured data.

Data becomes more useful when it is open, widely available, in shareable formats and when advanced computing and analysis can yield from it. The quality and amount of structured knowledge available on the web make it now feasible for companies to mine this huge amount of public data and integrate it in their next-generation enterprise information management systems. An example of this external data is the Linked Open Data (LOD) cloud. From 12 datasets cataloged in 2007, it has grown today to nearly 1000 datasets containing more than 82 billion triples[2] [22]. Data is being published by both the public and private sectors and covers a diverse set of domains from life sciences to media or government data. The LOD cloud is potentially a gold mine for organizations and individuals who are trying to leverage external data sources in order to produce more informed business decisions [29]. This external data can be accessed through public data portals like `datahub.io` and `publicdata.eu` or private ones like `quandl.com` and `enigma.io`. Analyzing this new type of data within the context of existing enterprise data should bring them new or more accurate business insights and allow better recognition of sales and market opportunities [93].

1.2 Use Case Scenario

To enable wide scale and efficient integration of data, there are some efforts needed from various sides. In this thesis, we tackle the issues and challenges from the point of views of two personae:

- **Data Analyst:** A Data Analyst is an experienced professional who is able to collect and acquire data from multiple data sources, filter and clean data, interpret and analyze results and provide ongoing reports.
- **Data Portal Administrator:** A Data Portal Administrator monitors the overall health of the portal. He oversees the creation of users, organizations

[1] http://freebase.com
[2] http://datahub.io/dataset?tags=lod

and datasets. Administrators try to ensure a certain data quality level by continuously checking for spam and manually enhancing dataset descriptions and annotations.

Throughout this thesis, we will use a use case scenario to illustrate the challenges and solutions that we provide.

In our scenario, **Bob** is a Data Analyst working with the Ministry of Transport in France. His favorite tool for crunching, manipulating and visualizing data is SAP Lumira[3], a self-service data visualization tool that makes it easy to import data from multiple sources, perform visual BI analysis using intuitive dashboards, interactive maps, charts, and infographics. Bob receives a memo from the management to create a report comparing the number of car accidents that occurred in France for that year, to its counterpart in the United Kingdom (UK). In addition, he is asked to highlight accidents related to illegal consumption of Alcohol in both countries.

After examining the ministry's records, Bob was able to collect the data needed to create his report for the French side. Bob issues an official request to the Department of Transport in UK to collect the data needed. However, Bob knows that the process takes long time and the management needs the report within days. Bob is familiar with the Open Data movement and starts his journey searching through different data portals in the UK.

Mark is a Data Portal Administrator for the `data.gov.uk`. He continuously oversees the processes of acquiring, preparing and publishing datasets. Mark always tries to ensure that the data published is of high quality and contains sufficient attached metadata to easily enable search and discovery. Mark often receives complaints about inaccurate or spam datasets. He manually removes and fix errors while keeping open communication channels with the data-publishing departments.

1.3 Research Challenges

In the scenario presented above, both publishers (Data Portal Administrators) and users (Data Analysts) need pragmatic solutions that help them in their tasks. To enable that, there are some challenging research questions that have to be addressed. These challenges are organized in three main types as the following:

1.3.1 Dataset Integration and Enrichment

- The enterprise heterogeneous data sources raise tremendous challenges. They have inherently different file formats, access protocols or query languages. They

[3] http://saplumira.com/

possess their own data model with different ways of representing and storing the data. Data across these sources may be noisy (e.g. duplicate or inconsistent), uncertain or be semantically similar yet different [14]. **Mark** needs powerful tools to map and organize the data in order to have a unified view for these heterogeneous and complex data structures.

- Attaching metadata and Semantic information to instances can be tricky. An entity is usually not associated with a single generic type in the knowledge base, but rather with a set of specific types which can be relevant or not given the context. **Mark** is challenged with finding the most relevant entity type within a given context.

- Entities play a key role in knowledge bases in general and in the Web of Data in particular. Entities like those in DBpedia, are generally described with a lot of properties. However, it is difficult for **Bob** to assess which ones are more "important" than others for particular tasks such data augmentation and visualizing the key facts of an entity.

- Social Networks are not just gathering Internet users into groups of common interests, they are also helping people follow breaking news, contribute to online debates or learn from others. They are transforming Web usage in terms of users' initial entry point, search, browsing and purchasing behavior. However, Integrating information from these Social Networks can be tricky to **Mark** due to the vast amount of data available which makes hard to spot what is relevant in a timely manner.

1.3.2 Dataset Maintenance & Discovery

- Even though popular datasets like DBPedia[4] and Freebase are well known and widely used, there are other hidden useful datasets not being used. Indeed these datasets may be useful for specialized domains, however without proper registry of topics, it is difficult for data analysts like **Bob** to find them [90].

- The growing amount of data requires rich metadata in order to reach its full potential. This metadata enables dataset discovery, understanding, integration and maintenance. Despite the various models and vocabularies describing datasets metadata, the ability to have an overview of the dataset by inspecting its metadata can be limited. For example, **Bob** had difficulty finding datasets with a specific geographical coverage as this information was missing from almost all of the examined datasets profiles.

[4] http://dbpedia.org

- Users, organizations and governments are empowered to publish datasets. However, data portal administrators like **Mark** need to continuously and manually check portals to detect Spam and maintain high quality data.

1.3.3 Dataset Quality:

Linked Data consists of structured information supported by models, ontologies and vocabularies and contains query endpoints and links. This makes data quality assurance a challenge. Despite the fact that Linked Open Data quality is a trending and highly demanded topic, very few efforts are currently trying to standardize, track and formalize frameworks to issue scores or certificates that will help data consumers in their integration tasks. Data portal administrators like **Mark** to have an overall view of their portals quality and want to incorporate such metrics in the existing dataset profiles. On the other hand, data analysts and users like **Bob** want to know beforehand if the dataset on hand is of a certain degree of quality to be used in their reports.

1.4 Thesis Contributions

In this thesis, we propose a framework (see Figure 1.1) to enable self-service data provisioning for internal and external data sources. The framework contributes to the three main challenges described above. In summary, the main contributions of this work are as follows:

1.4.1 Contributions on Dataset Integration and Enrichment

Regarding this aspect of our research, we have achieved the following tasks:

- We created a framework called RUBIX that enables mashing-up potentially noisy enterprise data and external data. The framework leverages reference knowledge bases to annotate data with a set of semantic concepts (metadata). One of the advantages of this metadata is to enhance the matching process of heterogeneous data sources within an enterprise.

- The metadata attached by RUBIX can be further used to enrich existing datasets. However, concepts are often represented with a large set of properties. To better recommend the top "important" properties for a concept, we reversed engineer the choices made by Google when creating knowledge graph panels and compared them to preferences obtained from a user survey. We further represented these choices explicitly using the Fresnel vocabulary, so that

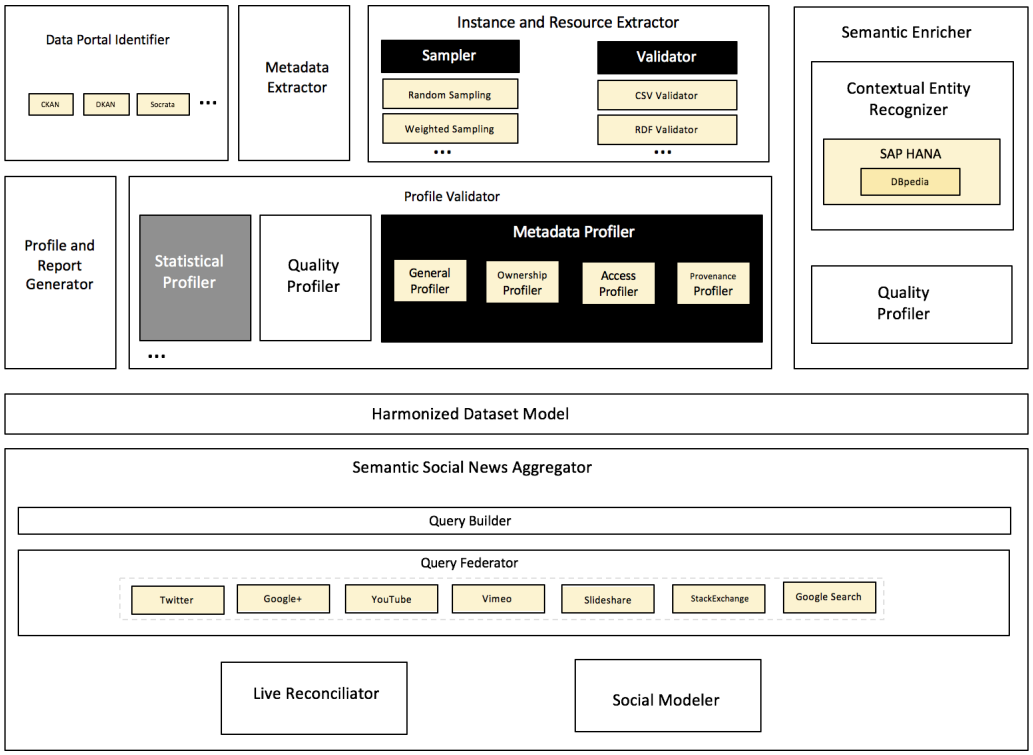

Figure 1.1: Processing pipeline for enabling self-service data provisioning

any application could read this configuration file for deciding which properties of an entity is worth to enrich.

- We have analyzed the landscape of dataset profiling tools and discovered various gaps (see section 4.3). As a result, we propose Roomba, a scalable automatic framework for extracting, validating, correcting and generating descriptive linked dataset profiles (see section 4.4). Roomba applies several techniques 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. We present the results of running Roomba over various data portals while focusing on analyzing the LOD cloud group hosted in the Datahub (see section 4.6).

- Aggregating relevant social news is not an easy task. We provide an Application Programming Interface (API) that enables semantic social news aggregation called SNARC. We designed a sample frontend application leveraging SNARC's capabilities to enable users to discover relevant social news instantly.

1.4.2 Contributions on Dataset Maintenance & Discovery

- We surveyed the landscape of various models and vocabularies that describe datasets on the web. Since establishing a common vocabulary or model is the key to communication, we identified the need for an harmonized dataset metadata model containing sufficient information so that consumers can easily understand and process datasets (see section 3.1).

- We implemented a set of mappings between each properties of the surveyed models. This has lead to the design of 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 (see section 3.3).

1.4.3 Contributions on Dataset Quality Control

Concerning our contributions on Linked Data quality assessment, we have achieved the following tasks:

- We propose five principle classes to describe the quality of a particular linked dataset. For each class, we define the principles that are involved at all stages of the data management process. We further refine these principles and propose a comprehensive objective quality framework applied to the Linked Open Data (see section 5.3).

- Upon surveying the landscape of data quality tools, we noticed a lack in automatic tools to check the dataset quality especially in terms of its completeness, licensing and provenance measures (see section 5.4). As a result, we extend Roomba to perform a set of data quality checks on Linked datasets. Our extension covers most of the quality indicators with its focus on completeness, correctness, provenance and licensing (see section 5.5).

1.5 Thesis Outline

The work presented in this thesis first describes a standard model to represent dataset profiles. Then it focuses on techniques to automatically generate and validate these profiles.

The rest of this manuscript is composed of two major parts:

In part I, we focus on the development of a framework that automatically validates and generates dataset profiles. We highlight the extensibility of this framework and show the results of running it across various data portals. The contributions of this part have been published in [6, 8, 9, 10, 11, 12]

- **Chapter 2**, overviews the background of our work in data profiling and quality assurance. We first introduce the basic concepts in the Semantic Web and the important aspects related to (Linked) Open Data. Then, we describe the concepts of data profiling and data quality.

- **Chapter 3**, conducts a unique and comprehensive survey of seven metadata models: CKAN, DKAN, Public Open Data, Socrata, VoID, DCAT and Schema.org. We propose a Harmonized Dataset modeL (HDL) based on this survey. We describe use cases that show the benefits of providing rich metadata to enable dataset discovery, search and Spam detection.

- **Chapter 4**, emphasizes the need for tools that are able to identify various issues in this metadata and correct them automatically. We introduce Roomba, a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. Afterwards, we present the results of running our framework on prominent data portals and analyze the results.

- **Chapter 5**, surveys the landscape of Linked Data quality tools and build upon previous efforts with focus on objective data quality measures. We further present a comprehensive objective quality framework applied to the Linked Open Data. We identify several gaps in the current tools and find the need for a comprehensive evaluation and assessment framework for measuring quality on the dataset level. We extend Roomba to calculate 82% of the suggested datasets objective quality indicators.

In part II, we focus on the challenges of external data integration in the enterprise. We focus on the development of a semantic enrichment framework and show the advantages of such enrichments in enhancing schema matching results and data enrichment.The contributions of this part have been published in [5, 7]

- **Chapter 6**, overviews the background of our work in data integration and enrichment. We introduce the basic concepts in Business Intelligence and Data Warehousing and describe the various technologies and systems in SAP's ecosystem.

- **Chapter 7**, presents a framework that enables business users to semi-automatically combine potentially noisy data residing in heterogeneous silos. Semantically related data is identified and appropriate mappings are suggested to users. We also show that it is possible to reveal what are the "important" properties of entities by reverse engineering the choices made by Google when creating knowledge graph panels and by comparing users preferences obtained from a user survey.

- **Chapter** 8, emphasizes the need for tools that are able to aggregate relevant social news to a certain context. We introduce SNARC, a semantic social news aggregation framework that leverages live rich data that social networks provide to build an interactive rich experience on the Internet.

Part I

Towards A Complete Dataset Profile

Overview of Part I

In Part I, we focus on the development of a framework that automatically validates and generates dataset profiles. We highlight the extensibility of this framework and show the results of running it across various data portals.

In Chapter 2, we overview the background of our work in data profiling and quality assurance. We first introduce the basic concepts in the Semantic Web and the important aspects related to (Linked) Open Data. Then, we describe the concepts of data profiling and data quality.

In Chapter 3, we conduct a unique and comprehensive survey of seven metadata models: CKAN, DKAN, Public Open Data, Socrata, VoID, DCAT and Schema.org. We propose a Harmonized Dataset modeL (HDL) based on this survey. We describe use cases that show the benefits of providing rich metadata to enable dataset discovery, search and Spam detection.

In Chapter 4, we note the need for tools that are able to identify various issues in this metadata and correct them automatically. We introduce Roomba, a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. Afterwards, we present the results of running our framework on prominent data portals and analyze the results.

In Chapter 5, we survey the landscape of Linked Data quality tools and build upon previous efforts with focus on objective data quality measures. We further present a comprehensive objective quality framework applied to the Linked Open Data. We identify several gaps in the current tools and find the need for a comprehensive evaluation and assessment framework for measuring quality on the dataset level. We extend Roomba to calculate 82% of the suggested datasets objective quality indicators.

Chapter 2

Background

2.1 Semantic Web

The web can be seen as a worldwide, distributed system of interconnected documents that humans can read, exchange and discuss. The original model behind the web can be roughly summarized as a way to publish documents represented in a standard way (e.g. HTML), containing links to other documents accessible through standard protocols (e.g. HTTP).

The great advantage of the web is that it abstracts the physical storage and network layers involved in the information exchange between machines. This enables documents to appear directly connected to one another. However, in this paradigm machines are not able to achieve tasks based on automated data processing such as search and query answering. To overcome this limitation, research fields such as Information Retrieval (IR), Machine Learning (ML), and Natural Language Processing (NLP) produced complex systems trying to automatically extract meaning from unstructured data. A typical example would be search engines such as Yahoo[1] and Google[2]. Despite their success, there is still a semantic gap between what the machine understands and how the user perceives the data [110]. This is where Semantic Web intervenes trying to fill the knowledge gap. In the same way that original Web abstracted away the network and physical layers, the Semantic Web abstracts away the document and application layers involved in the exchange of information. The Semantic Web connects facts, so that rather than linking to a specific document or application, you can instead refer to a specific piece of information contained in that document or application. Berners-Lee et al. [18] provide the following definition for the Semantic Web:

> *The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.*

The word semantic itself implies meaning or understanding. The fundamental

[1] http://www.yahoo.com
[2] http://www.google.com

differences between Semantic Web and other data-related technologies is that the Semantic Web is concerned with the meaning and not the structure of data. This fundamental difference engenders a completely different outlook on how storing, querying, and displaying information might be approached. Some applications, such as those that refer to a large amount of data from many different sources, benefit enormously from this feature.

What is meant by "semantic" in the Semantic Web is not that computers are going to understand the meaning of anything, but that the logical pieces of meaning can be mechanically manipulated by a machine to useful ends. For example, if a website publishes a database about a product line, with products and descriptions, while another publishes a database of product reviews. A third site for a retailer publishes a database of products in stock. The Semantic Web standards make it easier to write an application to mesh distributed databases together, so that a computer could use the three data sources together to help an end-user make better purchasing decisions.

Standards facilitate building applications, especially in decentralized systems. To realize the Semantic Web vision, a series of technologies and standards have been proposed. We describe some of these standards in the following:

2.1.1 Resource Description Framework (RDF)

Resource Description Framework (RDF) [92] is a recommendation of the World Wide Web Consortium (W3C) that describes the Web resources. It can be seen as the data modeling language for the Semantic Web.

Semantic Web resources can be anything that has an identity, they can be a person, document, image, location, etc. Each resource is assigned a Universal Resource Identifier (URI) [16] which is a Unicode string to identify an abstract or physical resource. The most common type of URI is the Universal Resource Locator (URL) which is used to identify Web resources. A special case of a resource is a blank node for which no URI or literal is given. Blank nodes denote the existence of resources with specific attributes but without providing any information about their identity or reference.

Resources can have atomic values named literal. They are simple Strings that describe data values that do not have a separate existence. They can be plain (simple string combined with an optional language tag (e.g. "thesis"@en)) or typed (string combined with a datatype URI and an optional language tag e.g. "0.99"^^datatype-URI). RDF reuses the XML Schema (W3C) datatypes[3] which can be string, integer, float, double or date, as defined by the XML Schema Datatype specification.

[3] http://www.w3.org/TR/xmlschema-2

RDF provides an intuitive knowledge representation using directed graphs, where the subjects and objects (resources) are the nodes and the predicates (properties) are the edges of that graph, this is referred to as an RDF Triple. Note that a property is a specific aspect, characteristic, attribute, or relation used to describe a resource [92]. Resources can be described and linked by other set of statements forming a larger graph or a semantic network. An atomic RDF statement is a triple which is usually denoted as $< s, p, o >$ and composed of:

- **Subject:** the URI of a resource or a blank node which the statement refers to.
- **Predicate:** describes a property of the subject and expresses the relationship between the subject and the object.
- **Object:** specifies the value of the property. It can be a URI of a resource, blank node or a literal.

Figure 2.1 depicts an example of RDF graph-based representation for an address. An address is a structure that consists of different values such as a street, a city, a state and a zip-code.

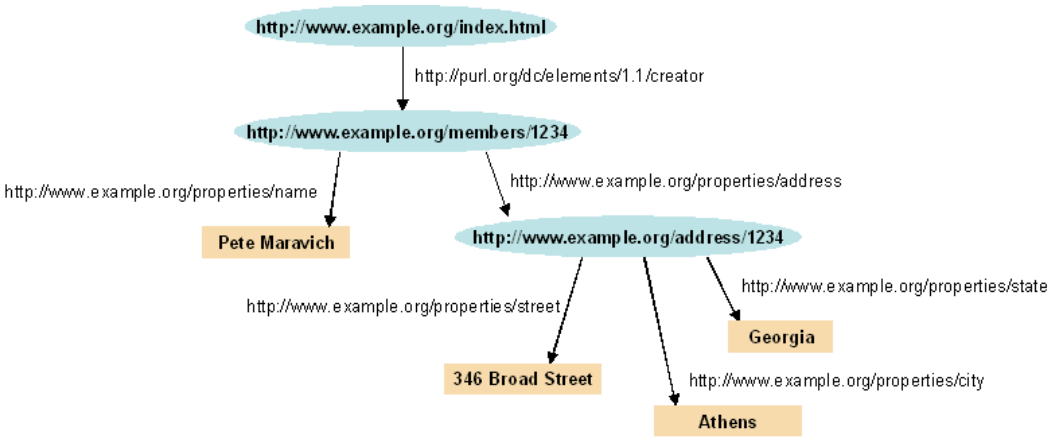

Figure 2.1: Example of RDF representation of an address

Several methods exist for serializing the RDF data model. The most common format is RDF/XML. There exist other text-based formats introduced by W3C such as Turtle[4] and N-Triples[5] which are easier to read than RDF/XML.

RDF also contains data structures (containers and collections) that allow aggregating nodes or facts together. They are basically a syntactic sugar that will ease the process of writing code with no semantic expressiveness whatsoever.

[4] http://www.w3.org/TeamSubmission/turtle
[5] http://www.w3.org/TR/n-triples

2.1.2 RDF Schema

"It's impossible to get everyone everywhere to agree on a single label for every specific thing that ever was, is, or shall be"
Cambridge Semantics [130]

RDF is a simple and flexible data model that describes resources using properties and values. Predicates in RDF are what describe and give meaning to statements. They act as a vocabulary or an ontology. An Ontology is an explicit specification of a conceptualization [59]. It is a formal way to organize knowledge and terms and reflect common understanding of a domain. Ontologies are typically represented as graphical relationships or networks as opposed to taxonomies which are usually presented hierarchically. Some core elements of an ontology are:

- Class: defines a concept, type or collection within a specific domain. It encapsulates objects sharing some properties. For instance, in a geographical domain, the class Country is more specialized than the class Place.
- Individual: also known as instance or object and is a member of a class. For instance, *France* is an instance of the class Country.
- Property: is a binary relation describing how classes and individuals relate to each other. A datatype property connects instances with RDF literals while object property connects instances of two classes. For example, *hasCity* is an object property that can relate two instances of the class City.

In order for Semantic Web applications to be able to share data, they must agree on common vocabulary. RDF doesn't provide ways to define those vocabularies and to specify domain specific classes and properties. To overcome this limitation, an extension of RDF called RDF Schema (RDFS) [30] provides a basic vocabulary to interpret RDF statements, describes taxonomies of classes and properties and defines very basic restrictions.

RDFS as a modeling language allows for: 1) definition of classes and their instantiation, 2) definition of properties and restrictions and 3) definition of hierarchies for classes and properties. In summary:

- Resources are instances of one or more class (*rdfs:class*). Classes are organized in a hierarchy using *rdfs:subClassOf* property.
- Properties have are assigned the class *rdf:Property* and are organized in a hierarchy using *rdfs:subPropertyOf*.

- Restrictions on properties can be specified. For example, *rdfs:domain* to define the class of the subject and *rdfs:range* to define the class of the object.

2.1.3 Web Ontology Language

RDFS provides basic hierarchies associated with simple restrictions. This limited expressivity triggered the need to define an explicit formal description of concepts in complex domains. As a result, the Web Ontology Language (OWL) [58] which adds more vocabulary for describing properties and classes on top of RDF is the current markup language endorsed by W3C. It provides more relations between classes (e.g. *disjointWith*), logical properties (e.g. *intersectionOf*, *sameAs*) and enumerations (e.g. *oneOf*, *allValuesFrom*), among others.

2.1.4 SPARQL Query Language

Relational databases can be efficient for semantic databases. However, in practice, they are designed for a different type of workload. The fundamental operation of semantic databases is join, which is naturally expensive in relational databases. Given that we have our data modeled as RDF regardless of the underlying database choice, it is now possible to query and ask questions about our data in a very powerful way. Protocol and RDF Query Language (SPARQL) [124] is the standardized query language for RDF.

A SPARQL query consists of a set of triples where each part (subject, predicate and/or object) can consist of variables alongside a set of conjunctions (e.g. logical "and") or disjunctions (e.g. logical "or"). It works by matching the triples in the query with the existing RDF triples and find solutions to the variables.

2.1.5 Linked Data

The traditional approach of sharing data through independent silos is diminishing with the various advances in the Web. The Semantic Web envisages the availability of large amount of interlinked RDF data. Linked Data (LD) is a major milestone towards achieving this vision. Formally, Linked Data has been defined as about "data published on the Web in such a way that it is machine readable, its meaning is explicitly defined, it is linked to other external datasets, and can in turn be linked to from external datasets" [22].

Linked Data follows four main principles outlined by Tim Berners-Lee [17] to publish information on the Web, which are:

1. Use URIs as names for things

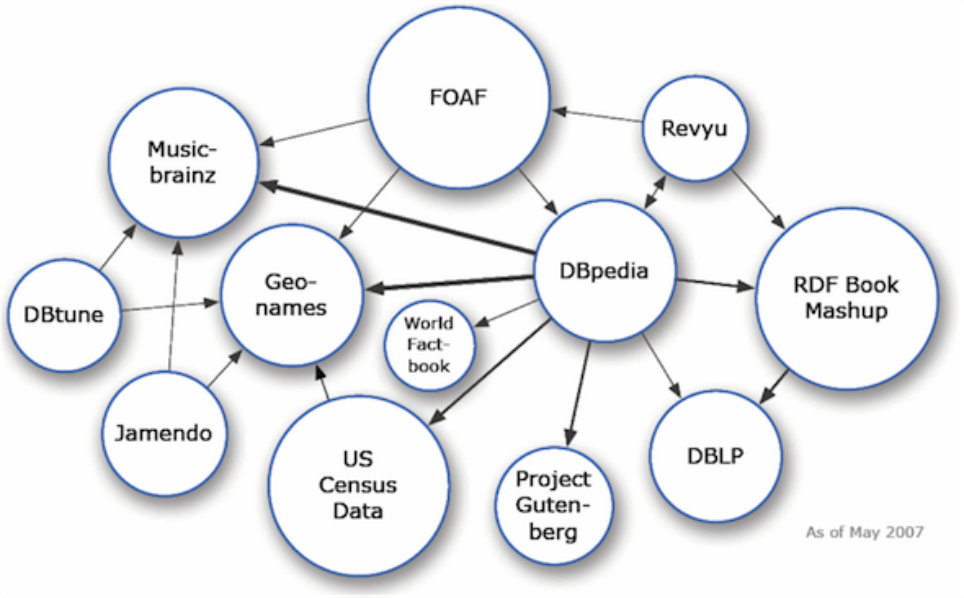

Figure 2.2: The LOD cloud as of May, 2007

2. Use HTTP URIs so that people can look up those names.

3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)

4. Include links to other URIs. so that they can discover more things.

Linked Data is continuously evolving, started in 2007 with a dozen of datasets (see Figure 2.2) to reach today thousands of datasets covering knowledge from various domains such as encyclopedic, government, geographic, entertainment, publications and so on. The datasets have tripled in size from 2011 to 2014, with a significant growth of nearly 271% [129]. The latest version published in April 2014 contains 1014 linked datasets connected by 2909 linksets (see Figure 2.5[6]).

One of the most widely used datasets is DBpedia[7]. It is a structured knowledge extracted from multilingual versions of Wikipedia [24]. At the time of writing, the English version of DBpedia consists of 470 millions RDF triples that describe 4.0 million things covering a wide range of topics, and contains 45 million RDF links to several hundred external datasets.

In order to achieve the Linked Data vision, datasets should contain outbound links to other datasets. Significant efforts try to automatically or semi-automatically generate these link to facilitate data discovery and to attach additional information.

[6]A more Web friendly version can be accessed at http://data.dws.informatik.uni-mannheim.de/lodcloud/2014/

[7]http://dbpedia.org

2.2 Open Data

Open Data (OD) is the data that can be easily discovered, accessed, reused and redistributed by anyone [41]. Open data should have both legal and technical dimensions. It should be placed in the public domain under liberal terms of use with minimal restrictions and should be available in electronic formats that are non-proprietary and machine readable. Similarly, Linked Open Data (LOD) refers to the semantically linked, machine-readable open data.

Businesses, citizens and governments are encouraged to publish, share and reuse data. Figure 2.3 shows the Open Data ecosystem described by [61]. Each party in this ecosystem supplies different types of data (e.g. Open Business Data (OBD), Open Government Data (OGD)) to different types of stakeholders.

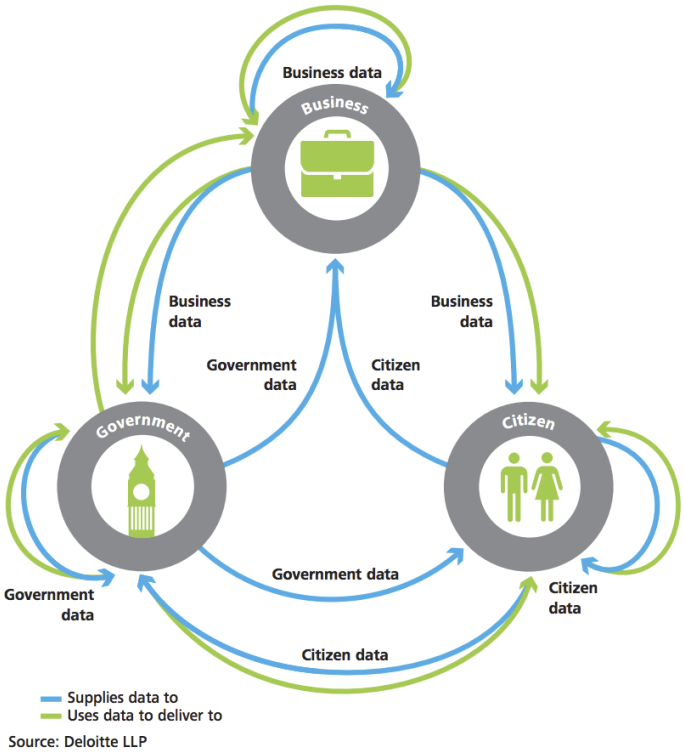

Figure 2.3: Open Data ecosystem

Similarly to the Linked Data principles, Tim Berners-Lee [17] outlined a 5 starts scheme to evaluate the availability of Linked Data as Linked Open Data:

1. Data available on the web in any format, even using PDF or image scan, but with an Open license.

2. Data delivered as machine-readable structured data, e.g. excel instead of image

scan of a table.

3. Data available in a non-proprietary format, e.g. CSV instead of excel.

4. All the above plus, data using open standards from W3C, e.g. RDF and SPARQL, to identify things and properties, so that people can point at other data.

5. All the above, plus, to link data to other people's data to provide context.

Open Data has major benefits for citizens, businesses, societies and governments. It increases transparency and enables self-empowerment by improving the visibility of previously inaccessible information; allows citizens to be better informed about policies, public spending and activities in the law making processes [61, 105].

Open Data is considered a gold mine for organizations which are trying to leverage external data sources in order to produce more informed business decisions [29]. Despite the legal issues surrounding Open Data licenses [77], McKinsey [105] estimates that Open Data in the health sector alone adds up over $300 billion to the economy every year.

These huge benefits led to a world-wide adoption of Open Data. Figure 2.4 shows the existence and support for open data initiatives, engagement with open data from outside government, legislative frameworks that support open data and the existence of training and support for data use and innovation [41]. Moreover, there are several reports and initiatives like Open Data Barometer[8], Open Data Monitor[9] and Global Open Data Index[10]that aim at analyzing and monitoring the adoption of Open Data across the world.

Going back to our scenario in 1.2, Open Data will help our analyst **Bob** in:

- Have a transparent view on the data available by Ministry of Transport in France. This helps in preventing the possibility of wasting time and funds recollecting data that has been already collected by a different department.

- Discover complementary datasets from other sources. The benefits of data transparency amplifies when it is widely adopted in all other departments and agencies.The additional data enrich reports and enable better-informed, data-driven decisions. For example, by providing extra details on traffic information at the time when accidents occurred, **Bob** could draw more accurate conclusions on the root cause of some of these accidents.

[8] http://barometer.opendataresearch.org/
[9] http://opendatamonitor.eu
[10] http://index.okfn.org/

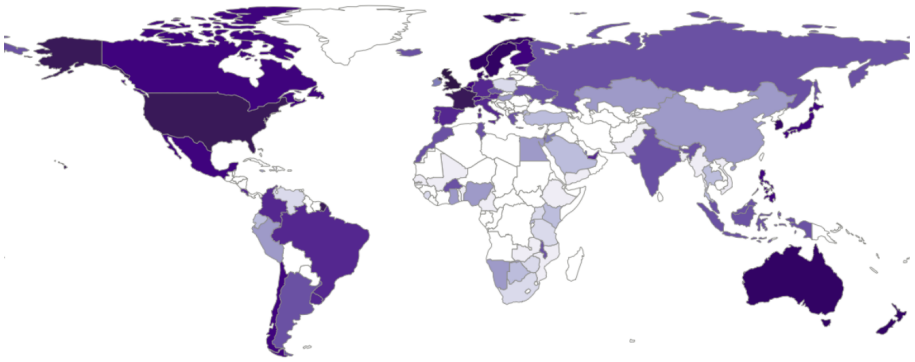

Figure 2.4: Heat map of Open Government Data adoption according to the Open Data Barometer 2015

2.2.1 Open Licenses

Project Open Data [11] emphasizes the importance of datasets reusability as one of the main principles for Open Data. Open data should be made available under an open license. This is of high importance specially for organizations looking to integrate data for commercial use.

The Open Definition [12] defines a license as the legal conditions under which an item or piece of knowledge (also referred to as "work") is made available. Domain dedications like *Creative Commons Zero* satisfy this definition although not technically a "license".

The Open Definition defines the following conditions for open licenses:

- Allow free use of the work without any fee arrangement or compensation.
- Allow redistribution (on its own or as part of a collection) of the work.
- Allow distribution of modified work under the same license of the original.
- Allow any part of the work to be freely used, distributed or modified separately.
- Allow distribution of the work alongside other distinct works without placing restrictions on the additional ones.
- Doesn't discriminate against any person or group.
- Allow use, redistribution, modification, and compilation for any purpose.
- Allow rights propagation to all to whom the work id distributed.

[11] https://project-open-data.cio.gov
[12] http://opendefinition.org/

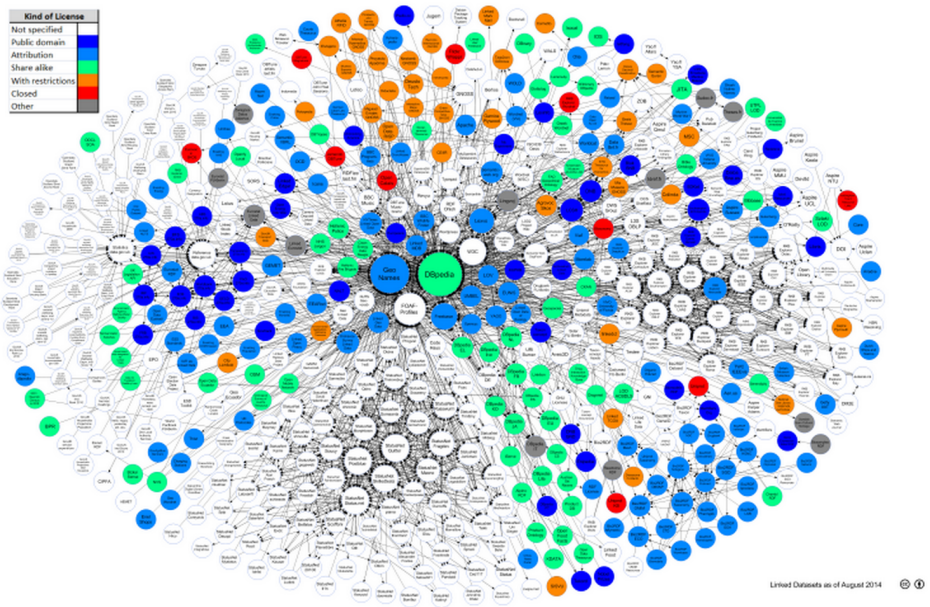

Figure 2.5: Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak colored by licensing types. http://www.cosasbuenas.es/blog/how-o-is-lod-2015

2.3 Data Profiling

The huge amount of published data makes it difficult to discover relevant datasets through traditional inspection of the raw data. *Data profiling* is the process of creating descriptive information and collect statistics about that data. It is a cardinal activity when facing an unfamiliar dataset [99, 83].

Data profiling is a vital task to monitor the quality of internal data in the enterprise. Halo BI report [73] states that nearly 40% of company's data is found to be inaccurate. 25% of which is considered critical data.

Data profiles reflect the importance of datasets without the need for detailed inspection of the raw data. It also helps in assessing the importance of the dataset, improving users' ability to search and reuse part of the dataset and in detecting irregularities to improve its quality. Data profiling includes typically several tasks:

- **Metadata profiling**: Provides general information on the dataset (dataset description, release and update dates), legal information (license information, openness), practical information (access points, data dumps), etc.

- **Statistical profiling**: Provides statistical information about data types and patterns in the dataset (e.g. properties distribution, number of entities and RDF triples).

- **Topical profiling**: Provides descriptive knowledge on the dataset content and structure. This can be in form of tags and categories used to facilitate search and reuse.

- **Quality profiling**: Discovers inconsistencies and anomalies in the data. Data is considered of high quality if is appropriate for use and if it correctly represents the world constructs to which it refers [101].

Dataset profiles are collections of data describing the internal structure of the dataset. They are presented as a set of metadata in different formats such as JSON, XML and RDF. The Linked Data publishing best practices [21] specifies that datasets should contain metadata needed to effectively understand and use them. *Metadata* is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource [123]. Having rich metadata helps in enabling:

- **Data discovery, exploration and reuse**: In [146], it was found that users are facing difficulties finding and reusing publicly available datasets. Metadata provides an overview of datasets making them more searchable and accessible. High quality metadata can be at times more important than the actual raw data especially when the costs of publishing and maintaining such data is high.

- **Organization and identification**: The increasing number of datasets being published makes it hard to track, organize and present them to users efficiently. Attached metadata helps in bringing similar resources together and distinguish useful links.

- **Archiving and preservation**: There is a growing concern that digital resources will not survive in usable forms to the future [123]. Metadata can ensure resources survival and continuous accessibility by providing clear provenance information to track the lineage of digital resources and detail their physical characteristics.

2.4 Data Portals

Data portals (or data catalogs) are the entry points to discover published datasets. They are a curated collections of datasets metadata that provide a set of complementary discovery and integrations services.

Data portals can be public like `Datahub.io` and `publicdata.eu` or private ones like `quandl.com` and `enigma.io`. Private portals harness manually curated data from various sources and expose them to users either freely or through

paid plans. Similarly, in some public data portals, administrators manually review datasets information, validate, correct and attach suitable metadata information. This information is mainly in the form of predefined tags such as *media, geography, life sciences* for organization and clustering purposes.

There are several Data Management Systems (DMS) that power public data portals. CKAN[13] is the world's leading open-source data portal platform powering web sites like DataHub, Europe's Public Data and the U.S Government's open data. Modeled on CKAN, DKAN[14] is a standalone Drupal distribution that is used in various public data portals as well. Socrata[15] helps public sector organizations improve data-driven decision making by providing a set of solutions including an open data portal. In addition to these tradition data portals, there is a set of tools that allow exposing data directly as RESTful APIs like `thedatatank.com`.

[13] http://ckan.org
[14] http://nucivic.com/dkan/
[15] http://www.socrata.com

Chapter 3

Dataset Profiles and Models

The value of Open Data is recognized when it is used. To ensure that, publishers need to enable people to find datasets easily. Data portals are specifically designed for this purpose. They make it easy for individuals and organizations to store, publish and discover datasets.

Data models vary across data portals. While exhaustively surveying the range of data models, we did not find any that offers enough granularity to completely describe complex datasets facilitating search, discovery and recommendation. For example, the Datahub uses an extension of the Data Catalog Vocabulary (DCAT) [102] which prohibits a semantically rich representation of complex datasets like DBpedia[1] that has multiple endpoints and thousands of dump files with content in several languages [32].

From our survey, we found that a proper integration of Open Data into businesses requires datasets to include the following information:

- *Access information*: a dataset is useless if it does not contain accessible data dumps or query-able endpoints;
- *License information*: businesses are always concerned with the legal implications of using external content. As a result, datasets should include both machine and human readable license information that indicates permissions, copyrights and attributions;
- *Provenance information*: depending on the dataset license, the data might not be legally usable if there are no information describing its authoritative and versioning information. Current models under-specify these aspects limiting the usability of many datasets.

3.1 Data Portals and Dataset Models

There are many data portals that host a large number of private and public datasets. Each portal present the data based on a model used by the underlying Data Man-

[1] http://dbpedia.org

agement Software. In this section, we present the results of our landscape survey of the most common data portals and dataset models.

3.1.1 DCAT

The Data Catalog Vocabulary (DCAT) is a W3C recommendation that has been designed to facilitate interoperability between data catalogs published on the Web [102]. The goal behind DCAT is to increase datasets discoverability enabling applications to easily consume metadata coming from multiple sources. Moreover, the authors foresee that aggregated DCAT metadata can facilitate digital preservation and enable decentralized publishing and federated search.

DCAT is an RDF vocabulary defining three main classes: `dcat:Catalog`, `dcat:Dataset` and `dcat:Distribution`. We are interested in both the `dcat:Dataset` class which is a collection of data that can be available for download in one or more formats and the `dcat:Distribution` class which describes the method with which one can access a dataset (e.g. an RSS feed, a REST API or a SPARQL endpoint).

3.1.2 DCAT-AP

The DCAT application profile for data portals in Europe (DCAT-AP)[2] is a specialization of DCAT to describe public section datasets in Europe. It defines a minimal set of properties that should be included in a dataset profile by specifying mandatory and optional properties. The main goal behind it is to enable cross-portal search and enhance discoverability. DCAT-AP has been promoted by the Open Data Support[3] to be the standard for describing datasets and catalogs in Europe.

3.1.3 ADMS

The Asset Description Metadata Schema (ADMS) [118] is also a profile of DCAT. It is used to semantically describe assets. An asset is broadly defined as something that can be opened and read using familiar desktop software (e.g. code lists, taxonomies, dictionaries, vocabularies) as opposed to something that needs to be processed like raw data. While DCAT is designed to facilitate interoperability between data catalogs, ADMS is focused on the assets within a catalog.

[2] https://joinup.ec.europa.eu/asset/dcat_application_profile/description
[3] http://opendatasupport.eu

3.1.4 VoID

VoID [27] is another RDF vocabulary designed specifically to describe linked RDF datasets and to bridge the gap between data publishers and data consumers. In addition to dataset metadata, VoID describes the links between datasets. VoID defines three main classes: `void:Dataset`, `void:Linkset` and `void:subset`. We are specifically interested in the `void:Dataset` concept. VoID conceptualizes a dataset with a social dimension. A VoID dataset is a collection of raw data, talking about one or more topics, originates from a certain source or process and accessible on the web.

3.1.5 CKAN

CKAN helps users from different domains (national and regional governments, companies and organizations) to easily publish their data through a set of workflows to publish, share, search and manage datasets. CKAN is the portal powering web sites like Datahub, the Europe's Public Data portal or the U.S Government's open data portal[4].

CKAN is a complete catalog system with an integrated data storage and powerful RESTful JSON API. It offers a rich set of visualization tools (e.g. maps, tables, charts) as well as an administration dashboard to monitor datasets usage and statistics. CKAN allows publishing datasets either via an import feature or through a web interface. Relevant metadata describing the dataset and its resources as well as organization related information can be added. A Solr[5] index is built on top of this metadata to enable search and filtering.

The CKAN data model[6] contains information to describe a set of entities (dataset, resource, group, tag and vocabulary). CKAN keeps the core metadata restricted as a JSON file, but allows for additional information to be added via "extra" arbitrary key/value fields. CKAN supports Linked Data and RDF as it provides a complete and functional mapping of its model to Linked Data formats. An extension called ckanext-dcat[7] provides plugins that allow CKAN to expose and consume metadata from other other catlogs using DCAT as their model.

[4] http://data.gov
[5] http://lucene.apache.org/solr/
[6] http://docs.ckan.org/en/ckan-1.8/domain-model.html
[7] https://github.com/ckan/ckanext-dcat

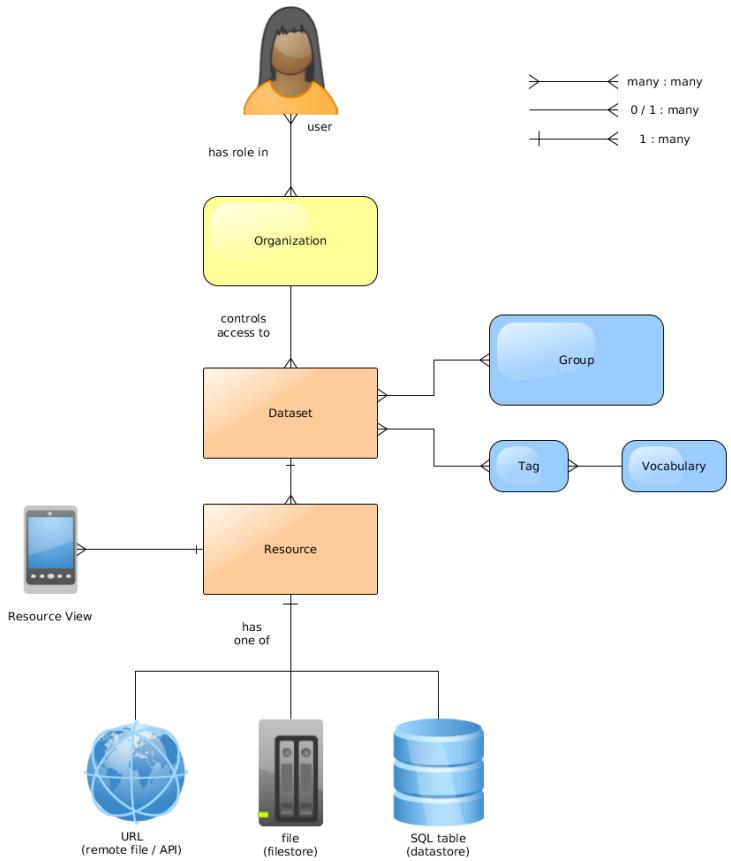

Figure 3.1: CKAN data model

3.1.6 DKAN

DKAN[8] is a Drupal-based DMS with a full suite of cataloging, publishing and visualization features. Built over Drupal, DKAN can be easily customized and extended. The actual data sets in DKAN can be stored either within DKAN or on external sites. DKAN users are able to explore, search and describe datasets through the web interface or a RESTful API.

The DKAN data model[9] is very similar to the CKAN one, containing information to describe datasets, resources, groups and tags.

[8] http://nucivic.com/dkan/

[9] http://docs.getdkan.com/dkan-documentation/dkan-developers/dataset-technical-field-reference/

3.1.7 Socrata

Socrata[10] is a commercial platform to streamline data publishing, management, analysis and reusing. It empowers users to review, compare, visualize and analyze data in real time. Datasets hosted in Socrata can be accessed using RESTful API that facilitates search and data filtering.

Socrata allows flexible data management by implementing various data governance models and ensuring compliance with metadata schema standards. It also enables administrators to track data usage and consumption through dashboards with real-time reporting. Socrata is very flexible when it comes to customizations. It has a consumer-friendly experience giving users the opportunity to tell their story with data. Socrata's data model is designed to represent tabular data: it covers a basic set of metadata properties and has good support for geospatial data.

3.1.8 Junar

Junar[11] adopts the Software-as-a-Service (SaaS) approach for data collection, enrichment, analysis and collaboration. Junar provides various functionalities that allow collaboration with colleagues to manage Open Data projects. Users are allowed to attach metadata to the information they publish to enhance search and discoverability.

3.1.9 INSPIRE metadata

The Infrastructure for Spatial Information in the European Community directive (INSPIRE)[12] aims at ensuring a compatible and usable spatial data infrastructure across the European Union.

The directive proposes a framework using a common metadata specification for data sharing, monitoring and reporting. The framework also defines rules to describe datasets and a set of implementation rules.

3.1.10 Schema.org

Schema.org[13] is a collection of schemas used to markup HTML pages with structured data. This structured data allows many applications, such as search engines, to understand the information contained in Web pages, thus improving the display of search results and making it easier for people to find relevant data.

[10] http://socrata.com
[11] http://junar.com/
[12] http://inspire.ec.europa.eu/index.cfm
[13] http://schema.org

Schema.org covers many domains. We are specifically interested in the `Dataset` schema. However, there are many classes and properties that can be used to describe organizations, authors, etc.

3.1.11 Common Core Metadaa Schema (CCMS)

Project Open Data (POD)[14] is an online collection of best practices and case studies to help data publishers. It is a collaborative project that aims to evolve as a community resource to facilitate adoption of open data practices and facilitate collaboration and partnership between both private and public data publishers.

The POD metadata model (CCMS)[15] is based on DCAT. Similarly to DCAT-AP, POD defines three types of metadata elements: Required, Required-if(conditionally required) and Expanded (optional). The metadata model is presented in the JSON format and encourages publishers to extend their metadata descriptions using elements from the "Expanded Fields" list, or from any well-known vocabulary.

3.2 Metadata Classification

A dataset metadata model should contain sufficient information so that consumers can easily understand and process the data that is described. After analyzing the most prominent models described in section 3.1, we find out that a dataset can contain four main sections:

- **Resources**: The actual raw data that can be downloaded or accessed directly via queryable endpoints. Resources can come in various formats such as JSON, XML or RDF.
- **Tags**: Descriptive knowledge about the dataset content and structure. This can range from simple textual representation to semantically rich controlled terms. Tags are the basis for datasets search and discovery.
- **Groups**: Groups act as organizational units that share common semantics. They can be seen as a cluster or a curation of datasets based on shared categories or themes.
- **Organizations**: Organizations are another way to arrange datasets. However, they differ from groups as they are not constructed by shared semantics or properties, but solely on the dataset's association to a specific administration party.

[14] http://project-open-data.cio.gov/
[15] https://project-open-data.cio.gov/v1.1/schema/

Upon close examination of the various data models, we grouped the metadata information into eight main types. Each section discussed above should contain one or more of these types. For example, resources have general, access, ownership and provenance information while tags have general and provenance information only. The eight information types are:

- **General information**: The core information about the dataset (e.g., title, description, ID). The most common vocabulary used to describe this information is Dublin Core[16].

- **Access information**: Information about dataset access and usage (e.g., URL, license title and license URL). In addition to the properties in the models discussed above, there are several vocabularies designed specially to describe data access right e.g. Linked Data Rights[17], the Open Digital Rights Language (ODRL)[18].

- **Ownership information**: Authoritative information about the dataset (e.g. author, maintainer and organization). The common vocabularies used to expose ownership information are Friend-of-Friend (FOAF)[19] for people and relationships, vCard [72] for people and organizations and the Organization ontology [126] designed specifically to describe organizational structures.

- **Provenance information**: Temporal and historical information about the dataset creation and update records, in addition to versioning information (e.g. creation data, metadata update data, latest version). Provenance information coverage varies across the modeled surveyed. However, its great importance lead to the development of various special vocabularies like the Open Provenance Model[20] and PROV-O [94]. DataID [32] is an effort to provide semantically rich metadata with focus on providing detailed provenance, license and access information.

- **Geospatial information**: Information reflecting the geographical coverage of the dataset represented with coordinates or geometry polygons. There are several additional models and extensions specifically designed to express geographical information. The Infrastructure for Spatial Information in the European Community (INSPIRE) directive[21] aims at establishing an infrastructure

[16] http://dublincore.org/documents/dcmi-terms/
[17] http://oeg-dev.dia.fi.upm.es/licensius/static/ldr/
[18] http://www.w3.org/ns/odrl/2/
[19] http://xmlns.com/foaf/spec/
[20] http://open-biomed.sourceforge.net/opmv/
[21] http://inspire.ec.europa.eu/

for spatial information. Mappings have been made between DCAT-AP and the INSPIRE metadata. CKAN provides as well a spatial extension[22] to add geospatial capabilities. It allows importing geospatial metadata from other resources and supports various standards (e.g. ISO 19139) and formats (e.g. GeoJSON).

- **Temporal information**: Information reflecting the temporal coverage of the dataset (e.g. from date to date). There has been some notable work on extending CKAN to include temporal information. `govdata.de` is an Open Data portal in Germany that extends the CKAN data model to include information like `temporal_granularity`, `temporal_coverage_to` and `temporal_granularity_from`.

- **Statistical information**: Statistical information about the data types and patterns in datasets (e.g. properties distribution, number of entities and RDF triples). This information is particularly useful to explore a dataset as it gives detailed insights about the raw data when provided properly. VoID is the only model that provides statistical information about a dataset. VoID defines properties to express different statistical characteristics of datasets like the total number of triples, total number of entities, total number of distinct classes, etc. However, there are other vocabularies such as SCOVO [68] that can model and publish statistical data about datasets.

- **Quality information**: Information that indicates the quality of the dataset on the metadata and instance levels. In addition to that, a dataset should include an openness score that measures its alignment with the Linked Data publishing standards [17]. Quality information is only expressed in the POD metadata. However, `govdata.de` extends the CKAN model also to include a `ratings_average` field. Moreover, there are various other vocabularies like daQ [42] that can be used to express datasets quality. The RDF Review Vocabulary[23] can also be used to express reviews and ratings about the dataset or its resources.

Figure 3.2 summarizes the information grouping. Each dataset describes one or more information section (resources, tags, groups or organizations) which can contain one more information type.

[22] https://github.com/ckan/ckanext-spatial
[23] http://vocab.org/review/

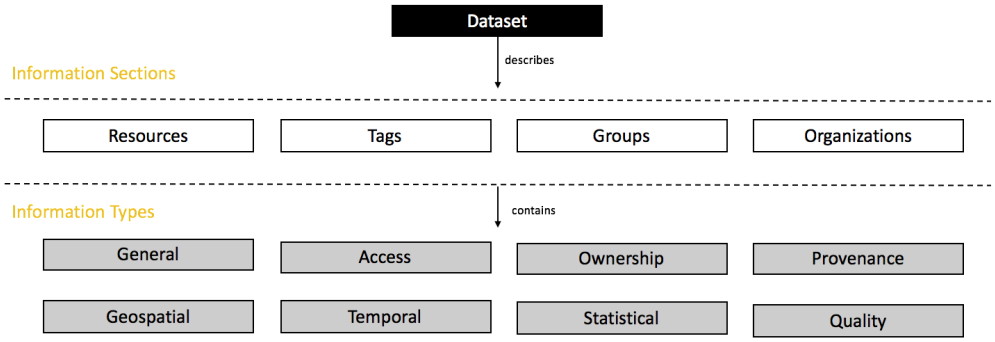

Figure 3.2: Information sections and groups across data models

3.3 Towards A Harmonized Model

Since establishing a common vocabulary or model is the key to communication, we identified the need for an harmonized dataset metadata model containing sufficient information so that consumers can easily understand and process datasets. To create the mappings between the different models, we performed various steps:

- Examine the model or vocabulary specification and documentation.
- Examine existing datasets using these models and vocabularies. Data Portals[24] provides a comprehensive list of Open Data Portals from around the world. It was our entry point to find out portals using CKAN or DKAN as their underlying DMS. We also investigated portals known to be using specific DMS. Socrata, for example, maintains a list of Open Data portals using their software on their homepage such as http://pencolorado.org and http://data.maryland.gov.
- Examine the source code of some portals. This was specifically the case for Socrata as their API returns the raw data serialized as JSON rather than the dataset's metadata. As a consequence, we had to investigate the Socrata Open Data API (SODA) source code[25] and check the different classes and interfaces.

CKAN	DKAN	POD	DCAT	VoID	Schema.org	Socrata
resources	resources	distribution	dcat:Distribution	void:Dataset→ void:dataDump	Dataset:distribution	attachments
tags	tags	keyword	dcat:Dataset→ :keyword	void:Dataset→ :keyword	CreativeWork:keywords	tags
groups	groups	theme	dcat:Dataset→ :theme	-	CreativeWork:about	category
organization	organization	publisher	dcat:Dataset→ :publisher	void:Dataset→ :publisher	-	-

[24] http://dataportals.org

[25] https://github.com/socrata/soda-java/tree/master/src/main/java/com/socrata/model

The first task is to map the four main information sections (resources, tags, groups and organization) across those models. Table 3.3 shows our proposed mappings. For the ontologies (DCAT, VoID), the first part represents the class and the part after → represents the property. For Schema.org, the first part refers to the schema and the second part after : refers to the property.

Table 3.1 presents the full mappings between the models across the information groups. Entries in the CKAN marked with ∗ are properties from CKAN extensions and not included in the original data model. Similar to the sections mappings, for the ontologies (DCAT, VoID), the first part represents the class and the part after → represents the property. However, sometimes the part after → refers to another resource. For example, to describe the dataset's maintainer email in DCAT, the information should be presented in the `dcat:Dataset` class using the `dcat:contactPoint` property. However, the range of this property is a resource of type `vcard` which has the property `hasEmail`.

For Schema.org, similar to the sections mapping, the first part refers to the schema and the second part after : refers to the property. However, if the property is inherited from another schema we denote that by using a → as well. For example, the size of a dataset is a property for a `Dataset` schema specified in its `distribution` property. However, the type of `distribution` is `dataDownload` which is inherited from the `MediaObject` schema. The size for `MediaObject` is defined in its `contentSize` property which makes the mapping string `Dataset:distribution`→ `DataDownload`→ `MediaObject:contentSize`.

The CKAN model controls the values to be used in describing some dataset properties. For example, the `resource_type` property can have the values: file: direct accessible bitstream, file.upload: file uploaded to the CKAN FileStore[26], api, visualization, code: the actual source code or a reference to a code repository and documentation. To know the set of values in these fields we examined the models of several datasets with a tool we created called Roomba, see chapter 4.

We created two main reports with Roomba. The first aims to collect the list of `extras` values using the query string `extras>value:extras>key` (see listing 3.1) and the second one is to list the file types specified for resources using the query string `resources>resource_type:resources>name` (see listing 3.2). We ran the report generation process on two prominent data portals: the Linked Open Data (LOD) cloud hosted on the Datahub containing 259 datasets and the Africa's largest open data portal, OpenAfrica[27] that contains 1653 datasets.

[26] http://docs.ckan.org/en/ckan-1.8/filestore.html
[27] http://africaopendata.org/

```
namespace with total count of: 1169
triples with total count of: 1193
publishingInstitution with total count of: 17
shortname with total count of: 753
links:dbpedia with total count of: 768
links:lcsh with total count of: 42
```

Listing 3.1: Excerpt of the report for aggregating *extras* field values on the LOD Cloud

```
file with total count of: 157
api with total count of: 91
metadata with total count of: 13
example with total count of: 26
file.upload with total count of: 8
documentation with total count of: 8
api, api/sparq, rdf with total count of: 5
Publication with total count of: 1
Dataset with total count of: 1
```

Listing 3.2: Result for aggregating *resource_type* field values on the LOD Cloud

After examining the results, we noticed that for OpenAfrica, 53% of the datasets contained additional information about the geographical coverage of the dataset (e.g. spatial-reference-system, spatial_harvester, bbox-east-long, bbox-north-long, bbox-south-long, bbox-west-long). In addition, 16% of the datasets have additional provenance and ownership information (e.g frequency-of-update, dataset-reference-date). For the LOD cloud, the main information embedded in the extras fields are about the structure and statistical distribution of the dataset (e.g. namespace, number of triples and links). The OpenAfrica resources did not specify any extra resource types. However, in the LOD cloud, we observe that multiple resources define additional types (e.g. example, api/sparql, publication, example).

Examining the different models, we noticed a lack of a complete model that covers all the information types. There is an abundance of extensions and application profiles that try to fill in those gaps, but they are usually domain specific addressing specific issues like geographic or temporal information. To the best of our knowledge, there is still no complete model that encompasses all the described information types. HDL aims at filling this gap by taking the best from these models (see appendix A).

Table 3.1: Harmonized Dataset Models Mappings

Data Model	CKAN	DKAN	POD	DCAT	VoID	Schema.org	Socrata
General Information	id	id	identifier	dcat:Dataset→ dct:identifier			id/externalId
	private	private	accessLevel				privateMetadata
	state	state					publicationStage
	type	type				Thing:additionalType	
	name	name				Thing:name	name
	isopen						
	notes	notes	description	dcat:Dataset→ dct:description	void:Dataset→ dct:description	Thing:description	description
	title	title	title	dcat:Dataset→ dct:title	void:Dataset→ dc:title	Thing:name	name
	num_resources				void:Dataset→ void:documents		
	num_tags						
			conformsTo	dcat:Dataset→ dct:conformsTo	void:Dataset→ dct:conformsTo		
			language	dcat:Dataset→ dct:language	void:Dataset→ dct:language	CreativeWork:inLanguage	
			accuralPeriodicity	dcat:Dataset→ dct:accuralPeriodicity	void:Dataset→ dct:accuralPeriodicity		
access information	license_title	license_title	license	dcat:Distribution→ dct:license	void:Dataset→ dct:license		license→ name
	license_id						licenseId
	license_url					CreativeWork:license	license → termsLink
	url	url	landingPage	dcat:Dataset→ dcat:landingPage		Thing:url	
			rights	dcat:Distribution→ dct:rights	void:Dataset→ dct:rights		
	attribution_text∗						attribution
							attributionLink
provenance	version					CreativeWork:version	
	revision_id						
	metadata_created	metadata_created		dcat:Distribution→ dct:created	void:Dataset→ dct:created	CreativeWork:dateCreated	
	metadata_modified	metadata_modified	modified	dcat:Distribution→ dct:modified	void:Dataset→ dct:modified	CreativeWork:dateModified	
	revision_timestamp	revision_timestamp					
			issued	dcat:Distribution→ dct:issued	void:Dataset→ dct:issued	CreativeWork:datePublished	
			temporal	dcat:Dataset→ dct:temporal	void:Dataset→ dct:temporal	Dataset:temporal	
ownership	maintainer	maintainer	contactPoint→ fn	dcat:Dataset→ dcat:contactPoint→ vcard:fn		CreativeWork:producer→ Thing:name	owner→ displayName / owner→ ScreenName
	maintainer_email	maintainer_email	contactPoint→ hasEmail	dcat:Dataset→ dcat:contactPoint→ vcard:hasEmail		CreativeWork:producer→ Person:email	
	owner_org					CreativeWork:sourceOrganization:LegalName	
	author			dcat:Dataset→ dct:creator→ foaf:Person:givenName	void:Dataset→ dct:creator→ foaf:Person:givenName	CreativeWork:author→ Thing:name	
	author_email	author_email		dcat:Dataset→ dct:creator→ foaf:Person:mbox	void:Dataset→ dct:creator→ foaf:Person:mbox	CreativeWork:author→ Person:email	
			bureauCode				
			programCode				
	description					CreativeWork:sourceOrganization→ Thing:description	
			isPartOf			CreativeWork:isPartOf	
						CreativeWork:hasPart	
			systemOfRecords				
			describedBy				
			describedByType				
GeoSpatial	spatial-text∗		spatial	dcat:Dataset→ dct:spatial	void:Dataset→ dct:spatial	Dataset:spatial	

Table 3.1 Harmonized Dataset Models Mappings

Data Model	CKAN	DKAN	POD	DCAT	VoID	Schema.org	Socrata
	geographical_granularity∗						
							bbox
							layers
							bboxCrs
							namespace
Temporal			temporal	dcat:Dataset→ dct:temporal	void:Dataset→ dct:temporal	Dataset:temporal	
	temporal_granularity∗						
	temporal_coverage_to∗						
	temporal_coverage_from∗						
Quality	ratings_average∗		dataQuality			CreativeWork:aggregateRating	
Organization							
General Information	title		name	dcat:Dataset→ dct:creator→ foaf:Organization:givenName	void:Dataset→ dct:creator→ foaf:Organization:givenName	CreativeWork:sourceOrganization:LegalName	
	description					CreativeWork:sourceOrganization→ Thing:description	
	id						
	type					CreativeWork:sourceOrganization→ Thing:additionalType	
	name					CreativeWork:sourceOrganization→ Thing:name	
	image_url						
	state						
	is_organization						
	approval_status						
provenance	revision_timestamp		subOrganizationOf			CreativeWork:sourceOrganization:subOrganization	
	revision_id						
Resources							
general	resource_group_id	resource_group_id					
	id	id					blobId
	size	size		dcat:Distribution→ dcat:byteSize		Dataset:distribution→ DataDownload→ MediaObject:contentSize	
	state	state					
	hash						
	description	description	description	dcat:Distribution→ dct:description		Dataset:distribution→ DataDownload→ Thing:description	
	format	format	format	dcat:Distribution→ dct:format	void:Dataset→ dct:format	Dataset:distribution→ DataDownload→ MediaObject:encodingFormat	
	mimetype	mimetype	mediaType	dcat:Distribution→ dcat:mediaType			
	mimetype_inner						
	name	name	title	dcat:Distribution→ dct:title		Dataset:distribution→ DataDownload→ Thing:name	filename / name
	position						
	resource_type					Dataset:distribution→ DataDownload→Thing:additionalType	
			describedBy				
			describedByType				
			conformsTo				
access information	cache_url						
	url-type						
	url	url	downloadURL	dcat:Distribution→ dcat:downloadURL	void:Dataset→ void:dataDump	Dataset:distribution→ DataDownload→ Thing:url	

Table 3.1 Harmonized Dataset Models Mappings

Data Model	CKAN	DKAN	POD	DCAT	VoID	Schema.org	Socrata
			accessURL	dcat:Distribution→ dcat:accessURL		Dataset:distribution→ DataDownload→ MediaObject:contentUrl	accessPoints
	webstore_url						
provenance	cache_last_updated						
	revision_timestamp	revision_timestamp					
	webstore_last_updated						
	created	created				Dataset:distribution→ DataDownload→ CreativeWork:dataCreated	created_at
	last_modified	last_modified				Dataset:distribution→ DataDownload→ CreativeWork:dataModified	updated_at
	revision_id	revision_id					
Groups							
General	display_name	display_name					
	description	description					
	title	title					
	image_display_url	image_display_url					
	id	id					
	name	name					
	subgroups∗						
Tags							
General	vocabulary_id	vocabulary_id		dcat:Dataset→ dcat:theme→ skos:ConceptScheme			
	display_name			dcat:Dataset→ dcat:keyword			
	name	name		dcat:Dataset→ dcat:theme→ skos:Concept			
	state						
	id	id					
Provenance	revision_timestamp						

3.4 Summary

In this chapter, we surveyed the landscape of various models and vocabularies that described datasets on the web. Since establishing a common vocabulary or model is the key to communication, we identified the need for a harmonized dataset metadata model containing sufficient information so that consumers can easily understand and process datasets. We have identified four main sections that should be included in the model: resources, groups, tags and organizations. Furthermore, we have classified the information to be included into eight types. Our main contribution is a set of mappings between each properties of those models. This has lead to the design of HDL, an 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.

Chapter 4

Dataset Profiles Generation and Validation

4.1 Introduction

Data is being published by both the public and private sectors and covers a diverse set of domains from life sciences to media or government data. The Linked Open Data cloud is potentially a gold mine for organizations and individuals who are trying to leverage external data sources in order to produce more informed business decisions [29]. This success lies in the cooperation between data publishers and consumers. Consumers are empowered to find, share and combine information in their applications easily. However, the heterogeneous nature of data sources reflects directly on the data quality as these sources often contain inconsistent as well as misinterpreted and incomplete metadata information. Considering the significant variation in size, the languages used and the freshness of the data, one realizes that finding useful datasets without prior knowledge is increasingly complicated. This can be clearly noticed in the LOD Cloud where few datasets such as DBPedia [24], Freebase [28] and YAGO [135] are favored over less popular datasets that may include domain specific knowledge more suitable for the tasks at hand. For example, for the task of building context-aware recommender systems in an academic digital library over the LOD cloud, popular datasets like the Semantic Web Dog Food[1], DBLP[2] or Yovisto[3] can be favored over lesser known but more specific datasets like VIAF[4] which links authority files of 20 national libraries, list of subject headings for public libraries in Spain[5] or the French dissertation search engine[6].

Dataset discovery can be done through public data portals like Datahub[7] and

[1] http://datahub.io/dataset/semantic-web-dog-food
[2] http://datahub.io/dataset/dblp
[3] http://datahub.io/dataset/yovisto
[4] http://datahub.io/dataset/viaf
[5] http://datahub.io/dataset/lista-encabezamientos-materia
[6] http://datahub.io/dataset/thesesfr
[7] http://datahub.io

Europe's Public Data[8] or private ones like Quandl[9] and Engima[10]. Private portals harness manually curated data from various sources and expose them to users either freely or through paid plans. The data available is of higher quality but lesser quantity compared to what is available in public portals. Similarly, in some public data portals, administrators manually review datasets information, validate, correct and attach suitable metadata information. This information is mainly in the form of predefined tags such as *media, geography, life sciences* for organization and clustering purposes. However, the diversity of those datasets makes it harder to classify them in a fixed number of predefined tags that can be subjectively assigned without capturing the essence and breadth of the dataset [90]. Furthermore, the increasing number of datasets available makes the metadata review and curation process unsustainable even when outsourced to communities.

In this chapter, we address the challenges of automatic validation and generation of descriptive datasets profiles. We describe Roomba, an extensible framework consisting of a processing pipeline that combines techniques for data portals identification, datasets crawling and a set of pluggable modules combining several profiling tasks. The framework validates the provided dataset metadata against an aggregated standard set of information. Metadata fields are automatically corrected when possible (e.g. adding a missing license URL reference). Moreover, a report describing all the issues highlighting those that cannot be automatically fixed is created to be sent by email to the dataset's maintainer. There exist various statistical and topical profiling tools for both relational and Linked Data. The architecture of the framework allows to easily add them as additional profiling tasks. However, in this chapter, we focus on the task of dataset metadata profiling. We validate our framework against a manually created set of profiles and manually check its accuracy by examining the results of running it on various CKAN-based data portals.

4.2 Motivation

Metadata provisioning is one of the Linked Data publishing best practices mentioned in [21]. Datasets should contain the metadata needed to effectively understand and use them. This information includes the dataset's license, provenance, context, structure and accessibility. The ability to automatically check this metadata helps in:

- **Delaying data entropy**: *Information entropy* refers to the degradation or loss limiting the information content in raw or metadata. As a consequence

[8] http://publicdata.eu
[9] https://quandl.com/
[10] http://enigma.io/

of information entropy, data complexity and dynamicity, the life span of data can be very short. Even when the raw data is properly maintained, it is often rendered useless when the attached metadata is missing, incomplete or unavailable. Comprehensive high quality metadata can counteract these factors and increase dataset longevity [88].

- **Enhancing data discovery, exploration and reuse**: Users who are unfamiliar with a dataset require detailed metadata to interpret and analyze accurately unfamiliar datasets. A study conducted by the European Union commission [146] found that both business and users are facing difficulties in discovering, exploring and reusing public data. due to missing or inconsistent metadata information.

- **Enhancing spam detection**: Portals hosting public open data like Datahub allow anyone to freely publish datasets. Even with security measures like captchas and anti-spam devices, detecting spam is increasingly difficult. In addition to that, the increasing number of datasets hinders the scalability of this process, affecting the correct and efficient spotting of datasets spam.

4.3 Related Work

Data Catalog Vocabulary (DCAT) [102] and the Vocabulary of Interlinked Datasets (VoID) [38] are concerned with metadata about RDF datasets. There exist several tools aiming at exposing dataset metadata using these vocabularies. In [27], the authors generate VoID descriptions limited to a subset of properties that can be automatically deduced from resources within the dataset. However, it still provides data consumers with interesting insights. Flemming's Data Quality Assessment Tool[11] provides basic metadata assessment as it computes data quality scores based on manual user input. The user assigns weights to the predefined quality metrics and answer a series of questions regarding the dataset. These include, for example, the use of obsolete classes and properties by defining 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 ODI certificate[12], on the other hand, provides a description of the published data quality in plain English. It aspires to act as a mark of approval that helps publishers understand how to publish good open data and users how to use it. It gives publishers the ability to provide assurance and support on their data while encouraging further improvements through an ascending

[11] http://linkeddata.informatik.hu-berlin.de/LDSrcAss/datenquelle.php
[12] https://certificates.theodi.org/

scale. ODI comes as an online and free questionnaire for data publishers focusing on certain characteristics about their data. Although these approaches try to perform metadata profiling, they are either incomplete or manual. In our framework, we propose a more automatized and complete approach.

Metadata profiling: The Project Open Data Dashboard[13] tracks and measures how US government web sites 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 Datahub LOD Validator[14] gives an overview of Linked Data sources cataloged on the Datahub. 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 entire LOD cloud group and it is very specific to the LOD cloud group rules and regulations.

Statistical profiling: Calculating statistical information on datasets is vital to applications dealing with query optimization and answering, data cleansing, schema induction and data mining [78, 55, 90]. Semantic sitemaps [37] and RDFStats [91] are one of the first to deal with RDF data statistics and summaries. ExpLOD [82] creates statistics on the interlinking between datasets based on `owl:sameAs` links. In [99], the author introduces a tool that induces the actual schema of the data and gather corresponding statistics accordingly. LODStats [13] is a stream-based approach that calculates more general dataset statistics. ProLOD++ [1] is a Web-based tool that allows LOD analysis via automatically computed hierarchical clustering [25]. Aether [104] generates VoID statistical descriptions of RDF datasets. It also provides a Web interface to view and compare VoID descriptions. LODOP [52] is a MapReduce framework to compute, optimize and benchmark dataset profiles. The main target for this framework is to optimize the runtime costs for Linked Data profiling. In [79] authors calculate certain statistical information for the purpose of observing the dynamic changes in datasets.

Topical Profiling: Topical and categorical information facilitates dataset search and reuse. Topical profiling focuses on content-wise analysis at the instances and ontological levels. GERBIL [143] is a general entity annotation framework that pro-

[13] http://labs.data.gov/dashboard/
[14] http://validator.lod-cloud.net/

vides machine processable output allowing efficient querying. In addition, there exist several entity annotation tools and frameworks [36] but none of those systems are designed specifically for dataset annotation. In [56], the authors created a semantic portal to manually annotate and publish metadata about both LOD and non-RDF datasets. In [90], the authors automatically assigned Freebase domains to extracted instance labels of some of the LOD Cloud datasets. The goal was to provide automatic domain identification, thus enabling improving datasets clustering and categorization. In [26], the authors extracted dataset topics by exploiting the graph structure and ontological information, thus removing the dependency on textual labels. In [48], the authors generate VoID and VoL descriptions via a processing pipeline that extracts dataset topic models ranked on graphical models of selected DBpedia categories.

Dataset Search: Dataset search can be done without relying on attached metadata (tags and categories). For example, there exist several approaches to create LOD indexes. In [3], the authors used VoID descriptions to optimize query processing by determining relevant query-able datasets. In [63], the authors created an approximate index structure (QTree) and an algorithm for answering conjunctive queries over Linked Data. SchemEX [85] is a stream-based approach leveraging type and property information of RDF instances to create schema-level indexes.

Semantic search engines like Sindice [44], Swoogle [45] and Watson [40] help in entities lookup but they are not designed specifically for dataset search. In [113], the authors utilized the sig.ma index [141] to identify appropriate data sources for interlinking. Dataset search and discovery is currently done via data portals that rely on attached metadata to provide dataset search features as they run a Solr index on the metadata schemas. Having missing or inconsistent information will affect the search results quality.

Although the above mentioned tools are able to provide various types of information about a dataset, there exists no approach that aggregates this information and is extensible to combine additional profiling tasks. To the best of our knowledge, this is the first effort towards extensible automatic validation and generation of descriptive dataset profiles.

4.4 Profiling Data Portals

In this section, we provide an overview of Roomba's architecture and the processing steps for validating and generating dataset profiles. Figure 4.1 shows the main steps which are the following: (i) data portal identification; (ii) metadata extraction; (iii) instance and resource extraction; (iv) profile validation (v) profile and report generation.

Roomba is built as a Command Line Interface (CLI) application using Node.js allowing users to:

- Fetch all the information about datasets from a data portal
- Fetch all the groups information from a data portal
- Crawl datasets (a specific dataset, datasets in a specific group, datasets in the whole portal)
- Execute aggregation report on a specific group or on the whole data portal
- Profile a specific dataset, a whole group or the whole data portal

Section B.1 detail the instructions for installing and running the framework. The various steps are explained in detail below.

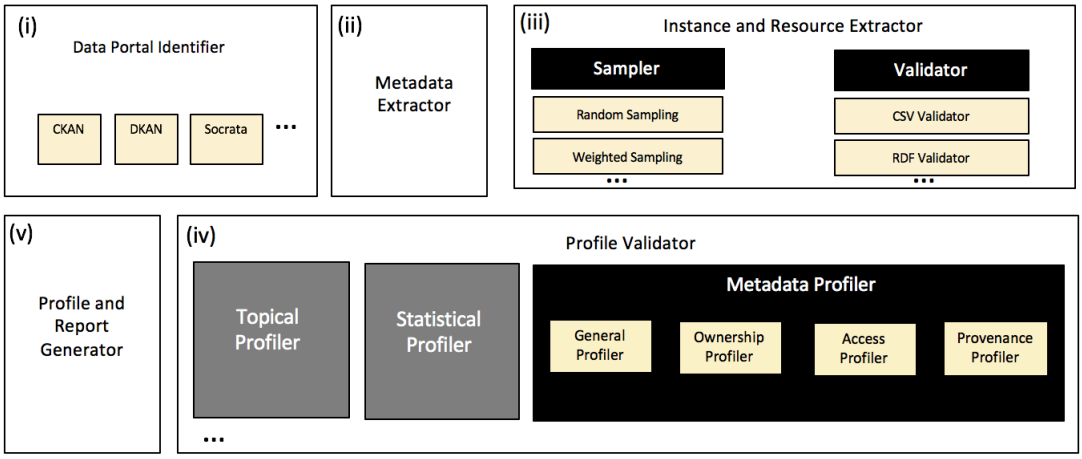

Figure 4.1: Processing pipeline for validating and generating dataset profiles

4.4.1 Data Portal Identification

Data portals can be considered as data access points providing tools to facilitate data publishing, sharing, searching and visualization. Section 3.1 highlights the main data portal softwares and models. In addition to these tradition data portals, there is a set of tools that allow exposing data directly as RESTful APIs like Datatank[15] and Database-to-API[16].

Roomba should be extensible to any data portal. Since every portal has its own API and data model, identifying the software powering data portals is a vital first

[15] http://thedatatank.com
[16] https://github.com/project-open-data/db-to-api

step. We rely on several Web scraping techniques in the identification process which includes a combination of the following:

- **URL inspection**: Various CKAN based portals are hosted on subdomains of the http://ckan.net. For example, CKAN Brazil (http://br.ckan.net). Checking the existence of certain URL patterns can detect such cases.

- **Meta tags inspection**: The <meta> tag provides metadata about the HTML document. They are used to specify page description, keywords, author, etc. Inspecting the content attribute can indicate the type of the data portal. We use CSS selectors to check the existence of these meta tags. An example of a query selector is meta[content*=''ckan''] (all meta tags with the attribute content containing the string *CKAN*). This selector can identify CKAN portals whereas the meta[content*=''Drupal''] can identify DKAN portals.

- **Document Object Model (DOM) inspection**: Similar to the meta tags inspection, we check the existence of certain DOM elements or properties. For example, CKAN powered portals will have DOM elements with class names like ckan-icon or ckan-footer-logo. A CSS selector like .ckan-icon will be able to check if a DOM element with the class name ckan-icon exists. The list of elements and properties to inspect is stored in a separate configurable object for each portal. This allows the addition and removal of elements as deemed necessary.

The identification process for each portal can be easily customized by overriding the default function. Moreover, adding or removing steps from the identification process can be easily configured.

After those preliminary checks, we query one of the portal's API endpoints. For example, DataHub is identified as CKAN, so we will query the API endpoint on http://datahub.io/api/action/package_list. A successful request will list the names of the site's datasets, whereas a failing request will signal a possible failure of the identification process.

4.4.2 Metadata Extraction

Data portals expose a set of information about each dataset as metadata. The model used varies across portals. However, a standard model (see section 3.2) should contain information about the dataset's title, description, maintainer email, update and creation date, etc.

Since Roomba operates on CKAN-based data portals, we validate the extracted metadata against the CKAN standard model[17] (see listing 4.1).

```
{
 "license_title": "License not specified",
 "maintainer": "",
 "relationships_as_object": [],
 "private": false,
 "maintainer_email": "",
 "num_tags": 4,
 "id": "7e4d4ef3-f452-4c35-963d-9c6e582374b3",
 "metadata_created": "2015-07-22T14:29:55.490069",
 "metadata_modified": "2015-07-22T14:30:18.584924",
 "author": "Lucy Chambers",
 "author_email": "",
 "state": "active",
 "version": "",
 "creator_user_id": "01b3756a-e1ca-4d4a-b8f1-6880a00095d6",
 "type": "dataset"
}
```

Listing 4.1: Excerpt of a dataset profile in CKAN standard model

After identifying the underlying portal software, we perform iterative queries to the API in order to fetch datasets metadata and persist them in a file-based cache system. Depending on the portal software, we can issue specific extraction jobs. For example, in CKAN-based portals, we are able to crawl and extract the metadata of a specific dataset, all the datasets in a specific group (e.g. LOD cloud) or all the datasets in the portal.

4.4.3 Instance and Resource Extraction

From the extracted metadata we are able to identify all the resources associated with that dataset. They can have various types like a SPARQL endpoint, API, file, visualization, etc. However, before extracting the resource instance(s) we perform the following steps:

- **Resource metadata validation and enrichment**: Check the resource attached metadata values. Similar to the dataset metadata, each resource should include information about its mimetype, name, description, format, valid dereferenceable URL, size, type and provenance. The validation process issue an

[17] http://demo.ckan.org/api/3/action/package_show?id=adur_district_spending

HTTP request to the resource and automatically fills up various missing information when possible, like the mimetype and size by extracting them from the HTTP response header. However, missing fields like name and description that needs manual input are marked as missing and will appear in the generated summary report.

- **Format validation**: Validate specific resource formats against a linter or a validator. For example, node-csv[18] for CSV files and n3[19] to validate N3 and Turtle RDF serializations.

Considering that certain datasets contain large amounts of resources and the limited computation power of some machines on which the framework might run on, a sampler module can be introduced to execute various sample-based strategies detailed as they were found to generate accurate results even with comparably small sample size of 10%. These strategies introduced in [48] are:

- **Random Sampling**: Randomly selects resources instances.
- **Weighted Sampling**: Weighs each resources as the ratio of the number of datatype properties used to define a resource over the maximum number of datatype properties over all the datasets resources.
- **Resource Centrality Sampling**: Weighs each resource as the ratio of the number of resource types used to describe a particular resource divided by the total number of resource types in the dataset. This is specific and important to RDF datasets where important concepts tend to be more structured and linked to other concepts.

However, the sampler is not restricted only to these strategies. Strategies like those introduced in [98] can be configured and plugged in the processing pipeline.

4.4.4 Profile Validation

A dataset profile should include descriptive information about the data examined. In Roomba, we have identified three main categories of profiling information. However, the extensibility of our framework allows for additional profiling techniques to be plugged in easily (section 5.5 describes an extension to measure the objective qualities of datasets).

Metadata validation process identifies missing information and the ability to automatically correct them. Each set of metadata (general, access, ownership and

[18] https://github.com/wdavidw/node-csv
[19] https://github.com/RubenVerborgh/N3.js

provenance) is validated and corrected automatically when possible. Each profiler task has a set of metadata fields to check against. The validation process check if each field is defined and if the value assigned is valid.

There exist many special validation steps for various fields. For example, the email addresses and urls should be validated to ensure that the value entered is syntactically correct. In addition to that, for urls, we issue an HTTP `HEAD` request in order to check if that URL is reachable. We also use the information contained in a valid `content-header` response to extract, compare and correct some resources metadata values like `mimetype` and `size`.

Despite the legal issues surrounding Linked Data licenses [77], it is still considered a gold mine for organizations who are trying to leverage external data sources in order to produce more informed business decisions [29]. In [105], the authors see the potential economic effect unfolding in education, transportation, consumer products, electricity, oil and gas, health care and consumer finance. They estimate the potential annual value enabled by Open Data in these domains to be 3 trillion US Dollars across seven domains. As a result, validating license related information is vital. However, from our experiments, we found out that datasets' license information is noisy. The license names if found are not standardized. For example, `Creative Commons CCZero` can be also `CC0` or `CCZero`. Moreover,the license URI if found and if de-referenceable can point to different reference knowledge bases e.g., `http://opendefinition.org`. To overcome this issue, we have manually created a mapping file standardizing the set of possible license names and the reference knowledge base (see listing F.1). In addition, we have also used the open source and knowledge license information[20] to normalize the license information and add extra metadata like the domain, maintainer and open data conformance.

```
{
        "license_id" : ["ODC-PDDL-1.0"],
        "disambiguations" : ["Open Data Commons Public Domain
           Dedication and License (PDDL)"]
},
{
        "license_id" : ["CC-BY-SA-4.0", "CC-BY-SA-3.0"],
        "disambiguations" : ["cc-by-sa", "CC BY-SA","Creative
           Commons Attribution Share-Alike"]
}
```

Listing 4.2: License mapping file sample

[20] https://github.com/okfn/licenses

4.4.5 Profile and Report Generation

The validation process highlights the missing information and presents them in a human readable report (see appendix C). The report can be automatically sent to the dataset maintainer email if exists in the metadata. In addition to the generated report, the enhanced profiles are represented in JSON using the CKAN data model and are publicly available[21].

```
================================================================================
                                Metadata Report
================================================================================
group information is missing. Check organization information as they can be
    mixed sometimes
organization_image_url field exists but there is no value defined
================================================================================
                                Tag Statistics
================================================================================
There is a total of: 21 [undefined] vocabulary_id fields  100.00%
================================================================================
                                 License Report
================================================================================
License information has been normalized !
================================================================================
                                Resource Statistics
================================================================================
There is a total of: 10 [missing] url-type fields  100.00%
There is a total of: 9 [missing] created fields  90.00%
There is a total of: 10 [undefined] cache_last_updated fields  100.00%
There is a total of: 10 [undefined] size fields  100.00%
There is a total of: 10 [undefined] hash fields  100.00%
There is a total of: 10 [undefined] mimetype_inner fields  100.00%
There is a total of: 7 [undefined] mimetype fields  70.00%
There is a total of: 10 [undefined] cache_url fields  100.00%
There is a total of: 6 [undefined] name fields  60.00%
There is a total of: 9 [undefined] webstore_url fields  90.00%
There is a total of: 9 [undefined] last_modified fields  90.00%
There is one [undefined] format field  10.00%
================================================================================
                            Resource Connectivity Issues
================================================================================
There are 2 connectivity issues with the following URLs:
  - \url{http://dbpedia.org/void/Dataset}
================================================================================
                            Un-Reachable URLs Types
================================================================================
There are: 1 unreachable URLs of type [file]
```

Listing 4.3: Excerpt of the DBpedia validation report

[21] https://github.com/ahmadassaf/opendata-checker/tree/master/results

Data portal administrators need an overall knowledge of the portal datasets and their properties. Our framework has the ability to generate numerous reports of all the datasets by passing formatted queries. There are two main sets of aggregation tasks that can be run:

- **Aggregating meta-field values**: Passing a string that corresponds to a valid field in the metadata. The field can be flat like `license_title` (aggregates all the license titles used in the portal or in a specific group) or nested like `resource>resource_type` (aggregates all the resources types for all the datasets). Such reports are important to have an overview of the possible values used for each metadata field.

- **Aggregating key:object meta-field values**: Passing two meta-field values separated by a colon `:` e.g., `resources>resource_type:resources>name`. These reports are important as you can aggregate the information needed when also having the set of values associated to it printed.

For example, the meta-field value query `resource>resource_type` run against the LODCloud group will result in an array containing $[file, api, documentation...]$ values. These are all the resource types used to describe all the datasets of the group. However, to be able to know also what are the datasets containing resources corresponding to each type, we issue a key:object meta-field query `resource>resource_type:name`. The result will be a JSON object having the `resource_type` as the key and an array of corresponding datasets titles that has a resource of that type.

4.5 Experiments and Evaluation

In this section, we provide the experiments and evaluation of Roomba. All the experiments are reproducible by our tool and their results are available in its Github repository. A CKAN dataset metadata describes four main sections in addition to the core dataset's properties. These sections are:

- **Resources**: The distributable parts containing the actual raw data. They can come in various formats (JSON, XML, RDF, etc.) and can be downloaded or accessed directly (REST API, SPARQL endpoint).

- **Tags**: Provide descriptive knowledge on the dataset content and structure. They are used mainly to facilitate search and reuse.

- **Groups**: A dataset can belong to one or more group that share common semantics. A group can be seen as a cluster or a curation of datasets based on shared categories or themes.

- **Organizations**: A dataset can belong to one or more organization controlled by a set of users. Organizations are different from groups as they are not constructed by shared semantics or properties, but solely on their association to a specific administration party.

Each of these sections contains a set of metadata corresponding to one or more type (general, access, ownership and provenance). For example, a dataset resource will have general information such as the resource name, access information such as the resource url and provenance information such as creation date. The framework generates a report aggregating all the problems in all these sections, fixing field values when possible. Errors can be the result of missing metadata fields, undefined field values or field value errors (e.g. unreachable URL or incorrect email addresses).

4.5.1 Experimental Setup

We ran our tool on two CAKN-based data portals. The first is the Datahub targeting specifically the LOD cloud group. The current state of the LOD cloud report [129] indicates that the LOD cloud contains 1014 datasets. They were harvested via an LDSpider crawler [75] seeded with 560 thousands URIs. Roomba on the other hand, fetches datasets hosted in data portals where datasets have attached relevant metadata. As a result, we relied on the information provided by the Datahub CKAN API. Examining the tags available, we found two candidate groups. The first tagged with "lodcloud" returned 259 datasets, while the second tagged with "lod" returned only 75 datasets. After manually examining the two lists, we found out the datasets grouped with the tag "lodcloud" are the correct ones. To qualify other CKAN-based portals for the experiments, we used `dataportals.org`, which contains a comprehensive list of Open Data portals from around the world. In the end, we chose the Amsterdam data portal [22]. The portal was commissioned in 2012 by the Amsterdam Economic Board Open Data Exchange (ODE), and covers a wide range of information domains (energy, economy, education, urban development, etc.) about Amsterdam metropolitan region.

We ran the Roomba instance and resource extractors in order to cache the metadata files for these datasets locally and ran the validation process. The experiments were executed on a 2.6 Ghz Intel Core i7 processor with 16GB of DDR3 memory

[22] http://data.amsterdamopendata.nl/

machine. The approximate execution time alongside the summary of the datasets' properties are presented in table 4.1.

Data Portal	No. Datasets	No. Groups	No. Resources	Processing Time
LOD Cloud	259	N/A	1068	140 mins
Amsterdam Open Data	172	18	480	35 mins

Table 4.1: Summary of the experiments details

In our evaluation, we focused on two aspects: i)*profiling correctness* which manually assesses the validity of the errors generated in the report, and ii)*profiling completeness* which assesses if the profilers cover all the errors in the datasets metadata.

4.5.2 Profiling Correctness

To measure profile correctness, we need to make sure that the issues reported by Roomba are valid on the dataset, group and portal levels.

On the dataset level, we choose three datasets from both the LOD Cloud and the Amsterdam data portal. The datasets details are shown in table 4.2.

Dataset Name	**Data Portal**	**Group ID**	**Resources**	**Tags**
dbpedia	Datahub	lodcloud	10	21
event-media	Datahub	lodcloud	9	15
bbc-music	Datahub	lodcloud	2	14
bevolking_cijfers_amsterdam	Amsterdam	bevolking	6	12
bevolking-prognoses-amsterdam	Amsterdam	bevolking	1	3
religieuze_samenkomstlocaties	Amsterdam	bevolking	1	8

Table 4.2: Datasets chosen for the correctness evaluation

To measure the profiling correctness on the groups level, we selected four groups from the Amsterdam data portal containing a total of 25 datasets. The choice was made to cover groups in various domains that contain a moderate number of datasets that can be checked manually (between 3-9 datasets). Table 4.3 summarizes the groups chosen for the evaluation.

Group Name	**Domain**	**Datasets**	**Resources**	**Tags**
bestuur-en-organisatie	Management	9	45	101
bevolking	Population	3	8	23
geografie	Geography	8	16	56
openbare-orde-veiligheid	Public Order & Safety	5	19	34

Table 4.3: Groups chosen for the correctness evaluation

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 indi-

vidual dataset level and on the aggregation level over groups. Since our portal level aggregation is extended from the group aggregation, we can infer that the portal level aggregation also produces complete correct profiles. However, the lack of a standard way to create and manage collections of datasets was the source of some errors when comparing the results from these two portals. For example, in Datahub, we noticed that all the datasets `groups` information were missing, while in the Amsterdam Open Data portal, all the `organisation` information was missing. Although the error detection is correct, the overlap in the usage of group and organization can give a false indication about the metadata quality.

4.5.3 Profiling Completeness

We analyzed the completeness of our framework by manually constructing a synthetic set of profiles. These profiles cover the range of uncommon problems that can occur in a certain dataset[23]. These errors are:

- Incorrect `mimetype` or `size` for resources;
- Invalid number of tags or resources defined;
- Check if the license information can be normalized via the `license_id` or the `license_title` as well as the normalization result;
- Syntactically invalid `author_email` or `maintainer_email`.

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 metadata problems that can be found in a CKAN standard model correctly.

4.6 Analyzing Profiling Results

In this section, we describe our experiments when running the Roomba tool on the LOD cloud. All the experiments are reproducible by our tool and their results are available on its Github repository.

Figures 4.2 and 4.3 show the percentage of errors found in metadata fields by section and by information type respectively. We observe that the most erroneous information for the dataset core information is related to ownership since this information is missing or undefined 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 contain missing or undefined values. Table 4.4 shows the top metadata fields errors for each metadata information type.

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

Metadata Field		Error %	Section	Error Type	Auto Fix
General	group	100%	Dataset	Missing	-
	vocabulary_id	100%	Tag	Undefined	-
	url-type	96.82%	Resource	Missing	-
	mimetype_inner	95.88%	Resource	Undefined	Yes
	hash	95.51%	Resource	Undefined	Yes
	size	81.55%	Resource	Undefined	Yes
Access	cahce_url	96.9%	Resource	Undefined	-
	webstore_url	91.29%	Resource	Undefined	-
	license_url	54.44%	Dataset	Missing	Yes
	url	30.89%	Resource	Unreachable	-
	license_title	16.6%	Dataset	Undefined	Yes
Provenance	cache_last_updated	96.91%	Resource	Undefined	Yes
	webstore_last_updated	95.88%	Resource	Undefined	Yes
	created	86.8%	Resource	Missing	Yes
	last_modified	79.87%	Resource	Undefined	Yes
	version	60.23%	Dataset	Undefined	-
Ownership	maintainer_email	55.21%	Dataset	Undefined	-
	maintainer	51.35%	Dataset	Undefined	-
	author_email	15.06%	Dataset	Undefined	-
	organization_image_url	10.81%	Dataset	Undefined	-
	author	2.32%	Dataset	Undefined	-

Table 4.4: Top metadata fields error % by type

We notice that 42.85% of the top metadata problems can be fixed automatically. Among them, 44.44% of these problems can be fixed by our tool while the others need tools that are plugged into the data portal. We further present and discuss the results grouped by metadata information type in the following sub-sections.

4.6.1 General information

34 datasets (13.13%) do not have valid `notes` values. `tags` information for the datasets are complete except for the `vocabulary_id` as this is missing from all the datasets' metadata. All the datasets `groups` information are missing `display_name`, `description`, `title`, `image_display_url`, `id`, `name`. After manual examination, we observe a clear overlap between group and organization information. Many datasets like `event-media` use the organization field to show group related information (being in the LOD Cloud) instead of the publishers details.

4.6.2 Access information

25% of the datasets access information (being the dataset URL and any URL defined in its groups) have issues: generally missing or unreachable URLs. 3 datasets (1.15%) do not have a URL defined (tip, uniprotdatabases, uniprotcitations) while 45 datasets (17.3%) defined 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 defined resources.

On the datasets resources level, we notice wrong or inconsistent values in the `size` and `mimetype` fields. However, 44 datasets have valid `size` field values and 54 have valid `mimetype` field values but they were not reachable, thus providing incorrect information. 15 fields (68%) of all the other access metadata are missing or have undefined values. Looking closely, we notice that most of these problems can be easily fixed automatically by tools that can be plugged to the data portal. For example, the top six missing fields are the `cache_last_updated`, `cache_url`, `urltype`, `webstore_last_updated`, `mimetype_inner` and `hash` which can be computed and filled 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 affecting highly the availability of these datasets. CKAN resources can be of various predefined types ($file$, $file.upload$, 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 files, 8 (2.39%) a re metadata and one (0.029%) are example and documentation types.

To have more details about the resources URL types, we created a $key : objectmeta-fieldvalues$ group level report on the LOD cloud with `resources> format:title`. This will aggregate the resources format information for each dataset. We observe that only 161 (62.16%) of the datasets valid URLs have SPARQL endpoints defined 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 defined `license_title` and `license_id` fields, where 141 (54.44%) have missing `license_url` field.

4.6.3 Ownership information

Ownership information is divided into direct ownership (author and maintainer) and organization information. Four fields (66.66%) of the direct ownership infor-

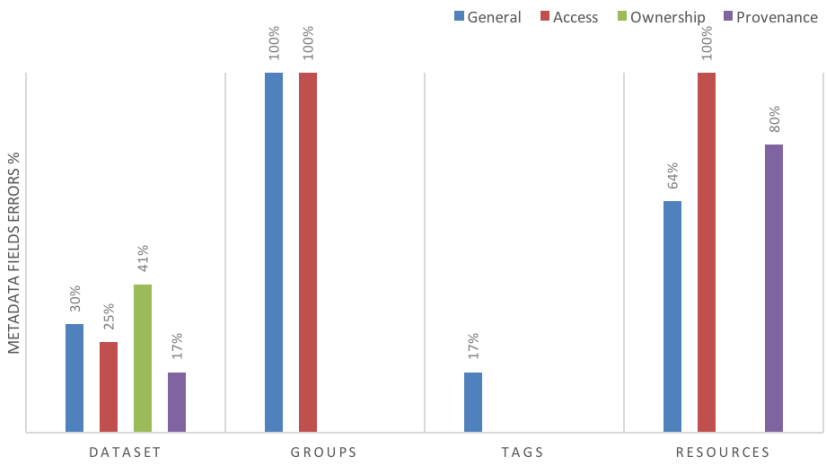

Figure 4.2: LOD Cloud error % by section

mation are missing or undefined. 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 (0.05%) and 6 (0.05%) of the defined `author_email` and `maintainer_email` fields are not valid email addresses respectively. For the organization information, two field values (16.6%) were missing or undefined. 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 undefined. 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 field requiring manual entry is the `version` field which was found to be missing in 60.23% of the datasets.

4.6.5 Enriched Profiles

Roomba can automatically fix, when possible, the license information (title, url and id) as well as the resources mimetype and size.

20 resources (1.87%) have incorrect `mimetype` defined, while 52 resources (4.82%) have incorrect `size` values. These values have been automatically fixed based on the values defined in the HTTP response header.

We have noticed that most of the issues surrounding license information are re-

lated to ambiguous entries. To resolve that, we manually created a mapping file[24] standardizing the set of possible license names and urls using the open source and knowledge license information[25]. As a result, we managed to normalize 123 (47.49%) of the datasets' license information.

To check the impact of the corrected fields, we seeded Roomba with the enriched profiles. Since Roomba uses file based cache system, we simply replaced all the datasets `json` files in the `\cache\datahub.io\datasets` folder with those generated in `\cache\datahub.io\enriched`. After running Roomba again on the enriched profiles, we observe that the errors percentage for missing `size` fields decreased by 32.02% and for `mimetype` fields by 50.93%. We also notice that the error percentage for missing `license_urls` decreased by 2.32%.

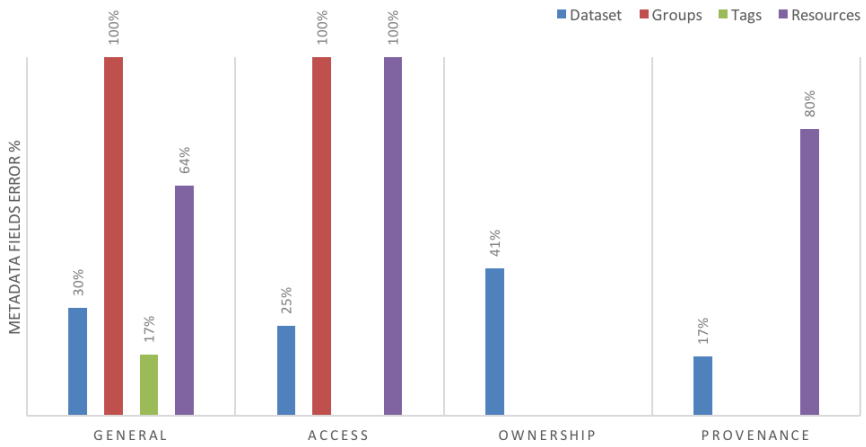

Figure 4.3: LOD Cloud error % by information type

4.7 Summary

In this chapter, we proposed a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. This approach applies several techniques 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 affect 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

[24] https://github.com/ahmadassaf/opendata-checker/blob/master/util/licenseMappings.json

[25] https://github.com/okfn/licenses

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 profiles 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 suffer 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 undefined 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 undefined values. We also showed that the automatic correction process can effectively enhance the quality of some information. We believe there is a need to have a community effort 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 [29]. However, the heterogeneous nature of sources reflects 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 field with several benchmarks and frameworks to grasp its dimensions [80, 20, 149]. 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 [101], thus directly relating to the ability of satisfying users' continuous needs.

Web documents that are by nature unstructured and interlinked require different 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 [115]. Ensuring data quality in Linked Open Data is a complex process as it consists of structured information supported by models, ontologies and vocabularies and contains queryable endpoints and links. This makes data quality assurance a challenge. Despite the fact that Linked Open Data quality is a trending and highly demanded topic, very few efforts are currently trying to standardize, track and formalize frameworks to issue scores or certificates 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 specific use case [23]. The dimensionality of data quality makes it dependent on the task and users requirements. For example, DBpedia [24] and YAGO [135] are knowledge bases containing data extracted from structured and

semi-structured sources. They are used in a variety of applications e.g., annotation systems [108], exploratory search [106] and recommendation engines [114]. However, their data is not integrated into critical systems e.g., life critical (medical applications) or safety critical (aviation applications) as its data quality is found to be insufficient. In this paper, we first propose a comprehensive objective framework to evaluate the quality of Linked Data sources. Secondly, we present an extensible quality measurement tool that 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. The aim of this paper is to provide researchers and practitioners with a comprehensive understanding of the objective issues surrounding Linked Data quality.

The framework we propose is based on a refinement of the data quality principles described in [6] and surveyed in [150]. 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.

Furthermore, we surveyed 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 which is a framework to assess and build dataset profiles with an extensible quality measurement tool and evaluate it by measuring the quality of the LOD cloud group. The results demonstrate that the general quality of LOD cloud needs more attention as most of the datasets suffer from various quality issues.

5.2 Data Quality Assessment

In [150], 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 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[1] provides a description of the published data quality in plain English. It aspires to act as a mark of approval that helps publishers understand how to publish good open data and users how to use it. It gives publishers the ability to provide assurance and support on their data while encouraging further improvements through an ascending scale.

ODI 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)[2] 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

[1] https://certificates.theodi.org/
[2] http://ldbc.eu/

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 publishing data that can ensure a certain level of uniformity reflecting directly data's usability [17]:

- **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 meaasures 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 [50] 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	Quality Indicator
Completeness	Dataset Level	1	Existence of supporting structured metadata [70]
		2	Supports multiple serializations [150]
		3	Has different data access points
		4	Uses datasets description vocabularies
		5	Existence of descriptions about its size
		6	Existence of descriptions about its structure (MIME Type, Format)
		7	Existence of descriptions about its organization and categorization
		8	Existence of information about the kind and number of used vocabularies [150]
	Links Level	9	Existence of dereferencable links for the dataset [70, 103, 60]
	Model Level	10	Absence of disconnected graph clusters [103]
		11	Absence of omitted top concept [70]

Continued on next page

Table 5.1 Objective Linked Data Quality Framework

Quality Attribute	Quality Category	ID	Quality Indicator
		12	Has complete language coverage [103]
		13	Absence of unidirectional related concepts [70]
		14	Absence of missing labels [103]
		15	Absence of missing equivalent properties [81]
		16	Absence of missing inverse relationships [81]
		17	Absence of missing domain or range values in properties [81]
Availability	Dataset Level	18	Existence of an RDF dump that can be downloaded by users [50][70]
		19	Existence of a queryable endpoint that responds to direct queries
		20	Existence of valid dereferencable URLs (respond to HTTP request)
Licensing	Dataset Level	21	Existence of human and machine readable license information [71]
		22	Existence of de-referenceable links to the full license information [71]
		23	Specifies permissions, copyrights and attributions [150]
Freshness	Dataset Level	24	Existence of timestamps that can keep track of its modifications [51]
Correctness	Dataset Level	25	Includes the correct MIME-type for the content [70]
		26	Includes the correct size for the content
		27	Absence of syntactic errors on the instance level [70]
	Links Level	28	Absence of syntactic errors [137]
		29	Use the HTTP URI scheme (avoid using URNs or DOIs) [103]
	Model Level	30	Contains marked top concepts [103]
		31	Absence of broader concepts for top concepts [103]
		32	Absence of missing or empty labels [2, 103]
		33	Absence of unprintable characters [2, 103] or extra white spaces in labels [136]
		34	Absence of incorrect data type for typed literals [70, 2]
		35	Absence of omitted or invalid languages tags [136, 103]
		36	Absence of terms without any associative or hierarchical relationships
Comprehensibility	Dataset Level	37	Existence of at least one exemplary RDF file [150]
		38	Existence of at least one exemplary SPARQL query [150]
		39	Existence of general information (title, URL, description) for the dataset
		40	Existence of a mailing list, message board or point of contact [50]
	Model Level	41	Absence of misuse of ontology annotations [103, 81]
		42	Existence of annotations for concepts [81]
		43	Existence of documentation for concepts [103, 81]
Provenance	Dataset Level	44	Existence of metadata that describes its authoritative information [51]
		45	Usage of a provenance vocabulary
		46	Usage of a versioning
Coherence	Model Level	47	Absence of misplaced or deprecated classes or properties [70]
		48	Absence of relation and mappings clashes [136]
		49	Absence of blank nodes [71]
		50	Absence of invalid inverse-functional values [70]
		51	Absence of cyclic hierarchical relations [132, 136, 103]
		52	Absence of undefined classes and properties usage [70]
		53	Absence of solely transitive related concepts [103]
		54	Absence of redefinitions of existing vocabularies [70]

Continued on next page

Table 5.1 Objective Linked Data Quality Framework

Quality Attribute	Quality Category	ID	Quality Indicator
		55	Absence of valueless associative relations [103]
Consistency	Model Level	56	Consistent usage of preferred labels per language tag [74, 103]
		57	Consistent usage of naming criteria for concepts [81]
		58	Absence of overlapping labels
		59	Absence of disjoint labels [103]
		60	Absence of atypical use of collections, containers and reification [70]
		61	Absence of wrong equivalent, symmetric or transitive relationships [81]
		62	Absence of membership violations for disjoint classes [70]
Security	Dataset Level	63	Uses login credentials to restrict access [150]
		64	Uses SSL or SSH to provide access to their dataset [150]

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 [103] and has documentation properties [111, 103]. 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 [70].
- Provides data in multiple serializations (N3, Turtle, etc.) [150].
- 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 DCAT[3] or VOID[4].
- 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 [150].

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

Links are considered to be complete if the dataset and all its resources have defined links [70, 103, 60]. Models are considered to be complete if they do not contain disconnected graph clusters [103]. 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) [103], do not contain omitted top concepts or unidirectional related concepts [70] and if they are not missing labels [103], equivalent properties, inverse relationships, domain or range values in properties [81].

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 [50, 70], its queryable endpoints e.g. SPARQL endpoint, are reachable and respond to direct queries and if all of its inbound and outbound links are dereferencable.

5.3.3 Correctness

A dataset is considered to be correct if it includes the correct MIME-type and size for the content [70] and doesn't contain syntactic errors [70]. Links are considered to be correct if they lack syntactic errors and use the HTTP URI scheme (avoid using URNs or DOIs) [103]. 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) [103]. Moreover, if they don't contain incorrect data type for typed literals [70][2], no omitted or invalid languages tags [136, 103], does not contain "orphan terms" (orphan terms are terms without any associative or hierarchical relationships and if the labels are not empty, do not contain unprintable characters [2, 103] or extra white spaces [136].

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) [74, 103], consistent preferred labels per language tag [103, 136], atypical use of collections, containers and reification [70], wrong equivalent, symmetric or transitive relationships [81], consistent naming criteria in the model [103, 81], overlapping labels in a

given language for concepts in the same scheme [103] and membership violations for disjoint classes [70, 81].

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 [51]. 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 [94].

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 [71], human readable license information in the documentation of the dataset or its source [71] and the indication of permissions, copyrights and attributions specified by the author [150].

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) [50]. A model is considered to be comprehensible if there is no misuse of ontology annotations and that all the concepts are documented and annotated [103, 81].

5.3.9 Coherence

Coherence is the ability to interpret data as expected by the publisher or vocabulary maintainer [70]. 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 [70], blank nodes [71], deprecated classes or properties [70], relations and mappings clashes [136], invalid inverse-functional values [70],

cyclic hierarchical relations [132, 136, 103], solely transitive related concepts [103], redefinitions of existing vocabularies [70] and valueless associative relations [103].

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 [150].

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 problems[5]. Syntactic RDF checkers are able to detect errors in RDF documents like the W3C RDF Validator[6], RDF:about validator and Converter[7] and The Validating RDF Parser (VRP)[8]. The RDF Triple-Checker[9] is an online tool that helps find typos and common errors in RDF data. Vapour[10] [19] is a validation service to check whether semantic Web data is correctly published according to the current best practices [17].

ProLOD [25], ProLOD++ [1], Aether [104] 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

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

long error-prone process especially when many contributors are working consecutively or collaboratively [137]. This can introduce deficiencies such as redundant concepts or conflicting relationships [62]. 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 [95] uses supervised machine learning techniques to learn concepts from user-provided examples. CROCUS [35] applies a cluster-based approach for instance-level 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

qSKOS[11] [103] scans SKOS vocabularies to provide reports on vocabulary resources and relations that are problematic. PoolParty checker[12] is an online service based on qSKOS. Skosify [136] 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 [121] evaluates OWL ontologies against a rules catalog and provides the user with a set of guidelines to solve them. ASKOSI[13] 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 [131, 139] the authors perform quality checking on OWL ontologies using integrity constraints involving the Unique Name Assumption (UNA) and the Closed World Assumption (CWA). Pellet[14] provides reasoning services for OWL ontologies. It incorporates a number of heuristics to detect and repair quality issues among disjoint properties, negative property assertions and reflexive, irreflexive, symmetric, and anti-symmetric properties. Eyeball[15] 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:Alerts[16] provides validation for many issues highlighted in [70] like misplaced, undefined or deprecated classes or properties.

[11] https://github.com/cmader/qSKOS
[12] http://www.poolparty.biz/
[13] http://www.w3.org/2001/sw/wiki/ASKOSI
[14] http://clarkparsia.com/pellet
[15] http://jena.sourceforge.net/Eyeball/
[16] http://swse.deri.org/RDFAlerts/

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 [107] is a framework for expressing quality assessment and fusion methods. It is implemented as a component of the Linked Data Integration Framework (LDIF)[17]. 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-configurable conflict 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 [57] 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)[18] 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 identified data quality problems.

5.4.3.2 Crowd-sourcing Approaches

There are several quality issues that can be difficult to spot and fix automatically. In [2] the authors highlight the fact that the RDFification process of some data can 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 [24]). 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 find and classify erroneous RDF triples and 2) Crowdsourcing through the Amazon Mechanical Turk[19].

[17] http://ldif.wbsg.de/
[18] http://spinrdf.org/
[19] https://www.mturk.com/

TripleCheckMate [87] is a crowdsourcing tool used by the authors to run out their assessment supported by a semi-automatic quality verification metrics. The tool allows users to select resources, identify and classify possible issues according to a predefined taxonomy of quality problems. It measures inter-rater agreements, meaning that the resources defined are checked multiple times. These features turn out to be extremely useful to analyze the performance of users and allow better identification 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 [43] is a generic Linked Data quality assessment framework. It can be easily extended through a declarative interface to integrate domain specific 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) [42]. Any additional quality metrics added to the framework should extend it.

RDFUnit[20] is a tool centered around the definition of data quality integrity constraints [86]. The input is a defined 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 specific semantics in the test cases.

LiQuate [127] 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 identifies redundancies (redundant label names for a given resource), incompleteness (incomplete links among a given set of resources) and inconsistencies (inconsistent links).

Quality Assessment of Data Sources (Flemming's Data Quality Assessment Tool)[21] calculates data quality scores based on manual user input. The user should assign weights to the predefined quality metrics and answer a series of questions regarding the dataset. These include, for example, the use of obsolete classes and properties by defining the number of described entities that are assigned disjoint classes, the usage

[20] http://github.com/AKSW/RDFUnit
[21] http://linkeddata.informatik.hu-berlin.de/LDSrcAss/datenquelle.php

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 [145] is the Open Refine[22] 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 Dashboard[23] 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 Validator[24] 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. Traditional link analysis has proven to be an effective way to measure the quality of Web documents search. Algorithms like PageRank [115] and HITS [84] 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 [31][34]. 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

[22] http://openrefine.org/
[23] http://labs.data.gov/dashboard/
[24] http://validator.lod-cloud.net/

short to provide reliable rankings as the types of the links can have a direct impact on the ranking computation [140]. The first adaption of PageRank for Semantic Web resources was the Ontology Rank algorithm implemented in the Swoogle search engine [45]. 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 [69] rankings are computed at query time based on two levels of granularity: resources and context graphs. DING [140] 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. DSNotify[25][66] 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 [60] 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 (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 [65][26] 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 [51] 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, freshness and plausibility. In [64] 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

[25] http://www.cibiv.at/~niko/dsnotify/
[26] http://trdf.sourceforge.net

Sindice [142] 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)[27] 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 [44]. 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 [33][28] 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 Extensible 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 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

[27] http://siren.sindice.com/
[28] http://labs.mondeca.com/sparqlEndpointsStatus/

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	Assessment Method
1	Check if there is a valid metadata file by issuing a `package_show` request to the CKAN API
2	Check if the `format` field for the dataset resources is defined and valid
3	Check the `resource_type` field with the following possible values `file`, `file.upload`, `api`, `visualization`, `code`, `documentation`
4	Check the resources `format` field for `meta/void` value
5	Check the resources `size` or the `triples` extras fields
6	Check the `format` and `mimetype` fields for resources
7	Check if the dataset has a `topic` tag and if it is part of a valid group in CKAN
9	Check if the dataset and all its resources have has a valid URI
18	Check if there is a dereferencable resource with a description containing string *dump*
19	Check if there is a dereferencable resource with `resource_type` of type `api`
20	Check if all the links assigned to the dataset and its resources are dereferencable
21	Check if the dataset contains valid `license_id` and `license_title`
22	Check if the `license_url` is dereferencable
24	Check if the dataset and its resources contain the following metadata fields `metadata_created`, `metadata_modified`, `revision_timestamp`, `cache_last_updated`
25	Check if the `content-type` extracted from the a valid HTTP request is equal to the corresponding `mimetype` field.
26	Check if the `content-length` extracted from the a valid HTTP request is equal to the corresponding `size` field.
28,29	Check that all the links are valid HTTP scheme URIs
37	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`
39	Check if the dataset and its tags and resources contain general metadata `id`, `name`, `type`, `title`, `description`, `URL`, `display_name`, `format`
40	Check if the dataset contain valid `author_email` or `maintainer_email` fields
44	Check if the dataset and its resources contain provenance metadata `maintainer`, `owner_org`, `organization`, `author`, `maintainer_email`, `author_email`
46	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 $\mathbf{D} = \{D_1, ...D_n\}$. We denote the set of resources $R_i = \{r_1, ..., r_k\}$, groups $G_i = \{g_1, ..., g_k\}$ and tags $T_i = \{t_1, ..., t_k\}$ for $D_i \in \mathbf{D}(i = 1, ..., n)$ by $\mathbf{R} = \{R_1, ..., R_n\}, \mathbf{G} = \{G_1, ..., G_n\}$ and $\mathbf{T} = \{T_1, ..., t_n\}$

respectively.

Our quality framework contains a set of measures $\mathbf{M} = \{M_1, ..., M_n\}$. We denote the set of quality indicators $Q_i = \{q_1, ..., q_k\}$ for $M_i \in \mathbf{M}(i = 1, ..., n)$ by $\mathbf{Q} = \{Q_1, ..., Q_n\}$. Each quality indicator has a weight, context and a score $Q_i < weight, context, score >$. In Roomba, all the weights are equal and set to 1. However, they can be adjusted manually to rank the quality indicators. Each Q_i of M_i (for i = 1,...n) is applied to one or more of the resources, tags or groups. The indicator context is defined where $\exists Q_i \in \mathbf{R} \cup \mathbf{G} \cup \mathbf{T}$.

The quality indicator score is based on a ratio between the number of violations $\mathbf{V}$ and the total number of instances where the rule applies $\mathbf{T}$ 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 dereferencable (QI.22).

$$Q \text{ weightedscore} = (V/T) * Q < weight > \tag{5.1}$$

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 $\mathbf{M}$ is calculated by dividing the weighted quality indicator scores sum by the total number of instances in its context, as the following formula shows:

$$M = 1 - ((\sum_{i=1}^{n} Q\text{i weightedscore})/ \mid Qi \text{ context} \mid) \tag{5.2}$$

5.5.2 Evaluation

In our evaluation, similarily to Roomba 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.

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

Table 5.3: Datasets chosen for the correctness evaluation

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 individual 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 profiles[29]. 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.

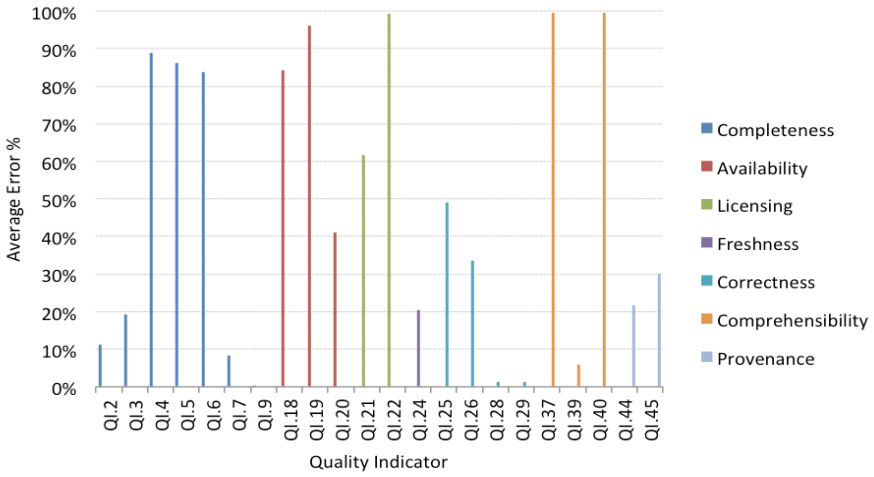

Figure 5.1: Average Error % per quality indicator for LOD group

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 C 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 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.

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

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.

```
=============================================================
                    Dataset Quality 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 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 [81]. 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 this 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 information[30]. We have also used the open source and knowledge license information[31] to normalize license information and add extra metadata like the domain, maintainer and open data conformance.

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

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

[31] https://github.com/okfn/licenses

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

Table 5.4: Functional Comparison of Automatic Linked Data quality Tools

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 **Mark** 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 **Mark** 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.

Bob 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 **Bob** to make more infomred decisions on which dataset to use in his report.

Part II

Data Integration in the Enterprise

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 further 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 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 [97]. 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 [53], 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.

6.1.1 Data Warehousing (DW)

A *Data Warehouse (DW)* 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 process is applied to import the data to the warehouse.

Querying DWs 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 multidimensional fashion. 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.

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 [128]. BI consists of various number of components including Data Integration, Data Quality and Data Warehousing among others.

6.2.1 Multidimensional Model

Traditional relational model is efficient in performing "online" transactions. However, it had clear shortcomings when the objective was to analyze large scale data. The multidimensional model was 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 containment-like 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.
- **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.

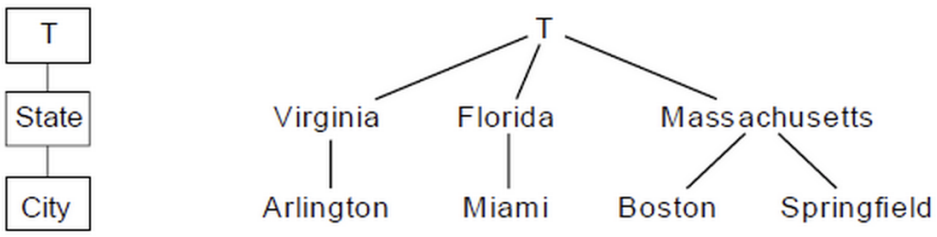

Figure 6.1: Example of a hierarchical geography dimension

- **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).

6.2.2 SAP BI Application Suite

The SAP BI application suite can be divided into the following main areas:

- **Analysis Solutions**: Empower 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 interaction with their data. SAP Lumira which is used by our analyst **Bob** is part of these solutions.
- **Reporting Solutions**: Offers powerful interfaces that enable analysts and non-technical users to ask spontaneous and iterative business questions about their 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 platform that is best suited for performing real-time analytics, and developing

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

and deploying real-time applications. At the core of this real-time data platform is the SAP HANA database which is fundamentally different than any other database engine in the market today. It 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**
- **Operational Reporting**: Provides real-time insights 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. It does this through the power of its in-database predictive algorithms and its R integration capability. With its text search/analysis capabilities SAP HANA also provides a robust way to leverage unstructured data.

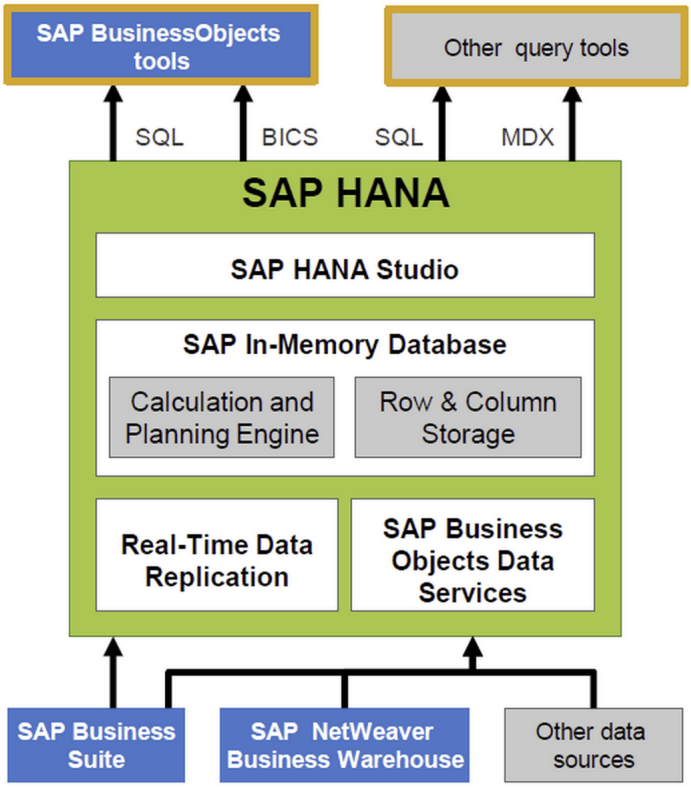

Figure 6.2: SAP's High Performance Analytic Appliance (HANA) in SAP BI suite

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; so 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; If i have my product details scattered in more than one table; but i need to model my product as one business entity; i need to create an attribute view that aggregates data from different tables into one single entity which is product.

- **Analytical Views**: Used for calculation and aggregation. Adds transactional tables and measures (key figures), calculates aggregates (e.g. No. 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 not possible with other views and uses SQLScript.

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; its a presentation logic on client sides that encapsulated control flow logic and calculation logic 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:

- **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.

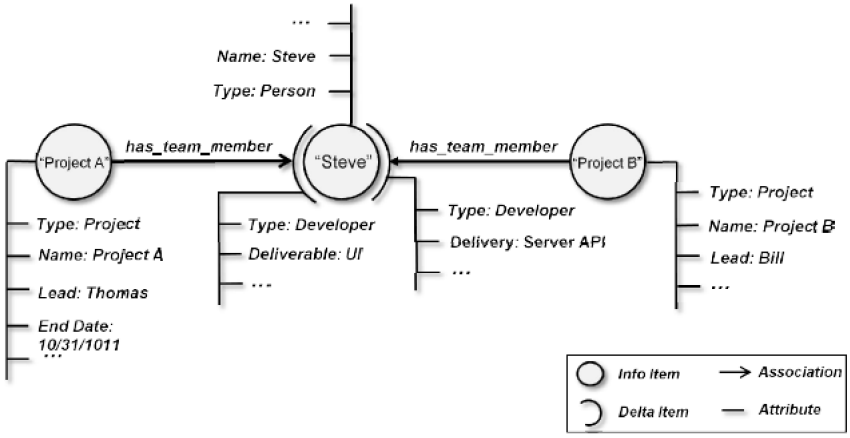

Figure 6.3: AIS data model

- **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

The social web (Web 2.0) is about websites and services designed and developed to support and foster social interactions [120]. The social web spans services like blogs, wikis, crowdsourcing and social media services. These services are often accompanied with APIs that allow rapid community driven expansion of complementary services. In this thesis, we use the following popular social media services:

- **Twitter**[2]: A service allowing users to send rich short messages of 140 characters called "tweets" or "microposts". Twitter allow users to "follow" each other and start private conversations. Users can interact with tweets by replying or forwarding (re-tweeting) them. Tweets can contain multimedia parts like photos or videos and can be tagged with certain keywords preceded by the hash character called "hashtags" e.g. #tag.
- **Google+**[3]: A social media service owned by Google focusing on driving interest-based conversations and interactions. In its core, Google+ evolves around the concept of "circles" which enable users organize people into groups or lists. Similarly to Twitter, posts can be tagged with hash-tags entered manually by users or added automatically by Google+.

[2] http://twitter.com
[3] https://plus.google.com

- **Stack Exchange**[4]: A question and answer group of websites. Each website specializes in a topic like technology, politics, food, etc. Users are encouraged to provide answers and helpful comments by providing a reputation award system allowing the content to be self-moderating.
- **YouTube**[5]: A video sharing website owned by Google. The site allows users to upload, view and share videos. Moreover, YouTube allows live streaming of events with the ability to interact with the video feed through comments.
- **Vimeo**[6]: Another video sharing website with a more focused community of professionals in various areas than YouTube.
- **Slideshare**[7]: A slide hosting service where users can upload their presentations in various formats to be viewed and shared on the website. The website also supports documents, videos and webinars and acts as an educational and e-learning hub.

[4] http://stackexchange.com
[5] http://youtube.com
[6] http://vimeo.com
[7] http://slideshare.com

Chapter 7

Data Integration in the Enterprise

Companies have traditionally performed business analysis based on transactional data stored in legacy relational databases. The enterprise data available for decision makers was typically relationship management or enterprise resource planning data. However social media feeds, weblogs, sensor data, or data published by governments or international organizations are nowadays becoming increasingly available [29].

The quality and amount of structured knowledge available make it now feasible for companies to mine this huge amount of public data and integrate it in their next-generation enterprise information management systems. Analyzing this new type of data within the context of existing enterprise data should bring them new or more accurate business insights and allow better recognition of sales and market opportunities [93].

These new distributed sources, however, raise tremendous challenges. They have inherently different file formats, access protocols or query languages. They possess their own data model with different ways of representing and storing the data. Data across these sources may be noisy (e.g. duplicate or inconsistent), uncertain or be semantically similar yet different [14]. Integration and provision of a unified view for these heterogeneous and complex data structures therefore require powerful tools to map and organize the data.

Establishing data knowledge bases in the enterprise can facilitate data integration services [54]. In this chapter, we present our work in integrating DBpedia as an internal knowledge base. We further present a set of services we implemented on top of DBpedia that allow entity disambiguation and enhance schema matching. These services enable business users to semi-automatically combine potentially noisy data residing in heterogeneous silos. Semantically related data is identified and appropriate mappings are suggested to users. On user acceptance, data is aggregated and can be visualized directly or exported to Business Intelligence reporting tools. Finally, we perform a reverse engineering of the Google Knowledge graph panel to find out what are the most "important" properties for an entity according to Google. We compare these results with a survey we conducted on 152 users and show how we can

represent and explicit this knowledge using the Fresnel vocabulary.

7.1 Enterprise Knowledge Bases

A Knowledge Base (KB) is a large repository of structured and unstructured information representing facts about the general world or specific domains. DBpedia [1] is an example of a general knowledge base that will be used in this chapter. It is a crowd-sourced community effort to extract information from Wikipedia and present it in structured accessible formats [24].

DBpedia provides dumps which are split into several parts making it easy to import and experiment with the data. In this part, we are mainly interested in the following datasets:

- **Mapping-based Types**: Contains triples that assign types from the DBpedia ontology to the entities extracted from Wikipedia.
- **Mapping-based Properties**: Contains properties extracted from Wikipedia. Since we can have different names for the same attribute (e.g. `birthplace` and `placeofbirth`), DBpedia uses a mapping-based approach to unify these attributes and generate high quality linked data.
- **Extended Abstracts**: Contains the first section of Wikipedia articles.
- **Images**: Images and their corresponding thumbnails together with a link to the license.
- **Inter-language Links**: Links between IRIs from different languages to the English IRI.

DBpedia is a modeled as an RDF graph. A natural way to import DBpedia into SAP HANA would have been to its graph engine, AIS. To do that, we would need to map the RDF triples to the AIS data model (see section 6.3.2). To do that, we need to:

- Create a new Term for every triple's distinct predicate. If the object of the triple is not a subject but a literal, the Term gets a technical type corresponding to the datatype (if not known the the type string is used). If the object refers to another Info Item, the Term gets the technical type for being an Association.
- Create a new Info Item for each distinct RDF subject.

[1] http://dbpedia.org

Table	Columns
ABSTRACTS	uri, abstract
ASSOCIATIONS	source, type, target
INTERLANGUAGE	uri, sameas
PROPERTIES	uri, typ, value
TYPES	uri, type, incomingno, order

Table 7.1: Tables structure for DBpedia in HANA column store

- Store all literals as Attributes that assigned to the Info Item which corresponds to the subject of the triple.
- Create Associations to the subject's Info Item that point to the Info Item which has the URI of the object of the triple.

However, since AIS is still under major development, various limitations and performance issues prevented us to import a reliable and functional version of DBpeida. As a result, we decided to import DBpedia into HANA's column store. Table 7.1 shows the tables structure used.

7.1.1 Entity Disambiguation with DPbedia in SAP HANA

After successfully, importing DBpedia into HANA's column store, we would like to create a service that is able to disambiguate a query string (Full documentation of the API is found in appendix E). HANA has a built-in (fuzzy) text search function that can be used. However, relying on string matching only is not sufficient. We have combined an page-rank-like approach that takes into account the number of incoming associations as shown in equation 7.1. While the number of outgoing associations can vary from entity to entity and is rather an indicator for how well described an entity in DBpedia is, the number of incoming associations is less dependent on one single entity. It takes into account how many other entities link to the entity. The more entities link to an entity, the more popular it is.

$$\frac{incomingWeight}{largestIncomingNo} \times incomingNo + txtScore \tag{7.1}$$

Table 7.2 shows the results of combining HANA's fuzzy text search score with the entity link-based rank with an `incomingWeight` parameter of 0.15 for query string "apple". From these results, we notice that a very high text score directly affects the overall score even if it has zero incoming associations compared to entities with significant number of incoming associations. To overcome this downfall, we need to use our incomingNoWeight to also decrease the combined score if an entity has

URI	Text Search Score	No. of Incoming	Combined Score
Apple	1.00000	31	1.00000
Apple_Inc	0.70711	393	0.85711
Apple_Records	0.70711	362	0.84527
Apple_II	0.70711	261	0.80673
Apple_IIGS	0.70711	95	0.74337
Apple_Corps	0.70711	39	0.72199
Fiona_Apple	0.70711	39	0.72199
Apple_IIe	0.70711	12	0.71169
Apple_Hong	0.70711	7	0.70978
Apple_DOS	0.70711	6	0.70940

Table 7.2: Results of combining text search score with the number of incoming associations for query "apple"

URI	Text Search Score	No. of Incoming	Combined Score
Apple	1.00000	31	0.87366
Apple_Inc	0.70711	393	0.85711
Apple_Records	0.70711	362	0.83344
Apple_II	0.70711	261	0.75634
Apple_IIGS	0.70711	95	0.62963
Apple_Corps	0.70711	39	0.58688
Fiona_Apple	0.70711	39	0.58688
Apple_IIe	0.70711	12	0.56627
Apple_Hong	0.70711	7	0.56245
Apple_DOS	0.70711	6	0.70940

Table 7.3: Results of the enhanced equation 7.2 and its affect on the overall score for query "apple"

very few or no incoming associations as shown in equation 7.2. Table 7.3 shows the enhanced results using this equation.

$$\frac{incNoWeight}{largestIncNo}\left(incNo - (largestIncNo - incNo)\right) + txtScore \tag{7.2}$$

7.1.2 Important Properties for Entities

Entities are generally described with a lot of properties. However, not all properties have the same importance. Some properties are considered as keys for performing instance matching tasks while other properties are generally chosen for quickly providing a summary of the key facts attached to an entity. In contrast to entities, It is difficult to assess which ones are more "important".

In this section we provide a method enabling to select what properties should be

used when depicting the summary of an entity, for example in a multimedia question answering system such as QakisMedia[2] or in a second screen application providing more information about a particular TV program[3]. We reverse engineered the Google Knowledge graph panel (see figure 7.1) to find out what are the most "important" properties for an entity according to Google. We compare these results with a survey we conducted on 152 users.

7.1.2.1 Reverse Engineering the Google KG Panel

Web scraping is a technique for extracting data from Web pages. We aim at capturing the properties depicted in the Google Knowledge Panel (GKP) that are injected in search result pages [15]. We have developed a Node.js application that queries all DBpedia concepts that have at least one instance which is `owl:sameAs` with a Freebase resource in order to increase the probability that the search engine result page (SERP) for this resource will contain a GKP. We assume in our experiments that the properties displayed for an entity are type and context dependent (country, time, query) which can affect the results. Moreover, we filter out generic concepts by excluding those who are direct subclasses of `owl:Thing` since they will trigger ambiguous queries. We obtained a list of 352 concepts[4].

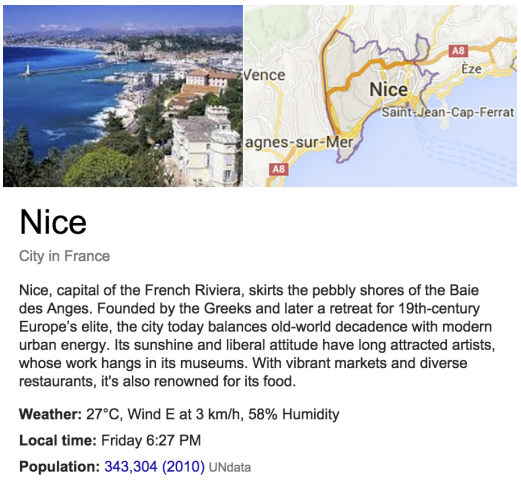

Figure 7.1: Google knowledge graph panel for the city of Nice, France

For each of these concepts, we retrieve n instances (in our experiment, n was equal to 100 random instances). For each of these instances, we issue a search query to Google containing the instance label. Google does not serve the GKP for all user agents and we had to mimic a browser behavior by setting the $User - Agent$ to a particular browser. We use CSS selectors to check the existence of and to extract data from a GKP. An example of a query selector is *._om* (all elements with class name *_om*) which returns the property DOM element(s) for the concept described in the GKP. From our experiments, we found out that we do not always

[2] http://qakis.org/
[3] http://www.linkedtv.eu/demos/linkednews/
[4] See also the SPARQL query at http://goo.gl/EYuGm1

Algorithm 1 Google Knowledge Panel reverse engineering algorithm

```
INITIALIZE equivalentClasses(DBpedia, Freebase) AS vectorClasses
Upload vectorClasses for querying processing
Set n AS number-of-instances-to-query
for each conceptType ∈ vectorClasses do
   SELECT n instances
   listInstances ← SELECT-SPARQL(conceptType, n)
   for each instance ∈ listInstances do
      CALL http://www.google.com/search?q=instance
      if knowledgePanel exists then
         SCRAP GOOGLE KNOWLEDGE PANEL
      else
         CALL http://www.google.com/search?q=instance + conceptType
         SCRAP GOOGLE KNOWLEDGE PANEL
      end if
      gkpProperties ← GetData(DOM, EXIST(GKP))
   end for
   COMPUTE occurrences for each prop ∈ gkpProperties
end for
gkpProperties
```

get a GKP in a SERP. If this happens, we try to disambiguate the instance by issuing a new query with the concept type attached. However, if no GKP was found again, we capture that for manual inspection later on. Listing 1 gives the high level algorithm for extracting the GKP. The full implementation can be found at `https://github.com/ahmadassaf/KBE`. Instructions for installing and running the tool are available in section B.3. We finally observe that this experiment is only valid for the English Google.com search results since GKP varies according to top level names.

7.1.2.2 Evaluation

We conducted a user survey in order to compare what users think should be the important properties to display for a particular entity and what the GKP shows.

User survey

We set up a survey[5] on February 25th, 2014 and for three weeks in order to collect the preferences of users in term of the properties they would like to be shown for a particular entity. We select only one representative entity for nine classes: `TennisPlayer`, `Museum`, `Politician`, `Company`, `Country`, `City`, `Film`, `SoccerClub` and `Book`. 152 participants have provided answers, 72% from academia, 20% coming from the industry and 8% having not declared their affiliation. 94% of the respondents have heard about the Semantic Web while 35% were not familiar with specific visualization tools. The detailed results[6] show the ranking of the top properties for each entity. We only keep the properties having received at least 10% votes for com-

[5] The survey is at `http://eSurv.org?u=entityviz`

[6] `https://github.com/ahmadassaf/KBE/blob/master/results/agreement-gkp-users.xls`

paring with the properties depicted in a KGP. We observe that users do not seem to be interested in the `INSEE code` identifying a French city while they expect to see the `population` or the `points of interest` of this city.

Comparison with the Knowledge Graphs

The results of the Google Knowledge Panel (GKP) extraction[7] clearly show a long tail distribution of the properties depicted by Google, with a top N properties (N being 4, 5 or 6 depending on the entity) counting for 98% of the properties shown for this type. We compare those properties with the ones revealed by the user study. Table 7.4 shows the agreement between the users and the choices made by Google in the GKP for the 9 classes. The highest agreement concerns the type `Museum` (66.97%) while the lowest one is for the `TennisPlayer` (20%) concept. We think properties for museums or books are more stable than for types such as person/agent which vary significantly. We acknowledge the fact that more than one instance should be tested in order to draw meaningful conclusion regarding what are the important properties for a type.

Classes	TennisPlayer	Museum	Politician	Company	Country	City	Film	SoccerClub	Book
Agr.	20%	66.97%	50%	40%	60%	60%	60%	50%	60%

Table 7.4: Agreement on properties between users and the Knowledge Graph Panel

With this set of 9 concepts, we are covering 301,189 DBpedia entities that have an existence in Freebase, and for each of them, we can now empirically define the most important properties when there is an agreement between one of the biggest knowledge base (Google) and users preferences.

Modeling the preferred properties with Fresnel

Fresnel[8] is a presentation vocabulary for displaying RDF data. It specifies *what* information contained in an RDF graph should be presented with the core concept `fresnel:Lens` [119]. We use the Fresnel and PROV-O ontologies[9] to explicitly represent what properties should be depicted when displaying an entity. This dataset can now be re-used as a configuration for any consuming application (see Appendix D for a snippet of the generated Fresnel file).

[7] https://github.com/ahmadassaf/KBE/blob/master/results/survey.json
[8] http://www.w3.org/2005/04/fresnel-info/
[9] http://www.w3.org/TR/prov-o/

7.2 Enhancing Schema Matching

Schema matching is typically used in business to business integration, metamodel matching, as well as ETL processes. For non-IT specialists the typical way of comparing financial data from two different years or quarters, for example, would be to copy and paste the data from one Excel spreadsheet into another one, thus creating redundancies and potentially introducing copy-and-paste errors. By using schema matching techniques it is possible to support this process semi-automatically, i.e. to determine which columns are similar and propose them to the user for integration. This integration can then be done with appropriate business intelligence tools to provide visualizations.

One of the problems in performing the integration is the quality of data. The columns may contain data that is noisy or incorrect. There may also be no column headers to provide suitable information for matching. A number of approaches exploit the similarities of headers or similarities of types of column data. In this section, we propose a new approach that exploits semantic rich typing provided by popular datasets from the Linked Data cloud.

7.2.1 Related Work

While schema matching has always been an active research area in data integration, new challenges are faced today by the increasing size, number and complexity of data sources and their distribution over the network. Data sets are not always correctly typed or labeled and that hinders the matching process.

In the past, some work has tried to improve existing data schemas [112] but literature mainly covers automatic or semi-automatic labeling of anonymous data sets through Web extraction. Examples include [125] that automatically labels news articles with a tree structure analysis or [148] that defines heuristics based on distance and alignment of a data value and its label. These approaches are however restricting label candidates to Web content from which the data was extracted. [39] goes a step further by launching speculative queries to standard Web search engines to enlarge the set of potential candidate labels. More recently, [100] applies machine learning techniques to respectively annotate table rows as entities, columns as their types and pairs of columns as relationships, referring to the YAGO ontology. The work presented aims however at leveraging such annotations to assist semantic search queries construction and not at improving schema matching.

With the emergence of the Semantic Web, new work in the area has tried to exploit Linked Data repositories. The authors of [138] present techniques to automatically infer a semantic model on tabular data by getting top candidates from Wikitology [49]

and classifying them with the Google page ranking algorithm. Since the authors' goal is to export the resulting table data as Linked Data and not to improve schema matching, some columns can be labeled incorrectly, and acronyms and languages are not well handled [138]. In the Helix project [67], a tagging mechanism is used to add semantic information on tabular data. A sample of instances values for each column is taken and a set of tags with scores are gathered from online sources such as Freebase[10]. Tags are then correlated to infer annotations for the column. The mechanism is quite similar to ours but the resulting tags for the column are independent of the existing column name and sampling might not always provide a representative population of the instance values.

7.2.2 Proposition

Open Refine (formerly Google Refine) [11] is a tool designed to quickly and efficiently process, clean and eventually enrich large amounts of data with existing knowledge bases such as Freebase. The tool has however some limitations: it was initially designed for data cleansing on only one data set at a time, with no possibility to compose columns from different data sets. Moreover, Open Refine has some strict assumptions over the input of spreadsheets which makes it difficult to identify primitive and complex data types.

The Automapping Core (AMC) [117] is a novel framework that supports the construction and execution of new matching components or algorithms. AMC contains several matching components that can be plugged and used, like string matchers (Levenshtein, JaroWinkler, etc.), data types matchers and path matchers. It also provides a set of combination and selection algorithms to produce optimized results (weighted average, average, sigmoid, etc.).

Open Refine makes use of a modular web application framework similar to OSGi called Butterfly[12]. The server-side written in Java maintains states of the data (undo/redo history, long-running processes, etc.) while the client-side implemented in Javascript maintains states of the user interface (facets and their selections, view pagination, etc.). Communication between the client and server is done through REST web services.

RUBIX is the framework we created to enable business users to semi-automatically combine potentially noisy data residing in heterogeneous silos. Semantically related data is identified and appropriate mappings are suggested to users. On user acceptance, data is aggregated and can be visualized directly or exported to Business

[10] http://www.freebase.com/
[11] http://openrefine.org/
[12] http://code.google.com/p/simile-butterfly/

Intelligence reporting tools. We first map cell values with instances and column headers with types from popular data sets from the Linked Open Data Cloud. RUBIX leverages Open Refine and defines three new Butterfly modules to extend the server's functionality (namely Match, Merge and Aggregate modules) and one JavaScript extension to capture user interaction with these new data matching capabilities.

7.2.3 Activity Flow

This section presents the sequence of activities and interdependencies between these activities when using our framework. 7.2 gives an outline of these activities.

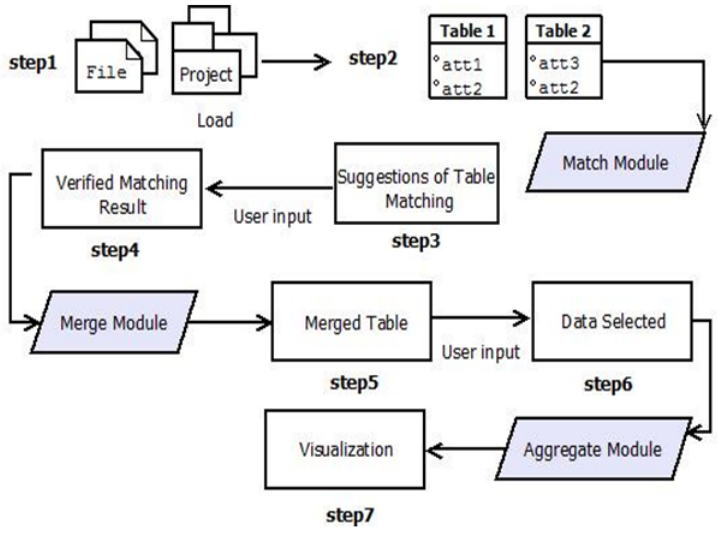

Figure 7.2: RUBIX Activity Workflow

The data sets to match can be contained in files (e.g. csv, Excel spreadsheets, etc.) or defined in Open Refine projects (step 1). The inputs for the match module are the source and target files and/or projects that contain the data sets. These projects are imported into the internal data structure (called schema) of the AMC [116] (step 2). The AMC then uses a set of built-in algorithms to calculate similarities between the source and target schemas on an element basis, i.e. column names in the case of spreadsheets or relational databases. The output is a set of similarities, each containing a triple consisting of source schema element, target element, and similarity between the two. These results are presented to the user in tabular form (step 3) such that s/he can check, correct, and potentially complete the mappings (step 4) as shown in figure 7.3.

Once the user has completed the matching of columns, the merge information is sent back to Open Refine, which calls the merge module. This module creates a new project, which contains a union of the two projects where the matched columns of

Reason for Trip	Reason for Trip	1.0	
Begins On	Trip Begins On	0.9166667	
Ends On	Trip Ends On	0.9	
Total	Total Cost	0.8666667	
Amount	Receipt Amount	0.8571428	
Pd by Comp	Paid by Company	0.8452381	
Trip	Trip Number	0.8333334	
Pers.No.	Sequential no.	0.7777778	
M/Km	Total Miles/Km	0.775	
Curr.	Currency	0.7094356	
Crcy	Currency	0.7094356	

Figure 7.3: Screenshot showing the results mapping view in RUBIX

the target project are appended to the corresponding source columns (step 5). The user can then select the columns that s/he wants to merge and visualize by dragging and dropping the required columns onto the fields that represent the x and y axes (step 6).

Once the selection has been performed, the aggregation module merges the filtered columns and the result can then be visualized (step 7). As aggregation operations can quickly become complex, our default aggregation module can be replaced by more advanced analytics on tabular data. The integration of such a tool is part of future work.

7.2.4 Data Reconciliation

Reconciliation enables entity resolution, i.e. matching cells with corresponding typed entities in case of tabular data. Google Refine already supports reconciliation with Freebase but requires confirmation from the user. For medium to large data sets, this can be very time-consuming. To reconcile data, we therefore first identify the columns that are candidates for reconciliation by skipping the columns containing numerical values or dates. We then use the disambiguation API in section 7.1.1 to query for each cell of the source and target columns the list of typed entities candidates. Results are cached in order to be retrieved by our similarity algorithms.

7.2.5 Matching Unnamed and Untyped Columns

The AMC has the ability to combine the results of different matching algorithms. Its default built-in matching algorithms work on column headers and produce an overall similarity score between the compared schema elements. It has been proven that combining different algorithms greatly increases the quality of matching results [117][134]. However, when headers are missing or ambiguous, the AMC can only exploit domain intersection and inclusion algorithms based on column data. We have therefore implemented three new similarity algorithms that leverage the

rich types retrieved from Linked Data in order to enhance the matching results of unnamed or untyped columns. They are presented below.

7.2.5.1 Cosine Similarity

The first algorithm that we implemented is based on vector algebra. Let v be the vector of ranked candidate types returned by Freebase for each cell value of a column. Then:

$$v := \sum_{i=1}^{K} a_i * \overrightarrow{t_i} \tag{7.3}$$

where a_i is the score of the entry and $\overrightarrow{t_i}$ is the type returned the disambiguation API. The vector notation is chosen to indicate that each distinct answer determines one dimension in the space of results.

Each cell value has now a weighted result set that can be used for aggregation to produce a result vector for the whole column. The column result V is then given by:

$$V? \sum_{i=1}^{n} v_i \tag{7.4}$$

We compare the result vector of candidate types from the source column with the result vector of candidate types from the target column. Let W be the result vector for the target column, then the similarity s between the columns pair can be calculated using the absolute value of the cosine similarity function:

$$s? \frac{|(V * W)|}{\|V\| * \|W\|} \tag{7.5}$$

7.2.5.2 Pearson Product-Moment Correlation Coefficient (PPMCC)

The second algorithm that we implemented is PPMCC, a statistical measure of the linear independence between two variables (x, y) [89]. In our method, x is an array that represents the total scores for the source column rich types, y is an array that represents the mapped values between the source and the target columns. The values present in x but not in y are represented by zeros. We have:

$$\begin{aligned} SourceColumn\ & [\{R_1, C_{sr1}\}, \{R_2, C_{sr2}\}, \{R_3, C_{sr3}\} \dots\ \{R_n, C_{srn}\}] \\ TargetColumn\ & [\{R_1, C_{tr1}\}, \{R_2, C_{tr2}\}, \{R_3, C_{tr3}\} \dots\ \{R_n, C_{trn}\}] \end{aligned} \tag{7.6}$$

Where $R_1,\ R_2, \dots, R_n$ are different rich type values retrieved from Freebase,

$C_{sr1},\ C_{sr2}, \ldots,\ C_{srn}$ are the sum of scores for each corresponding r occurrence in the source column, and $C_{tr1},\ C_{tr2},\ \ldots,\ C_{trn}$ are the sum of scores for each corresponding r occurrence in the target column.

The input for PPMC consists of two arrays that represent the values from the source and target columns, where the source column is the column with the largest set of rich types found. For example:

$$X = [C_{sr1}, C_{sr2}, C_{sr4},\ \ldots,\ C_{srn}]\, Y = [0, C_{tr2}, C_{tr4},\ \ldots,\ C_{trn}] \tag{7.7}$$

Then the sample correlation coefficient (r) is calculated using:

$$r = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2}} \tag{7.8}$$

Based on a sample paired data(x_i, y_i), the sample PPMCC is:

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \overline{x}}{s_x} \right) \left(\frac{y_i - \overline{y}}{s_y} \right) \tag{7.9}$$

Where $\left(\frac{x_i - \overline{x}}{s_x}\right)$, $\overline{x}$ and s_x are the standard score, sample mean and sample standard deviation, respectively.

7.2.5.3 Spearman's Rank Correlation Coefficient

The last algorithm that we implemented to match unnamed and untyped columns is Spearman's rank correlation coefficient. It applies a rank transformation on the input data and computes PPMCC afterwards on the ranked data. In our experiments we used Natural Ranking with default strategies for handling ties and NaN values. The ranking algorithm is however configurable and can be enhanced by using more sophisticated measures.

7.2.6 Column Labeling

We showed in the previous section how to match unnamed and untyped columns. Column labeling is however beneficial as the results of our previous algorithms can be combined with traditional header matching techniques to improve the quality of matching.

Rich types retrieved from Freebase are independent from each other. We need to find a method that will determine normalized score for each type in the set by balancing the proportion of high scores with the lower ones. We used Wilson score

interval for a Bernoulli parameter that is presented in the following equation:

$$w = \left(\widehat{p} + \frac{z^2_{\alpha/2}}{2n} \pm z_{\alpha/2} \sqrt{\left[\hat{p}\,(1 - \hat{p}) + z^2_{\alpha/2}/4n \right] / n} \right) / \left(1 + z^2_{\alpha/2}/n \right) \quad (7.10)$$

Here $\hat{p}$ is the average score for each rich type, n is the total number of scores and $z_{\alpha/2}$ is the score level; in our case it is 1.96 to reflect a score level of 0.95.

7.2.7 Handling Non-String Values

So far, we have covered several methods to identify the similarity between "String" values, but how about other numeral values such as dates, money, distance, etc. ? For this purpose, we have implemented some basic type identifier that can recognize dates, money, numerical values, numerals used as identifiers. This will help us in better match corresponding entries. Adjusting AMC's combination algorithms can be of great importance at this stage. For example, assigning weights to different matchers and tweaking the configuration can yield more accurate results.

7.2.8 Experiments

We present in this section results from experiments we conducted using the different methods described above. To appreciate the value of our approach, we have used a real life scenario that exposes common problems faced by the management in SAP. The data we have used come from two different SAP systems: the Event tracker and the Travel Expense Manager.

The Event Tracker provides an overview of events (Conferences, Internal events, etc.) that SAP Research employees contribute to or host. The entries in this system contain as much information as necessary to give an overview of the activity like the activity type and title, travel destination, travel costs divided into several sub categories (conference fees, accommodation, transportation and others), and duration related information (departure, return dates). Entries in the Event Tracker are generally entered in batches as employees fill in their planned events that they wish to attend or contribute to at the beginning of each year. Afterwards, managers can either accept or reject these planned events according to their allocated budget.

On the other hand, the Travel Expense Manager contains the actual expenses data for the successfully accepted events. This system is used by employees to enter their actual trip details in order to claim their expenses. It contains more detailed information and aggregated views of the events, such as the total cost, duration calculated in days, currency exchange rates and lots of internal system tags and

identifiers.

Matching reports from these two systems is of great benefit to managers to organize and monitor their allocated budget. They mainly want to:

1. Find the number of the actual (accepted) travels compared with the total number of entered events.

2. Calculate the deviation between the estimated and actual cost of each event.

However, matching from these two sources can face several difficulties that can be classified in two categories: column headers and cells. Global labels (or column headers as we are dealing with spreadsheet files) can have the following problems:

1. Missing labels: importing files into Google Refine with empty headers will result in assigning that column a dummy name by concatenating the word "column" with a number starting from 0.

2. Dummy labels or semantically unrelated names: this is a common problem especially from the data coming from the Travel Expense Manager. This can be applied to columns that are labeled according to the corresponding database table (i.e. lbl_dst to denote destination label). Moreover, column labels do not often convey the semantic type of the underlying data.

The second category of difficulties is at cell (single entry) level:

1. Detecting different date formats: we have found out that dates field coming from the two systems have different formats. Moreover, the built-in type detection in Google Refine converts detected date into another third format.

2. Entries from different people can be made in different languages.

3. Entries in the two systems can be incomplete, an entry can be shortened automatically by the system. For example, selecting a country in the Travel Expense Manager will result in filling out that country code in the exported report (i.e. France = FR).

4. Inaccurate entries: this is one of the most common problems. Users enter sometimes several values in some fields that correspond to the same entity. For example, in the destination column, users can enter the country, the airport at the destination, the city or even the exact location of the event (i.e. office location).

The data used in our evaluation consists of around 60 columns and more than 1000 rows. Our source data set will be the data coming from Event Tracker, and our target data set will be the data from the Travel Expense Manager.

By manually examining the two data sets, we have found out that most of the column headers in the source table exist and adequately present the data. However, we have noticed few missing labels in the target table and few ambiguous column headers. We have detected several entries in several languages: the main language is English but we have also identified French, German. Destination field had entries in several formats: we have noticed airport names, airports by their IATA code, country codes, and cities.

Running AMC with its default matchers returns the matching results shown in Table 7.5.

Source Column	**Target Column**	**Similarity Score**
Reason for Trip	Reason for Trip	1
Begins On	Trip Begins On	0.8333334
Ends On	Trip Ends On	0.8
Total	Total Cost	0.7333335
Trip	Trip Destination	0.72727275
Amount	Receipt Amount	0.7142875
Pd by Comp	Paid by Company	0.6904762
Period	Period Number	0.6666667
Pers.No.	Sequential no.	0.5555556
M/Km	Total Miles/Km	0.55
Curr.	Currency	0.5
Crcy	Currency	0.5

Table 7.5: Similarity Scores Using the AMC Default Matching Algorithms

The AMC has perfectly matched the two columns labeled “Reason for Trip” using name and data type similarity calculations (the type here was identified as a String). Moreover, it has computed several similarities for columns based on the pre-implemented String matchers that were applied on the column headers and the primitive data types of the cells (Integer, Double, Float, etc.). However, there is no alignment found between the other columns since their headers are not related to each other, although the actual cell values can be similar. AMC’s default configuration has a threshold of 50%, so any similarity score below that will not be shown.

The Cosine Similarity algorithm combined with the AMC default matchers produces the results shown in Table 7.6.

We notice that we have an increased number of matches (+2), and that the similarity score for several matches has improved. For example, the “tr_dst” column is

Source Column	Target Column	Similarity Score
Reason for Trip	Reason for Trip	1
tr_dst		0.9496432
Begins On	Trip Begins On	0.9166667
Ends On	Trip Ends On	0.9
Amount	Receipt Amount	0.8571428
Curr.	Currency	0.75
Crcy	Currency	0.75
Total	Total Cost	0.7333335
Trip	Trip Destination	0.7321428
Pd by Comp	Paid by Company	0.6904762
Period	Period Number	0.6666667
Trip	Trip Number	0.6666667
Pers.No.	Sequential no.	0.5555556
M/Km	Total Miles/Km	0.55

Table 7.6: Similarity Scores Using the AMC Default Matching Algorithms + Cosine Similarity Method

now aligned to the blank header. This shows that our approach allows performing schema matching on columns with no headers.

For simplicity reason we have used the default combination algorithm for AMC which is an average of the applied algorithms (AMC's native and Cosine). We should also note that we have configured AMC's matchers to identify a "SIMILARTY_UNKOWN" value for columns that could not be matched successfully, which will allow other matchers to perform better. For example, our semantic matchers will skip columns that do not convey semantic meaning thus not affecting the score of other matchers. Moreover, the relatively high similarity score of "tr_dst" column is explained by the fact that the native AMC matching algorithm has skipped that column as it does not have a valid header, and the results are solely those of the Cosine matcher. Likewise, the Cosine matcher skips checking the "Cost" columns as they contain numeric values, and the implemented numerical matchers with the AMC's native matcher results are taken into account. Our numerical matchers' implementation gives a perfect similarity score for columns that are identified as date or money or IDs. However, this can be improved in the future as we can have different date hierarchy and numbers as IDs can present different entities. Combining this approach with the semantic and string matchers was found to yield good matching results.

The (PPMCC) Similarity algorithm combined with the AMC default matchers produces the results shown in Table 7.7.

We notice that by plugging the Spearman method, the number of matches and

Source Column	Target Column	Similarity Score
Reason for Trip	Reason for Trip	1
tr_dst		0.97351624
Begins On	Trip Begins On	0.833334
Ends On	Trip Ends On	0.8
Total	Total Cost	0.7333335
Trip	Trip Destination	0.7321428
Amount	Receipt Amount	0.7142857
Curr.	Currency	0.7041873
Crcy	Currency	0.6931407
Pd by Comp	Paid by Company	0.6904762
Period	Period Number	0.6666667
Trip	Trip Number	0.6666667
Pers.No.	Sequential no.	0.5555556
M/Km	Total Miles/Km	0.55

Table 7.7: Similarity Scores Using the AMC Default Matching Algorithms+ the PPMCC Similarity Method

similarity results have decreased (-4). After Several experiments we have found that this method does not work well with noisy data sets. For instance, the similarity results returned by Cosine, Pearson's and Spearman's matchers for the {tr_dst, empty header} pair is much higher: 95%, 97% and 43% respectively.

To properly measure the impact of each algorithm, we have tested the three algorithms (Cosine, PPMCC and Spearman) alone by de-activating the AMC's default matchers on the above data set. We have noticed that generally, the Cosine and PPMCC matchers perform well, resulting in more matching and better similarity score. However, the Spearman method was successful in finding more matches but with a lower similarity score than the others.

To better evaluate the three algorithms, we have tested them on four different data sets extracted from the Travel Expense Manager and Event Tracker systems. We ensured that the different experiments will cover all the cases needed to properly evaluate the matcher dealing with all the problems mentioned earlier.

We have found that generally the Cosine method is the best performing algorithm compared to the other two especially when dealing with noisy data sets. This was noticed particularly in our fourth experiment as the Cosine algorithm performed around 20% better than the other two methods. After investigating the dataset, we have found that several columns contained noisy and unrelated data. For example, in a "City" column, we had values such as "reference book" or "NOT_KNOWN".

To gain better similarity results we decided to combine several matching algorithms together. By doing so, we would benefit from the power of the AMC's string

Source Column	Target Column	Similarity Score
Reason for Trip	Reason for Trip	1
Begins On	Trip Begins On	0.8333334
Ends On	Trip Ends On	0.8
Total	Total Cost	0.7333335
Amount	Receipt Amount	0.7142857
Pd by Comp	Paid by Company	0.6904762
Currency2	Curr.	0.6689202
Trip	Trip Number	0.6666667
Pers.No.	Sequential no.	0.5555556
M/Km	Total Miles/Km	0.55

Table 7.8: Similarity Scores Using the AMC Default Matching Algorithms + Spearman Similarity Method

matchers that will work on column headers and our numeral and semantic matchers.

The Cosine and PPMCC Similarity algorithms combined with the AMC default matchers produces the results shown in Table 7.8.

Source Column	Target Column	Similarity Score
Reason for Trip	Reason for Trip	1
tr_dst		0.96351624
Curr.	Currency	0.79221311
Crcy	Currency	0.78173274
Begins On	Trip Begins On	0.77777785
Ends On	Trip Ends On	0.76666665
Amount	Receipt Amount	0.7380952
Total	Total Cost	0.7333335
Trip Country/Group	Ctr2	0.7194848
Pd by Comp	Paid by Company	0.6904762
Period	Period Number	0.6666667
Trip	Trip Number	0.6666667
Pers.No.	Sequential no.	0.5555556
M/Km	Total Miles/Km	0.55

Table 7.9: Similarity Scores Using the Combination of Cosine, PPMCC and AMC's defaults

The combination of the above mentioned algorithms have enhanced generally the similarity scores for the group. Moreover, we notice that the column "Trip Country/Group" was matched with "Ctr2". This match was not computed singularly by any of the previous algorithms. However, we notice that the match {Trip, Trip Destination} is now missing, probably as the similarity score is below the defined threshold.

Now, we will try and group all the mentioned algorithms. The combination of all Similarity algorithms with the AMC default matchers produces the results shown in Table 7.9.

Source Column	Target Column	Similarity Score
Reason for Trip	Reason for Trip	1
tr_dst		0.8779132
Curr.	Currency	0.80033726
Crcy	Currency	0.79380125
Begins On	Trip Begins On	0.7708334
Trip Country/Group	Ctr2	0.767311
Ends On	Trip Ends On	0.7625
Amount	Receipt Amount	0.7410714
Total	Total Cost	0.7333335
Trip	Trip Destination	0.7321428
Pd by Comp	Paid by Company	0.6904762
Period	Period Number	0.6666667
Trip	Trip Number	0.6666667
Pers.No.	Sequential no.	0.5555556
M/Km	Total Miles/Km	0.55

We notice that now we have an increased number of matches (15 compared to 14 in the previous trials). The column {Trip, Trip Destination} is matched again and the newly previously matched column {Trip Country/Group, Ctr2} has a higher similarity score. We have found that combining matching algorithms resulted in higher number of matches. Several tuning methods can be applied in order to enhance the similarity score as well. Trying other combination algorithms instead of the naiive average will be an essential part of our future work.

7.3 Summary

In this chapter, we presented a framework enabling mashup of potentially noisy enterprise and external data. The implementation is based on Open Refine and uses a dedicated service built on top of SAP HANA to annotate data with rich types. As a result, the matching process of heterogeneous data sources is improved. Our preliminary evaluation shows that for datasets where mappings were relevant yet not proposed, our framework provides higher quality matching results. Additionally, the number of matches discovered is increased when Linked Data is used in most datasets. In addition, we have shown that it is possible to reveal what are the "important" properties of entities by reverse engineering the choices made by Google when creating knowledge graph panels and by comparing users preferences obtained

from a user survey. This is fundamentally different from the work in [133] where the authors created a generalizable approach to open up closed knowledge bases like Google's by means of crowd-sourcing the knowledge extraction task. We are aware that this knowledge is highly dynamic, the Google Knowledge Graph panel varies across geolocation and time.

Chapter 8

Semantic Social News Aggregation

8.1 Introduction

With the rapid advances of the Internet, social media become more and more intertwined with our daily lives. The ubiquitous nature of Web-enabled devices, especially mobile phones, enables users to participate and interact in many different forms like photo and video sharing platforms, forums, newsgroups, blogs, micro-blogs, bookmarking services, and location-based services. Social networks are not just gathering Internet users into groups of common interests, they are also helping people follow breaking news, contribute to online debates or learn from others. They are transforming Web usage in terms of users' initial entry point, search, browsing and purchasing behavior [47].

A common scenario that often happens while reading an interesting article, coming across a nice video or participating in a discussion in a forum is the growing interest to check related material around the information read. To do so, users might go to Twitter, Google+ or YouTube. They can try several times with several keywords to obtain the desired results. In the end, they might end up with several browser tabs opened and get distracted by the information overload from all these resources. The same happens in companies when business users are interested in information provided by corporate web applications like enterprise communities. In this chapter, we present SNARC, a semantic social news aggregator that leverages live rich data that social networks provide to build an interactive rich experience on the Internet. The service retrieves news related to the current page from popular platforms like Twitter, Google+, YouTube, Vimeo, Slideshare, StackExchange and the Web. As a possible front-end implementation, we have created a Google Chrome extension which enriches the user experience by augmenting related contextual information to entities on the page itself, as well as displaying related social news on a floating sidebar.

8.2 Underlying Mechanism

The back-end of SNARC consists of three major components: a document handler that creates a "Semantic Model" that represents any web resource, a query layer that is responsible for disseminating queries to the supported social services and a data parser which processes the search results, wraps them in a common social model and generates the desired output.

8.2.1 Document Handler

The main idea behind SNARC is to provide a uniform model for web entities, whether they are blog entries, multimedia objects or micro-posts. To do so, SNARC creates a "Semantic Model" containing all the annotations and meta-data needed to query and reconcile social results.

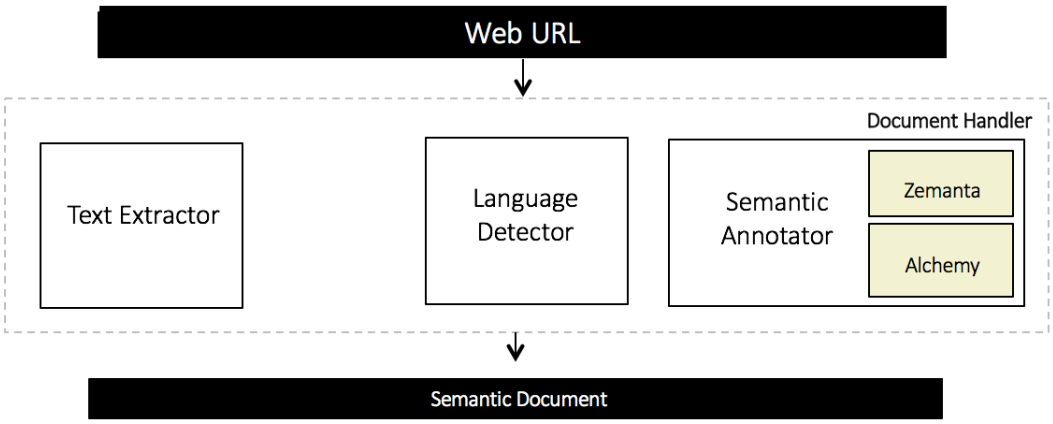

Figure 8.1: SNARC's Document Handler

The Semantic Model is created by the Document Handler (see 8.1) which receives a web page URL and performs these three main steps:

1. **Text Extraction:** Fetch the webpage that corresponds to the received URL and extract the textual content using a set of heuristics. These latter identify the main content of the page by stripping unwanted HTML tags and rank the different sections based on their semantics, class names and order. In the beginning we have used Alchemy API[1] to perform text extraction; but we have chosen to implement a simpler method ourselves which saved us an extra API call.

[1] http://www.alchemyapi.com

2. **Language Detection:** Detect the web page language using the Language Detection service of Alchemy API. This is necessary to match the desired language with compatible services like Twitter, YouTube, etc.

3. **Semantic Annotation:** Annotating the extracted text is the most important step in this process. We use Zemanta Suggest[2] and Alchemy API in order to extract:

   - **Tags:** These are the finest-grained queryable "keywords" that we use to retrieve the social results. From our experiments, combining tags results in better findings than using entities or concepts. However, we plan to evaluate the combination of keywords, entities and concepts in order to find the top-queryable terms that will retrieve the most relevant results on different abstraction levels.

     Tags retrieved from these services are ranked by confidence values calculated by their internal algorithms, these values are normalized for each service. According to our experiments we have found that Alchemy's Keywords Extraction API returns a large set of closely related keywords (i.e. Android, Android Phone, Android Tablet, ...). To construct a good query we therefore need to provide a certain level of abstraction.We perform a cleaning process on those keywords by applying the Levenshtein distance to rule out closely related keywords by disregarding those with lower confidences. We perform a similar process on the result of the union between the keywords returned by Alchemy and Zemanta to ensure a sparse keywords set.

   - **Semantic Entities:** Entities provide a higher abstraction level of the document. They are used to reconcile the social results in order to maintain relevancy with the document. Similar to the keywords extraction services, the entities retrieved are ranked and contain outbound links to the matched entity on dbpedia, Wikipedia, Freebase, etc. A union is made between the results from Alchemy and Zemanta to ensure a wider coverage of entities. When a match is found, we merge the links from the two sources to ensure that we include all the resources that can be used to augment extra information about that entity in the document.

   - **Categories:** These are high-level taxonomies that can generally describe the document's content. A taxonomy is used to narrow down our query scope when targeting services like YouTube. In our Semantic Document

[2] http://developer.zemanta.com/docs/suggest/

model we define two possible category sets, one retrieved from Alchemy's Text Categorization API[3] and the other retrieved from Zemanta Suggest API that follows the DMOZ categorization scheme[4].

At the end of this process, we will have constructed the needed elements (keywords, entities and high level categories) wrapped in our Semantic Model to be passed to the query generator. For example, a summary of the Semantic Model for a web page titled "Turkey protests: Erdogan in 'final' warning[5]" looks like:

1. **Categories:** Culture_Politics, Regional and Society
2. **Keywords:** Taksim Square, Protesters, Gezi Park, Mr Erdogan, Istanbul ...
3. **Entities:** Gezi Park, Recep Tayyip Erdogan, Taksim Square, Justice and Development Party (Turkey), Police of Turkey ...

8.2.2 Query Layer

In this component, the calls to the social services are made. SNARC uses the extracted keywords from the Semantic Document in order to construct the queries and disseminate them to the appropriate services. 8.2 shows the different steps in order to retrieve a set of social results.

1. **Query Builder:** Responsible for identifying targeted services and building tailored queries for each service. For example, if the processed document is categorized as a computer or technology related one, Stackoverflow service will be targeted with the queries constructed. However, other categories will correspond to different services from the Stack Exchange websites[6].
2. **Query Federator:** Responsible for federating the queries identified in the previous step to the corresponding services. To enhance performance, we tried to reduce the number of external calls. Yahoo Query Language (YQL)[7] helped us in minimizing the number of calls and batching them into a single one. It is an expressive SQL-like language that lets you query, filter, and join data across Web services. However, we have found that we cannot fully rely on YQL due to their API calls limit and the restriction on the query execution time that is set to 30 seconds. To overcome this, we have implemented a fallback mechanism

[3] http://www.alchemyapi.com/api/categ/categs.html
[4] http://www.dmoz.org/desc/Top
[5] http://www.bbc.co.uk/news/world-europe-22889060
[6] http://stackexchange.com/sites
[7] http://developer.yahoo.com/yql/

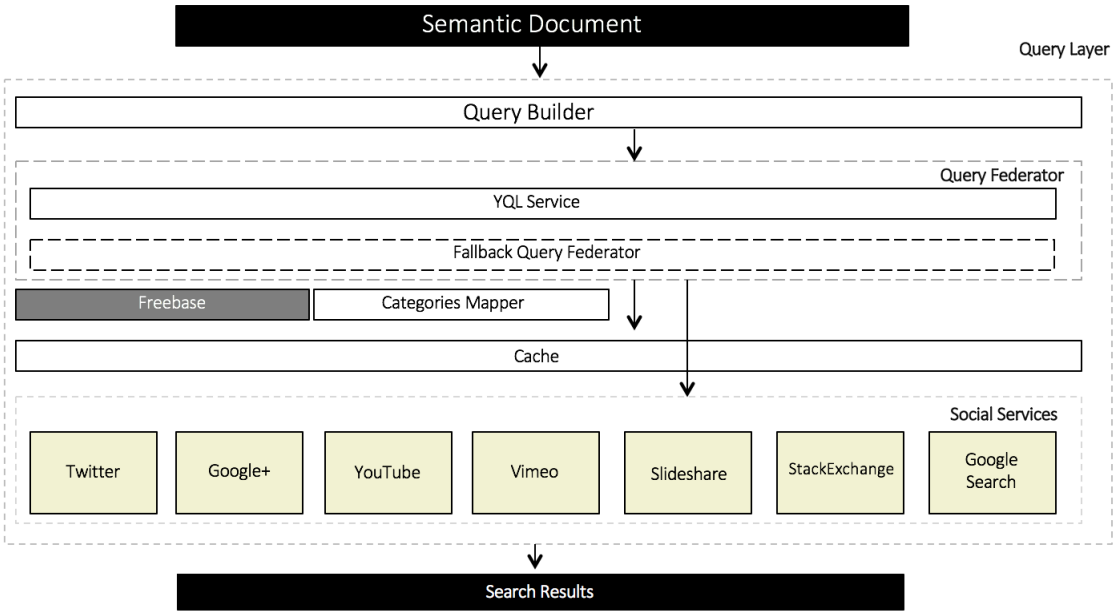

Figure 8.2: SNARC's Query Layer

that federates the queries to the selected social services and groups the result to be passed afterwards to the parser.

To further optimize the number of calls, we have decided to take the top two ranked keywords. We do not apply logical operator (AND/OR) in our queries; instead, we perform one-to-one mapping between each keyword and query. Indeed, we have found that gathering keywords even if semantically related might bring up noise in the results. However, as mentioned earlier, a part of the future work will be investigating the best method to construct the most relevant queryable entity using different logical operators.

3. **Caching:**The main setback in the query layer was the variable limited number of calls we can make to external APIs. To overcome this, we have implemented a simple cache mechanism that saves the results on disk up to an hour. There are several cache levels; the first is a URL level one where the results of the parsed queries are cached. For example, if a user visited a certain article on the CNN webpage the results might take up to 15 seconds to appear, whereas a second user visiting the same article minutes afterwards will have the cached results in few seconds. The second level is keyword and service specific. This can be very helpful as users generally browse articles of related topics or interests (semantic concepts), so for each user we can end up with the same high level concepts

being requested frequently. An important thing to note is that the caching is done on the server side and is disk-based.

The social services queried can be grouped as follows:

1. **Multimedia Services:** They include Slideshare, Vimeo and YouTube. Slideshare and YouTube allow the results to be fetched in a specific language that was detected in the previous step. In addition to that, YouTube search services are called twice; the first call is done to the YouTube V2 API[8] where we specify in addition to the keywords a high level category to be targeted. To do so, we have manually created a category mapping file that maps the high-levels categories of Alchemys API and DMOZ to those provided by YouTube. The second call is done to YouTube V3 API[9]. The new feature provided by Google in this version is the ability to search using a semantic concept that corresponds to a Freebase concept ID; it proves to retrieve better results that the normal search. Freebase concept calls are cached for longer periods as they are less prone to changes.

2. **Micro-posts Services:** They include Twitter, Google+ and Stackoverflow. Language filtering is done where applicable.

3. **General Search:** This includes similar results found via Google search or those retrieved from the Zemanta API call. They are general articles or blog posts related to the current active page.

8.2.3 Data Parser

This is the last step where the results and unified and wrapped in a single social model. 8.3 shows the different steps needed to produce the final parsed results that will be pushed back to the front-end.

1. **Live Reconciliator:** Social (or folksonomic) tagging has become a trending method to describe, search and discover content on the web. Folksonomies empower users by giving them total freedom in choosing their categories and keywords that they think describe best the content. This contrasts with taxonomies that over-impose hierarchical categorization of content [144]. However, in services like Twitter and Google+, tagging has been abused in a way that increased noise in the stream of results. To overcome this problem, we align the incoming stream of posts with the set of semantic concepts or keywords that describe the document. There are several approaches and tools

[8] https://developers.google.com/youtube/2.0/
[9] https://developers.google.com/youtube/v3/

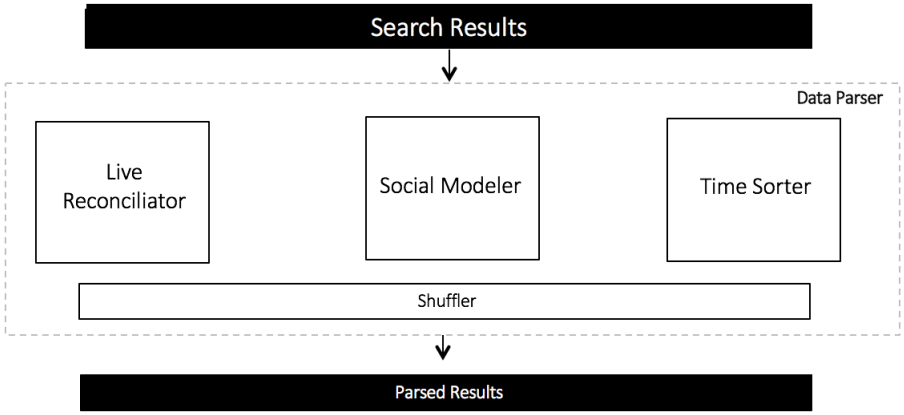

Figure 8.3: SNARC's Data Parser

like [76, 46, 122, 144] that aim at solving this problem. In SNARC we rely on two levels of reconciliation: one uses the high-level taxonomy (categories); and the other uses the vector of entities defined in the Semantic Document. For example, if SNARC wants to reconcile a blog post result retrieved from a general search, it constructs a Semantic Document Model for that result and applies the Cosine Similarity on the vector of ranked entities for each Semantic Model. Currently, we only reconcile against blog posts as it is very straightforward to construct a Semantic Document Model for them. However, an integral part of the future work will be the integration of SNARC's model to micro-posts and video search services.

2. **Social Modeler:** Every social network has its own underlying data model. To overcome this problem, we need to present the social results in a common wrapper. To do so, we have created an optimized universal social model that contains all the necessary data to model social information and can be reused in other projects. The model contains service related attributes like the service name and type, general post information like the author's name, profile link, image and geo-location information and post-specific information like the title, thumbnail, embed code, main content and link.

3. **Time Sorter and Results Shuffler:** To better display the results on the front-end, we unify the time representation and sort the results based on it. Afterwards we pick the top N results and shuffle them to generate a random order.

8.3 Front-End

SNARC is a service that generates a JSON file containing the results wrapped in our universal social model. As a possible front-end implementation, we have implemented a chrome extension that loads SNARC on any web page or application (see 8.4). This UI implementation offers more flexibility to users by loading related social news anytime on any webpage or application. The results are visualized using jQuery templates as a sliding panel on one of the screen edges, extracted entities are highlighted in the page itself and a short excerpt is displayed when hovering over them.

Figure 8.4: SNARC's User Interface - The Google Chrome Extension

8.4 Summary

Aggregating relevant social news is not an easy task. SNARC performs the task in a nice and intuitive way that allows the user to discover what is happening instantly and without the need to navigate away from the current page. One of the important things to consider for the future is the integration of better reconciliation features and tools to ensure the display of relevant social posts. Moreover, real-time feature that can also push new related posts would be a great addition.

Conclusion of Part II

In this part, we presented the various parts required to enable Data Integration in the enterprise.

First, we created an internal knowledge base by importing DBpedia into SAP HANA. On top of that, we built a set of services that enable entity disambiguation, semantic enrichment and schema matching. We presented RUBIX, a framework enabling mashup of potentially noisy enterprise and external data. The implementation is based on Open Refine and uses our entity disambiguation service to annotate data with rich types. As a result, the matching process of heterogeneous data sources is improved. We have also shown that it is possible to reveal what are the "important" properties of entities by reverse engineering the choices made by Google when creating knowledge graph panels and by comparing users preferences obtained from a user survey. Our motivation is to represent this choice explicitly, using the Fresnel vocabulary, so that any application could read this configuration file for deciding which properties of an entity is worth to visualize.

Last, we cover the aspect of integrating external data coming from social media outlets. Data nowadays is spread over heterogeneous silos of archived and live data. People willingly share data on social media by posting news, views, presentations, pictures and videos. We presented SNARC, 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.

Going back to our scenario, the proposed frameworks and services will allow our analyst **Bob** to be able to find and match various reports he is working on based on the new matching capabilities of SAP Lumira. Moreover, he will be able to augment extra measures and dimensions to his reports using DBpedia. In addition, **Bob** will be able to monitor relevant social feeds. This will allow him to either embed social snippets directly in this infographics and reports or discover new information sources.

Chapter 9

Conclusions and Future Perspectives

In this chapter, we summarize the major achievements of this thesis and we give an outlook on future perspectives.

9.1 Achievements

9.2 Perspectives

Appendix A

HDL - A Harmonized Dataset Model

```
{
    "id" : "Dataset unique identification",
    "name" : "Dataset machine-readable name",
    "title" : "Dataset human-readable name",
    "description" : "Human-readable description (e.g., an abstract)
        with sufficient detail to enable a user to quickly
       understand whether the dataset is of interest",
    "url" : "Dataset accessible URI",
    "type" : "Dataset type (private, open, etc.)",
    "accessLevel" : "The degree to which this dataset could be made
        publicly-available, regardless of whether it has been made
        available. ",
    "rights" : "This may include information regarding access or
       restrictions based on privacy, security, or other policies
       .",
    "state" : "",
    "language" : "[list of languages used in the dataset]",
    "languageCode" : "[list of language code used in the dataset]",
    "conformsTo" : "URI used to identify a standardized
       specification the dataset conforms to",
    "accuralPeriodicity" : "",
    "licenseTitle" : "The human-readable title of the license used"
       ,
    "licenseID" : "The machine-readable identification of the
       license used",
    "licenseUrl" : "The URL of the license used",
    "attributionText" : "",
    "attributionLink" : "",
    "version" : "",
    "revisionId" : "",
    "revisionTimestamp" : "",
    "metadataCreated" : "The creation date of the metadata",
```

```
"metadataModified" : "Most recent date on which the dataset
   metadata was changed, updated or modified",
"maintainer" : {
    "name" : "Contact person name for the dataset maintainer",
    "email" : "Contact person name for the dataset maintainer",
    "role" : "Contact person role in the organization for the
       dataset maintainer"
},
"author" : {
    "name" : "Contact person name for the dataset author",
    "email" : "Contact person email for the dataset author",
    "role" : "Contact person role in the organization for the
       dataset author"
},
"hasPart" : ["The collection of which the dataset is a subset"]
   ,
"ratingsAverage" : "Average score of the resource rating by
   users",
"dataQualityAverage" : "Average score of the objective data
   quality for the dataset",
"numberOfResources" : "",
"numberOfTags" : "",
"organization" : [
    {
        "id" : "Unique identification of the organization",
        "name" : "Human-readable organization name",
        "description" : "Human-readable description (e.g., an
           abstract) with sufficient detail to enable a user
           to have an overview of the organization",
        "type" : "",
        "organizationalUnit" : "An Organization such as a
           department or support unit which is part of some
           larger Organization and only has full recognition
           within the context of that Organization",
        "subOrganizationOf" : "Represents hierarchical
           containment of Organizations or OrganizationalUnits
           ; indicates an organization which is a sub-part or
           child of this organization",
        "basedAt" : "Indicates the site at which a person is
           based. We do not restrict the possibility that a
           person is based at multiple sites",
        "hasSite" : {
```

```
                "siteAddress" : "human-readable address for the
                    company's site"
            },
        "hasUnit" : "Indicates a unit which is part of this
            Organization",
        "location" : "Location description for the organization
            "
    }
],
"resources" : [
    {
        "id" : "Unique identification of the resource",
        "name" : "Resource machine-readable name",
        "title" : "Resource human-readable name",
        "description" : "Human-readable description (e.g., an
            abstract) with sufficient detail to enable a user
            to quickly understand whether the resource is of
            interest",
        "type" : "The human-readable format of the resource",
        "downloadUrl" : "URL providing direct access to a
            dataset, for example via API or a graphical
            interface",
        "accessUrl" : "URL providing indirect access to a
            dataset, for example via API or a graphical
            interface",
        "format" : "A human-readable description of the file
            format of a distribution",
        "hash" : "Calculated unique hash ID",
        "state" : "",
        "accessLevel" : "The degree to which this resource
            could be made publicly-available",
        "mimetype" : "Machine-readable file format",
        "size" : "Actual size of the resource in bytes",
        "describedBy" : "URL to the data dictionary for the
            distribution found at the downloadUrl",
        "describedByType" : "The machine-readable file format",
        "conformsTo" : "URI used to identify a standardized
            specification the distribution conforms to",
        "rating" : "Normalized score of the resource rating by
            users",
        "dataQuality" : "Score of the resource objective
            quality",
        "cacheUrl" : "A URL of the resource cache",
```

```
            "temporal" : {
                "temporalGranularity" : "The range of temporal
                   granularity of a dataset",
                "temporalCoverageFrom" : "Start date of
                   applicability for the data",
                "temporalCoverageTo" : "End date of applicability
                   for the data"
            },
            "spatial" : {
                "spatialText" : "",
                "spatialGranularity" : "",
                "bbox" : "",
                "layers" : "",
                "namespace" : ""
            },
            "created" : "The creation date of the resource",
            "modified" : "Most recent date on which the resource
               was changed, updated or modified",
            "cacheModified" : "Most recent date on which the
               resource cache was changed, updated or modified",
            "revisionnId" : "",
            "revisionTimestamp" : ""
        }
    ],
    "groups" : [
        {
            "id" : "Group unique identification",
            "name" : "Group machine-readable name",
            "title" : "Group human-readable name",
            "description" : "Human-readable description (e.g., an
               abstract) with sufficient detail to enable a user
               to quickly understand whether the group is of
               interest",
            "created" : "The creation date of the group",
            "modified" : "Most recent date on which the group was
               changed, updated or modified",
            "administrator" : [
                {
                    "name" : "Contact person name for the group
                       administrator",
                    "email" : "Contact person name for the group
                       administrator",
```

```
                    "role" : "Contact person name for the group
                        administrator"
                }
            ],
            "subGroupOf" : "Defines if the group is part of a
                parent group and specifies the parent group id"
        }
    ],
    "tags" : [
        {
            "id" : "Unique identification of the tag",
            "name" : "Tag machine-readable name",
            "title" : "Tag human-readable name",
            "vocabularyId" : "Unique identifier of the tag
                vocabulary if any",
            "created" : "The creation date of the tag",
            "modified" : "Most recent date on which the tag was
                changed, updated or modified"
        }
    ]
}
```

Listing A.1: The suggested harmonized dataset model (HDL)

Appendix B

Installation and cutomization instructions

B.1 Installation and cutomization instructions for Roomba

You can either download Roomba from the Github repo[1] as a zip file or clone directly through git. Pay attention when cloning as there is a submodule defined and has to be cloned recursively as well. This can be done via:

```
git clone --recursive http://github.com/ahmadassaf/opendata-checker
```

If you have cloned without `--recursive`, you may find out that some folders are empty. To fix this:

```
git submodule update --init
```

After successfully having cloned Roomba to your local machine.

```
1. Install dependencies by running the command "npm install"
2. Start Roomba by running "node DC.js"
```

B.2 Customizing Roomba

There are a set of options that you can customize. They can be edited from `options.json` file (see listing B.1)

```
{
    "locale"         : "en",
    "cacheFolderName": "/cache/",
    "licensesFolder" : "/util/licenses/",
    "mappingFileName": "licenseMappings",
    "proxy"          : ""
}
```

Listing B.1: Roomba's customization via the options file

[1] https://github.com/ahmadassaf/opendata-checker

- **locale**: the language of the messages and the prompts. Default: **en**
- **cacheFolderName**: The name of cache folder separated by /. Default: **/cache/**
- **licensesFolder**: The location of the Open Licenses Repo[2]. It is defined as a submodule and by default it is located in `/util/licenses/`
- **mappingFileName**: The name of manual license mappings. Default: **licenseMappings**
- **proxy**: Proxy server e.g. `proxy:8080`. Default: ""

B.2.1 Localization

If you wish to translate the messages and prompts into other languages than English. You have to create a new language entry in the `util/messages.js` with the new locale code e.g. `fr`. Afterwards, you should keep the object keys intact by translate the values into the desired language. For example:

```
var messages =   {
    "en": {
        "error" : {
            "unKnownError" : "An unknown Error occurred .. ",
            .
            .
    "fr": {
        "error" : {
            "unkownError" : "Erreur inconnue .."
        }
    }
```

Listing B.2: Roomba's localization file example

B.3 Installation and cutomization instructions for Knowledge Graph Scraper

You can either download the knowledge graph scrapper from the Github repo[3] as a zip file or clone directly through git.

```
git clone http://github.com/ahmadassaf/kbe
```

[2] https://github.com/okfn/licenses
[3] https://github.com/ahmadassaf/kbe

After successfully having cloned the repository to your local machine.

```
1. Install dependencies by running the command "npm install"
2. Start Roomba by running "node KBE.js"
```

The script will automatically create all the required Cache folders:

- Main cache folder called **cache** in the root folder of the application
- folder called **GKB** inside the cache folder: This will hold the aggregated Google Knowledge boxes extracted for a DBpedia concept (type)
- folder called **instances_GKB** inside the cache folder: This will hold the Google Knowledge box for a single instance
- folder called **instances** inside the cache folder: This will hold the DBpedia instances for each concept (type)
- folder called **instance_properties** inside the cache folder: This ill hold the distinct list of properties for all the instances of a certain concept

The application is run in the console and the output will be available in texttt/-cache/result.json.

B.3.1 Customization

There are a set of options that you can customize. They can be edited from `options.json` file (see listing B.3)

```
{
        cache_dbpedia_concepts       : true,
        limit_dbpedia_concepts       : true,
        limit_dbpedia_instances      : true,
        limit_dbpedia_concepts_value : 10,
        limit_dbpedia_instances_value: 10,
        proxy  : null
}
```

Listing B.3: Roomba's localization file example

- **cache_dbpedia_concepts**: Caches the concepts retrieved from DBpedia.
- **limit_dbpedia_concepts**: Limits the number of concepts retrieved by DBpedia, false will retrieve all the concepts

- **limit_dbpedia_instances**: Limits the number of instances retrieved for each concept, false will retrieve all the instances
- **limit_dbpedia_concepts_value**: Specifies the number of concepts you wish to retrieve
- **limit_dbpedia_instances_value**: Specifies the number of instances you wish to retrieve for each concept
- **proxy**: Specifies the proxy address string containing ports e.g. `proxy:8080`. Default: ””

Since Google changes the CSS selectors dynamically at random times, the user can always check the corresponding CSS class name selectors for the Google Knowledge Panel and edit them if needed in the same `options.json` file as shown in listing B.4.

```
{
        "knowledgeBox"                : "#kno-result",
        "knowledgeBox_disambiguate"   : ".kp-blk",
        "property"                    : "._Nl",
        "property_value"              : ".kno-fv",
        "label"                       : ".kno-ecr-pt",
        "description"                 : ".kno-rdesc",
        "type"                        : "._kx",
        "images"                      : ".bicc",
        "special_property"            : ".kno-sh",
        "special_property_value"      : "._Zh",
        "special_property_value_link" : "a._dt"
}
```

Listing B.4: Roomba's localization file example

Appendix C

Roomba Results

C.1 Roomba profiling report for the LOD Cloud

Metadata Report

```
[259] group information is missing. Check organization information as they can
    be mixed sometimes
[133] maintainer field exists but there is no value defined
[143] maintainer_email field exists but there is no value defined
[6] author field exists but there is no value defined
[39] author_email field exists but there is no value defined
[156] version field exists but there is no value defined
[44] The url defined for this dataset is not reachable
[28] organization_image_url field exists but there is no value defined
[34] notes field exists but there is no value defined
[1] Tags information [Tags, Vocabularies] is missing
[224] resources information (API endpoints, downloadable dumpds, etc.) is
    missing
[11] author_email is not a valid e-mail address
[6] maintainer_email is not a valid e-mail address
[3] organization_description field exists but there is no value defined
[3] url field exists but there is no value defined
[2] The organization image_url defined for this dataset is not reachable
```

Dataset Statistics

```
There is a total of: 259 [missing] group fields  100.00%
There is one [missing] tag field  0.39%
There is a total of: 224 [missing] resources fields  86.49%
There is a total of: 133 [undefined] maintainer fields  51.35%
There is a total of: 143 [undefined] maintainer_email fields  55.21%
There is a total of: 6 [undefined] author fields  2.32%
There is a total of: 39 [undefined] author_email fields  15.06%
There is a total of: 156 [undefined] version fields  60.23%
There is a total of: 28 [undefined] organization_image_url fields  10.81%
There is a total of: 34 [undefined] notes fields  13.13%
There is a total of: 3 [undefined] organization_description fields  1.16%
There is a total of: 3 [undefined] url fields  1.16%
```

```
                    Dataset Connectivity Issues
--------------------------------------------------------------------------------

There are 44 connectivity issues with the following URLs:
   - http://id.loc.gov/authorities/
   - http://libris.kb.se
   - http://www.london-gazette.co.uk/mashup/gazettesdata.htm
   - http://www4.wiwiss.fu-berlin.de/eures/
   - http://www.linkedopenservices.org/services/geo/SpatialResources/point/
     ICAO/
--------------------------------------------------------------------------------

                            Tag Report
--------------------------------------------------------------------------------

[3220] vocabulary_id field exists but there is no value defined
--------------------------------------------------------------------------------

                          Tag Statistics
--------------------------------------------------------------------------------

There is a total of: 3220 [undefined] vocabulary_id fields  100.00%
Total elements count: 3220
--------------------------------------------------------------------------------

                          License Report
--------------------------------------------------------------------------------

[142] license_url field is missing
[136] We could not normalize the license information as no valid mapping was
    found
[123] License information has been normalized
[43] license_title field exists but there is no value defined
[43] license_id field exists but there is no value defined
--------------------------------------------------------------------------------

                        License Statistics
--------------------------------------------------------------------------------

There is a total of: 141 [missing] license_url fields  54.44%
There is a total of: 43 [undefined] license_title fields  16.60%
There is a total of: 43 [undefined] license_id fields  16.60%
Total elements count: 259
--------------------------------------------------------------------------------

                          Resource Report
--------------------------------------------------------------------------------

[1035] cache_last_updated field exists but there is no value defined
[1024] webstore_last_updated field exists but there is no value defined
[871] size field exists but there is no value defined
[1020] hash field exists but there is no value defined
[69] format field exists but there is no value defined
[1024] mimetype_inner field exists but there is no value defined
```

```
[1035] url-type field is missing
[832] mimetype field exists but there is no value defined
[1035] cache_url field exists but there is no value defined
[817] name field exists but there is no value defined
[928] created field is missing
[975] webstore_url field exists but there is no value defined
[853] last_modified field exists but there is no value defined
[488] resource_type field exists but there is no value defined
[335] The url for this resource is not reachable
[98] description field exists but there is no value defined
[52] The size for resource is not defined correctly
[53] mimetype value defined where the resource is not reachable
[44] size value defined where the resource is not reachable
[20] The mimeType for resource is not defined correctly
```

Resource Statistics

```
There is a total of: 1035 [undefined] cache_last_updated fields  96.91%
There is a total of: 1024 [undefined] webstore_last_updated fields  95.88%
There is a total of: 871 [undefined] size fields  81.55%
There is a total of: 1020 [undefined] hash fields  95.51%
There is a total of: 69 [undefined] format fields  6.46%
There is a total of: 1024 [undefined] mimetype_inner fields  95.88%
There is a total of: 832 [undefined] mimetype fields  77.90%
There is a total of: 1035 [undefined] cache_url fields  96.91%
There is a total of: 817 [undefined] name fields  76.50%
There is a total of: 975 [undefined] webstore_url fields  91.29%
There is a total of: 853 [undefined] last_modified fields  79.87%
There is a total of: 488 [undefined] resource_type fields  45.69%
There is a total of: 98 [undefined] description fields  9.18%
There is a total of: 1034 [missing] url-type fields  96.82%
There is a total of: 927 [missing] created fields  86.80%
Total elements count: 1068
```

Resource Connectivity Issues

There are 330 connectivity issues with the following URLs:

- http://www.data.gov/semantic/data/alpha
- http://www.data.gov/catalog/raw
- http://www.data.gov/catalog/geodata
- http://lab3.libris.kb.se/sparql
- http://www.london-gazette.co.uk/mashup/LondonGazetteData.zip
- http://www.london-gazette.co.uk/issues/59535/notices/1196300
- http://www.idref.fr/027182800/id

.....

Un-Reachable URLs Types

```
There are: 212 unreachable URLs of type [null]
There are: 8 unreachable URLs of type [api]
There are: 112 unreachable URLs of type [file]
There are: 1 unreachable URLs of type [metadata]
There are: 1 unreachable URLs of type [example]
There are: 1 unreachable URLs of type [documentation]
```

Listing C.1: The result of running Roomba on the LOD Cloud group. Note that some URLs are cut for display purposes

C.2 Roomba quality profiling report for the LOD Cloud

```
-------------------------------------------------------------------------------

                              Dataset Quality Report

-------------------------------------------------------------------------------

completeness quality Score      :   50.22%
availability quality Score      :   23.32%
licensing quality Score         :   21.14%
freshness quality Score         :   79.49%
correctness quality Score       :   72.36%
comprehensibility quality Score:    31.63%
provenance quality Score        :   74.07%
Average total quality Score     :   50.32%

-------------------------------------------------------------------------------

                        Quality Indicators Average Error %

-------------------------------------------------------------------------------

Supports multiple serializations                                    :   11.35%
Has different data access points                                    :   19.31%
Uses datasets description vocabularies                              :   88.80%
Existence of descriptions about its size                            :   86.30%
Existence of descriptions about its structure (MIME Type, Format) :   83.67%
Existence of descriptions about its organization and categorization:    8.33%
Existence of dereferencable links for the dataset and its resources:    0.45%
Existence of an RDF dump that can be downloaded by users            :   91.51%
Existence of queryable endpoints that respond to direct queries     :   96.14%
Existence of valid dereferencable URLs (respond to HTTP request) :   42.41%
Existence of human and machine readable license information         :   61.58%
Existence of dereferencable links to the full license information  :   96.14%
Existence of timestamps that can keep track of its modifications    :   20.51%
Includes the correct MIME type for the content                      :   48.92%
Includes the correct size for the content                           :   32.71%
Absence of Syntactic errors on the links level                      :   1.28%
Existence of at least one exemplary RDF file                        :   99.61%
Existence of general information (title, URL, description)          :   5.89%
Existence of mailing list, message board or point of contact        :   99.61%
Existence of metadata that describes its authoritative information :   21.75%
Usage of versioning                                                 :   30.12%
```

Listing C.2: The result of running Roomba quality profiler on the LOD Cloud group

Appendix D

DBpedia ranked properties in Fresnel vocabulary

```
<rdf:RDF xml:base="http://dbpedia.org/ontology/"
    xmlns="http://dbpedia.org/ontology/"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
   <owl:Ontology rdf:about="">
      <owl:versionInfo xml:lang="en">Version 3.8</owl:versionInfo>
   </owl:Ontology>
   <owl:Class rdf:about="http://dbpedia.org/ontology/BasketballLeague">
      <rdfs:label xml:lang="el">
                                 </rdfs:label>
      <rdfs:label xml:lang="fr">ligue de basketball</rdfs:label>
      <rdfs:label xml:lang="en">basketball league</rdfs:label>
      <rdfs:label xml:lang="it">lega di pallacanestro</rdfs:label>
      <rdfs:label xml:lang="ja">         ł         </rdfs:label>
      <rdfs:comment xml:lang="en">a group of sports teams that compete
          against each other in Basketball</rdfs:comment>
      <rdfs:subClassOf
          rdf:resource="http://dbpedia.org/ontology/SportsLeague"/>
   </owl:Class>
   <owl:Class rdf:about="http://dbpedia.org/ontology/LunarCrater">
      <rdfs:label xml:lang="en">lunar crater</rdfs:label>
      <rdfs:label xml:lang="fr">cratère lunaire</rdfs:label>
      <rdfs:label xml:lang="el">                        </rdfs:label>
      <rdfs:label xml:lang="nl">maankrater</rdfs:label>
      <rdfs:subClassOf
          rdf:resource="http://dbpedia.org/ontology/NaturalPlace"/>
   </owl:Class>
   <owl:Class rdf:about="http://dbpedia.org/ontology/MotorsportSeason">
      <rdfs:label xml:lang="en">motorsport season</rdfs:label>
      <rdfs:subClassOf
          rdf:resource="http://dbpedia.org/ontology/SportsSeason"/>
   </owl:Class>
   <owl:Class rdf:about="http://dbpedia.org/ontology/MilitaryPerson">
```

```
    <rdfs:label xml:lang="el">           </rdfs:label>
    <rdfs:label xml:lang="fr">militaire</rdfs:label>
    <rdfs:label xml:lang="en">military person</rdfs:label>
    <rdfs:label xml:lang="it">militare</rdfs:label>
    <rdfs:label xml:lang="nl">militair</rdfs:label>
    <rdfs:label xml:lang="ko"></rdfs:label>
    <rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/Person"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/TimePeriod">
    <rdfs:label xml:lang="el">                </rdfs:label>
    <rdfs:label xml:lang="fr">période temporelle</rdfs:label>
    <rdfs:label xml:lang="en">time period</rdfs:label>
    <rdfs:label xml:lang="nl">tijdvak</rdfs:label>
    <rdfs:label xml:lang="es">periodo temporal</rdfs:label>
    <rdfs:subClassOf
        rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <owl:disjointWith
        rdf:resource="http://dbpedia.org/ontology/Person"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/AutomobileEngine">
    <rdfs:label xml:lang="el">                          </rdfs:label>
    <rdfs:label xml:lang="fr">moteur d'automobile</rdfs:label>
    <rdfs:label xml:lang="en">automobile engine</rdfs:label>
    <rdfs:label xml:lang="it">motore d'automobile</rdfs:label>
    <rdfs:label xml:lang="ja">   </rdfs:label>
    <rdfs:label xml:lang="pt">motor de automvel</rdfs:label>
    <rdfs:label xml:lang="de">Fahrzeugmotor</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/Device"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/Enzyme">
    <rdfs:label xml:lang="el">   </rdfs:label>
    <rdfs:label xml:lang="en">enzyme</rdfs:label>
    <rdfs:label xml:lang="it">enzima</rdfs:label>
    <rdfs:label xml:lang="ja">ł</rdfs:label>
    <rdfs:label xml:lang="nl">enzym</rdfs:label>
    <rdfs:label xml:lang="de">enzym</rdfs:label>
    <rdfs:subClassOf
        rdf:resource="http://dbpedia.org/ontology/Biomolecule"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/TelevisionShow">
    <rdfs:label xml:lang="el">                </rdfs:label>
    <rdfs:label xml:lang="fr">émission de télévision</rdfs:label>
    <rdfs:label xml:lang="en">television show</rdfs:label>
    <rdfs:label xml:lang="ja">  ł  </rdfs:label>
    <rdfs:label xml:lang="sl">televizijska oddaja</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/Work"/>
</owl:Class>
```

```
<owl:Class rdf:about="http://dbpedia.org/ontology/LaunchPad">
   <rdfs:label xml:lang="en">launch pad</rdfs:label>
   <rdfs:label xml:lang="el">                </rdfs:label>
   <rdfs:label xml:lang="fr">rampe de lancement</rdfs:label>
   <rdfs:subClassOf
      rdf:resource="http://dbpedia.org/ontology/Infrastructure"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/CyclingLeague">
   <rdfs:label xml:lang="en">cycling league</rdfs:label>
   <rdfs:label xml:lang="el">                          </rdfs:label>
   <rdfs:label xml:lang="fr">ligue de cyclisme</rdfs:label>
   <rdfs:comment xml:lang="en">a group of sports teams that compete
      against each other in Cycling</rdfs:comment>
   <rdfs:subClassOf
      rdf:resource="http://dbpedia.org/ontology/SportsLeague"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/Territory">
   <rdfs:label xml:lang="en">territory</rdfs:label>
   <rdfs:subClassOf
      rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/MusicFestival">
   <rdfs:label xml:lang="el">                    </rdfs:label>
   <rdfs:label xml:lang="fr">festival de musique</rdfs:label>
   <rdfs:label xml:lang="en">music festival</rdfs:label>
   <rdfs:label xml:lang="nl">muziekfestival</rdfs:label>
   <rdfs:label xml:lang="ko"></rdfs:label>
   <rdfs:label xml:lang="es">festival de msica</rdfs:label>
   <rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/Event"/>
   <rdfs:subClassOf rdf:resource="http://schema.org/Festival"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/Tax">
   <rdfs:label xml:lang="el"> </rdfs:label>
   <rdfs:label xml:lang="fr">taxe</rdfs:label>
   <rdfs:label xml:lang="en">tax</rdfs:label>
   <rdfs:label xml:lang="ja"></rdfs:label>
   <rdfs:label xml:lang="nl">belasting</rdfs:label>
   <rdfs:label xml:lang="es">impuesto</rdfs:label>
   <rdfs:label xml:lang="de">Steuer</rdfs:label>
   <rdfs:subClassOf
      rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:about="http://dbpedia.org/ontology/IceHockeyPlayer">
   <rdfs:label xml:lang="en">ice hockey player</rdfs:label>
   <rdfs:label xml:lang="el">              </rdfs:label>
   <rdfs:label xml:lang="fr">joueur de hockey sur glace</rdfs:label>
   <rdfs:label xml:lang="nl">ijshockeyspeler</rdfs:label>
```

```
      <rdfs:subClassOf
         rdf:resource="http://dbpedia.org/ontology/Athlete"/>
   </owl:Class>
   <owl:Class rdf:about="http://dbpedia.org/ontology/BloodType">
      <rdfs:label xml:lang="el">            </rdfs:label>
      <rdfs:label xml:lang="en">academic journal</rdfs:label>
      <rdfs:label xml:lang="ja"></rdfs:label>
      <rdfs:label xml:lang="nl">bloedgroep</rdfs:label>
      <rdfs:label xml:lang="pt">tipo sanguneo</rdfs:label>
      <rdfs:label xml:lang="de">Blutgruppe</rdfs:label>
      <rdfs:subClassOf
         rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
   </owl:Class>
   <owl:Class rdf:about="http://dbpedia.org/ontology/FootballMatch">
      <rdfs:label xml:lang="en">football match</rdfs:label>
      <rdfs:label xml:lang="el">                    </rdfs:label>
      <rdfs:label xml:lang="pl">mecz piki nonej</rdfs:label>
      <rdfs:label xml:lang="es">partido de ftbol</rdfs:label>
      <rdfs:comment xml:lang="en">a competition between two football
         teams</rdfs:comment>
      <rdfs:subClassOf
         rdf:resource="http://dbpedia.org/ontology/SportsEvent"/>
   </owl:Class>
   <owl:Class rdf:about="http://dbpedia.org/ontology/MilitaryConflict">
      <rdfs:label xml:lang="en">military conflict</rdfs:label>
      <rdfs:label xml:lang="el">                        </rdfs:label>
      <rdfs:label xml:lang="fr">conflit militaire</rdfs:label>
      <rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/Event"/>
   </owl:Class>
   <owl:Class rdf:about="http://dbpedia.org/ontology/FilmFestival">
      <rdfs:label xml:lang="el">                            </rdfs:label>
      <rdfs:label xml:lang="fr">festival du film</rdfs:label>
      <rdfs:label xml:lang="en">film festival</rdfs:label>
      <rdfs:label xml:lang="ja"></rdfs:label>
      <rdfs:label xml:lang="nl">filmfestival</rdfs:label>
      <rdfs:label xml:lang="ko"></rdfs:label>
      <rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/Event"/>
      <rdfs:subClassOf rdf:resource="http://schema.org/Festival"/>
   </owl:Class>
</rdf:RDF>
```

Listing D.1: An excerpt of the Fresnel vocabulary for top properties mappings of DBpedia 3.9

Appendix E

SAP HANA Semantic Services API

E.1 XSJS API Implementation

There are four different `XSJS` files in this project:

- `disambiguate.xsjs`: Gets most appropriate entities / categories for a searchword
- `describe.xsjs`: Loads the properties which are available for an entity
- `enrich.xsjs`: Retrieves the data, filtering is possible
- `tableMatch.xsjs`: Matches two tables by sending their cell values

The `disambiguate.xsjs` and `tableMatch.xsjs` use an XSJS lib called `DbpediaLib.xsjslib`. You can import it with this code:

```
$.import("harubixMatch.WebContent","DBpediaLib");
```

This library contains the core functionalities of our services. You can use the functions in this library to build your own API extension. Just add another XSJS file, import with the command above, and you can use one of those functions:

```
$.harubixMatch.WebContent.DbpediaLib.queryEntitiesWithTypes(query,fuzzyStr, limit)
```

This will query for a searchword in the **TYPES** table. It returns a JSON object. More specific, it will return an array containing all found entities (called entities). For each element of entities the text search score (`txtScore`) and the number of incoming associations (`incomingNo`) is returned, as well as an array of the associated types (`types`). Each element in types has a name and an order. Each element of entities also has a `finalScore`, but this is set to zero and will be calculated in the method getEntitiesWithTypes

Generally, you should not use this method at all, as it is a helper method for `getEntitiesWithTypes`

```
$.harubixMatch.WebContent.DbpediaLib.getEntitiesWithTypes(query, limit, incomingNoWeight,
fuzzy, multilang)
```

This method can be used externally. First of all it processes the fuzzy parameter (which should be double) into a string. Here `textSearch=compare` is added for all queries except for the **INTERLANGUAGE** table. This is done to achieve similar text search scores for both search with and without fuzzy. The **INTERLANGUAGE** table currently does not support `textSearch=compare` because of its column type (has to be `SHORTTEXT` or `TEXT` for that)

If `multilang` is set to true, the **INTERLANGUAGE** table is searched (fuzzy, if specified). The top rated result is then used for further processing (replaces the query parameter).

Then the `queryEntitiesWithTypes` method is called. If it returns no results, the **RAWPROPERTIES** are searched (example: AAPL). The top rated result will replace query parameter and once again `queryEntitiesWithTypes` is called.

Afterwards, the biggest number of incoming associations is determined from the returned `ratingArray`. The finalScore is calculated and saved, the ratingArray is sorted by that finalScore. The finalScore is currently calculated like this:

```
$.harubixMatch.WebContent.DBpediaLib.getEntityTypeWithContext(query,context,limit)
```

This returns one entity which fits most likely in the context given (JSON object). For each entity in the context the method `getCategories` is called. From this collection of types, a type vector is created a list of types, and if a type occurs more than once, its score is added to the existing entry. Then the list is sorted by score.

Then the possible entities for the query are retrieved by calling `getEntityWithTypes`. Now the types of the entities are compared (helper method `compareQueryWithContext`) to the created type vector, starting with the most likely entity and the highest rated score. As soon as there is a match, the corresponding entity/type is returned.

```
$.harubixMatch.WebContent.DBpediaLib.getEntities(query, limit, incomingNoWeight,
fuzzy, multilang, verbose)
```

A small method to only return entities and no types.

```
$.harubixMatch.WebContent.DBpediaLib.getCategories(query, limit, incomingNoWeight,
orderWeight, fuzzy, multilang, verbose)
```

Here, only categories are returned. The order of the categories is determined by the `finalScore` of the entity they belong to and their order. The higher the order, the less specific the category is. The parameter orderWeight influences how much is subtracted from the finalScore in case the value of order exceeds 1.

Table match for example uses this function for every cell value:

```
$.harubixMatch.WebContent.DBpediaLib.getCategories(tables[i].getColumn(j).getCell(k).getValue(),
20, 0.15, 0.1, fuzzy, multilang, false);
```

E.2 API Documentation

E.2.1 Entity Enrichment

E.2.1.1 Describe Entity (`describe.xsjs`)

```
{
    "abstract": true,
    "thumbnail": true,
    "attributes": [{
        "type": "http://dbpedia.org/property/name"
    }, ...],
     outgoingAssociations : []
}
```

Listing E.1: Entity description API call return values sample

E.2.1.2 Enrich Entity (`enrich.xsjs`)

```
{
    abstract: Give a query, // Mandatory
    filter: Filter the query with a JSON array
        /*
        { abstract :true, attributes :[{ type :
             name }],..],..}
        default: {} [no filter]
        */
}
```

Listing E.2: API call paramteres for entity enrichment

```
{
    abstract :  Apple is a company...  ,
     attributes : [{
        type :  name ,
          value :  Apple Inc.
    }, ...],
     outgoingAssociations : [...]
}
```

Listing E.3: Entity enrichment API call return values sample

E.2.2 Entity Disambiguation (`disambiguate.xsjs`)

```
{
    query: Give a query , // Mandatory
    entityMode: Switch between returning entities or categories {
        values: true or false
        default: true
    },
    limit: Limit for the algorithm(bigger = more results) {
        values: 1 - inf[int]
        default: 75
    },
    incomingNoWeight: Weight of the number of incoming associations
         of an entity(bigger means more weight) {
        values: +-inf[float]
        default: 0.15
    },
    orderWeight: Weight of more specific types(bigger means
        specific types are favored more) {
        values: +-inf[float]
        default: 0.1
    },
    fuzzy: Parameter for fuzzy search {
        values: 0 - 1.0[float]
        default: 1.0[no fuzzy search]
    },
    multilang: Search in INTERLANGUAGE table {
        values: true or false
        default: false
    },
    context: Give context as a JSON array {
        values: {
            context: [ entity : SAP_AG ]
        }
        default: {}
    },
    verbose: Make the output more detailed {
        values: true or false
        default: false
    },
    openSearch: Used for SAPUI5 search suggestion {
        values: true or false
        default: false
```

```
    }
}
```

Listing E.4: API call paramteres for entity disambiguation

```
// Entity mode
{"entities": [{"name":"Microsoft","score":0.9681761690228368},...]}

// Category mode
{"types": [{"name":"Company","score":0.9681761690228368},...]}
```

Listing E.5: Entity disambiguation API call return values sample

E.2.3 Schema Matching (`matchTables.xsjs`)

```
{
    fuzzy: Parameters for fuzzy search {
        values: 0-1.0 [float],
        default: 1.0 [fuzzy disabled]
    },
    multilang: Search in INTERLANGUAGE table {
        values: true or false,
        default: false
    },
    debug: Returns information useful for debugging i.e. weighted
       types per cell
}
```

Listing E.6: API call paramteres for schema matching

```
{
    "matches": [
    {
        column1: 0,
        column2: 0,
        score: 0.627861,
        suggestedName: "Plant"
    },{
        column1 :1,
        column2 :1,
        score: 0.868278,
        suggestedName: "Company"
    },{
```

```
        column1: 2,
        column2: 2,
        score: 0.952602,
        suggestedName: "Country"
    }]
}
```

Listing E.7: Schema matching API call return values sample

Appendix F

Source code for mappings

F.1 Open Licenses Mappings

```
{
        "license_id": ["ODC-PDDL-1.0"],
        "disambiguations": ["Open Data Commons Public Domain
           Dedication and License (PDDL)"]
}, {
        "license_id": ["CC-BY-SA-4.0", "CC-BY-SA-3.0"],
        "disambiguations": ["cc-by-sa", "CC BY-SA", "Creative
           Commons Attribution Share-Alike"]
}, {
        "license_id": ["CC-BY-NC-4.0"],
        "disambiguations": ["Creative Commons Non-Commercial (Any)"
           ]
}, {
        "license_id": ["ODC-BY-1.0"],
        "disambiguations": ["Open Data Commons Attribution License"
           ]
}, {
        "license_id": ["CC-BY-4.0"],
        "disambiguations": ["Creative Commons Attribution", "CC-BY"
           , "CreativeCommonsAttributionCCBY25"]
}, {
        "license_id": ["geogratis"],
        "disambiguations": ["Geogratis"]
}, {
        "license_id": ["CC0-1.0"],
        "disambiguations": ["Creative Commons CCZero", "CC0"]
}, {
        "license_id": ["ODbL-1.0"],
        "disambiguations": ["Open Data Commons Open Database
           License (ODbL)", "ODBL"]
}, {
        "license_id": ["OGL-UK-1.0", "OGL-UK-2.0", "OGL-UK-3.0"],
```

```
        "disambiguations": ["UK Open Government Licence (OGL)", "
            OGL"]
}, {
        "license_id": ["GPL-3.0", "GPL-2.0"],
        "disambiguations": ["GNU General Public License", "gpl-2.0"
            ]
}, {
        "license_id": ["ukclickusepsi"],
        "disambiguations": ["UK PSI (Public Sector Information)
            Click-Use Licence", "ukclickusepsi", "UK Click Use PSI"
            ]
}, {
        "license_id": ["GFDL-1.3-no-cover-texts-no-invariant-
            sections"],
        "disambiguations": ["GNU Free Documentation License"]
}, {
        "license_id": ["MIT"],
        "disambiguations": ["The MIT License (MIT)", "mit-license",
             "MIT License (MIT)", "MIT"]
}, {
        "license_id": ["ukcrown-withrights"],
        "disambiguations": ["UK Crown Copyright with data.gov.uk
            rights", "ukcrown-withrights"]
}, {
        "license_id": ["canadacrown"],
        "disambiguations": ["Canada Crown Copyright", "canada-crown
            "]
}, {
        "license_id": ["BSD-2-Clause", "BSD-3-Clause"],
        "disambiguations": ["bsd-license"]
}, {
        "license_id": ["LGPL-2.1", "LGPL-3.0"],
        "disambiguations": ["GNU Lesser General Public License", "
            lgpl-2.1"]
}, {
        "license_id": ["SPL-1.0"],
        "disambiguations": ["sunpublic", "Sun Public License", "SPL
            "]
}, {
        "license_id": ["GPL-3.0", "GPL-2.0"],
        "disambiguations": ["GNU General Public License", "gpl-3.0"
            ]
}, {
```

```
        "license_id": ["Apache-2.0", "Apache-1.1"],
        "disambiguations": ["Apache License", "apache"]
}
```

Listing F.1: The mappings of the Open Licenses for the LOD Cloud on the Datahub

F.2 Semantic Social News Aggregation Mappings

```
{
        "license_id": ["ODC-PDDL-1.0"],
        "disambiguations": ["Open Data Commons Public Domain
           Dedication and License (PDDL)"]
}, {
        "license_id": ["CC-BY-SA-4.0", "CC-BY-SA-3.0"],
        "disambiguations": ["cc-by-sa", "CC BY-SA", "Creative
           Commons Attribution Share-Alike"]
}, {
        "license_id": ["CC-BY-NC-4.0"],
        "disambiguations": ["Creative Commons Non-Commercial (Any)"
           ]
}, {
        "license_id": ["ODC-BY-1.0"],
        "disambiguations": ["Open Data Commons Attribution License"
           ]
}, {
        "license_id": ["CC-BY-4.0"],
        "disambiguations": ["Creative Commons Attribution", "CC-BY"
           , "CreativeCommonsAttributionCCBY25"]
}, {
        "license_id": ["geogratis"],
        "disambiguations": ["Geogratis"]
}, {
        "license_id": ["CC0-1.0"],
        "disambiguations": ["Creative Commons CCZero", "CC0"]
}, {
        "license_id": ["ODbL-1.0"],
        "disambiguations": ["Open Data Commons Open Database
           License (ODbL)", "ODBL"]
}, {
        "license_id": ["OGL-UK-1.0", "OGL-UK-2.0", "OGL-UK-3.0"],
        "disambiguations": ["UK Open Government Licence (OGL)", "
           OGL"]
```

```
}, {
        "license_id": ["GPL-3.0", "GPL-2.0"],
        "disambiguations": ["GNU General Public License", "gpl-2.0"
           ]
}, {
        "license_id": ["ukclickusepsi"],
        "disambiguations": ["UK PSI (Public Sector Information)
           Click-Use Licence", "ukclickusepsi", "UK Click Use PSI"
           ]
}, {
        "license_id": ["GFDL-1.3-no-cover-texts-no-invariant-
           sections"],
        "disambiguations": ["GNU Free Documentation License"]
}, {
        "license_id": ["MIT"],
        "disambiguations": ["The MIT License (MIT)", "mit-license",
            "MIT License (MIT)", "MIT"]
}, {
        "license_id": ["ukcrown-withrights"],
        "disambiguations": ["UK Crown Copyright with data.gov.uk
           rights", "ukcrown-withrights"]
}, {
        "license_id": ["canadacrown"],
        "disambiguations": ["Canada Crown Copyright", "canada-crown
           "]
}, {
        "license_id": ["BSD-2-Clause", "BSD-3-Clause"],
        "disambiguations": ["bsd-license"]
}, {
        "license_id": ["LGPL-2.1", "LGPL-3.0"],
        "disambiguations": ["GNU Lesser General Public License", "
           lgpl-2.1"]
}, {
        "license_id": ["SPL-1.0"],
        "disambiguations": ["sunpublic", "Sun Public License", "SPL
           "]
}, {
        "license_id": ["GPL-3.0", "GPL-2.0"],
        "disambiguations": ["GNU General Public License", "gpl-3.0"
           ]
}, {
        "license_id": ["Apache-2.0", "Apache-1.1"],
        "disambiguations": ["Apache License", "apache"]
```

```
}
```

Listing F.2: The mappings of YouTube categories with DMOZ and Alchemy API

```
{
        "alchemy": "Arts & Entertainment",
        "alchemyCode" : "arts_entertainment",
        "DMOZ": ["Arts","Society"],
        "stack" : ["music","movies"]
},
{
        "alchemy": "Business",
        "alchemyCode" : "business",
        "DMOZ": ["Business", "News", "Shopping"],
        "stack" : ["money","pm","answers.onstartups","patents","
           quant"]
},
{
        "alchemy": "Computers & Internet",
        "alchemyCode" : "computer_internet",
        "DMOZ": ["Computers", "Science"],
        "stack" : ["stackoverflow","serverfault","superuser"]
},
{
        "alchemy": "Culture & Politics",
        "alchemyCode" : "culture_politics",
        "DMOZ": ["News", "Society","history"],
        "stack" : ["politics"]
},
{
        "alchemy": "Gaming",
        "alchemyCode" : "gaming",
        "DMOZ": ["Games"],
        "stack" : ["gaming"]
},
{
        "alchemy": "Health",
        "alchemyCode" : "health",
        "DMOZ": ["Health", "Society"],
        "stack" : ["fitness","sustainability"]
},
{
        "alchemy": "Law & Crime",
```

```
        "alchemyCode": "law_crime",
        "DMOZ": ["News", "Society"]
},
{
        "alchemy": "Religion",
        "alchemyCode": "religion",
        "DMOZ": ["Reference", "Society"],
        "stack" : ["islam","christianity"]
},
{
        "alchemy": "Recreation",
        "alchemyCode": "recreation",
        "stack" : ["philosophy","photo"],
        "DMOZ": ["Recreation", "Society"]
},
{
        "alchemy": "Science & Technology",
        "alchemyCode": "science_technology",
        "DMOZ": ["Science", "News"],
        "stack" : ["stats","math"]
},
{
        "alchemy": "Sports",
        "alchemyCode": "sports",
        "DMOZ": ["Sports", "News"],
        "stack" : ["sports"]
},
{
        "alchemy": "Weather",
        "alchemyCode": "weather",
        "DMOZ": ["News"]
}
```

Listing F.3: The mappings of the StackExchange services with DMOZ and Alchemy API

Bibliography

[1] Z. Abedjan, T. Gruetze, A. Jentzsch, and F. Naumann. Profiling and mining RDF data with ProLOD++. In *30th IEEE International Conference on Data Engineering (ICDE)*, pages 1198–1201, 2014.

[2] Maribel Acosta, Amrapali Zaveri, Elena Simperl, and Dimitris Kontokostas. Crowdsourcing Linked Data quality assessment. In *12th International Semantic Web Conference (ISWC)*, 2013.

[3] K. Alexander, R. Cyganiak, M. Hausenblas, and J.Zhao. Describing Linked Datasets. In *2nd International Workshop on Linked Data on the Web (LDOW)*, 2009.

[4] Rula Anisa and Amrapali Zaveri. Methodology for Assessment of Linked Data Quality. In *1st Workshop on Linked Data Quality (LDQ)*, 2014.

[5] Ahmad Assaf, Eldad Louw, Aline Senart, Corentin Follenfant, Raphaël Troncy, and David Trastour. RUBIX: a framework for improving data integration with linked data. In *International Workshop on Open Data (WOD'12)*, pages 13–21, 2012.

[6] Ahmad Assaf and Aline Senart. Data Quality Principles in the Semantic Web. In *6th International Conference on Semantic Computing ICSC '12*, 2012.

[7] Ahmad Assaf, Aline Senart, and Raphaël Troncy. SNARC - An Approach for Aggregating and Recommending Contextualized Social Content. In *The Semantic Web: ESWC 2013 Satellite Events, Revised Selected Papers*, pages 319–326, 2013.

[8] Ahmad Assaf, Aline Sénart, and Raphaël Troncy. Roomba: Automatic Validation, Correction and Generation of Dataset Metadata. In *24th World Wide Web Conference (WWW), Demos Track*, Florence, Italy, 2015.

[9] Ahmad Assaf, Raphaël Troncy, and Aline Sénart. An Objective Assessment Framework & Tool for Linked Data Quality - Enriching Dataset Profiles with Quality Indicators (Major Revision). *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2015.

[10] Ahmad Assaf, Raphaël Troncy, and Aline Sénart. HDL-Towards a Harmonized Dataset Model for Open Data Portals. In *2nd International Workshop on*

Dataset PROFIling & fEderated Search for Linked Data, Portoroz, Slovenia, 2015.

[11] Ahmad Assaf, Raphaël Troncy, and Aline Sénart. Roomba: An Extensible Framework to Validate and Build Dataset Profiles. In *12th European Semantic Web Conference (ESWC)*, Portoroz, Slovenia, 2015.

[12] Ahmad Assaf, Raphaël Troncy, and Aline Sénart. Whats up LOD Cloud? Observing The State of Linked Open Data Cloud Metadata. In *12th European Semantic Web Conference (ESWC)*, Portoroz, Slovenia, 2015.

[13] Sören Auer, Jan Demter, Michael Martin, and Jens Lehmann. LODStats - an Extensible Framework for High-performance Dataset Analytics. In *18th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 353–362, Galway, Ireland, 2012.

[14] C. Avitha, G. Sadasivam Sudha, and Sangeetha N Shenoy. Ontology Based Semantic Integration of Heterogeneous Databases. *European Journal of Scientific Research*, page 115, 2011.

[15] Mike Bergman. Deconstructing the Google Knowledge Graph. http://www.mkbergman.com/1009/deconstructing-the-google-knowledge-graph.

[16] Tim Berners-Lee. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986, 2005. http://tools.ietf.org/html/rfc3986.

[17] Tim Berners-Lee. Linked Data - Design Issues. W3C Personal Notes, 2006. http://www.w3.org/DesignIssues/LinkedData.

[18] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[19] Dimitris Berrueta, Sergio Fern"andez, and Iv"an Frade. Cooking HTTP content negotiation with Vapour. In *4th Workshop on Scripting for the Semantic Web (SFSW'08)*, 2008.

[20] Gasser Les Stvilia Besiki, , Michael B. Twidale, and Linda C. Smith. A framework for information quality assessment. *Journal of the American Society for Information Science and Technology*, 2007.

[21] Christian Bizer. Evolving the Web into a Global Data Space. In *28th British National Conference on Advances in Databases*, 2011.

[22] Christian Bizer and Tom Heathand Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[23] Christian Bizer and Richard Cyganiak. Quality-driven information filtering using the wiqa policy framework. *Jorunal of Web Semantics*, 7(1), 2009.

[24] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - A Crystallization Point for the Web of Data. *Journal of Web Semantics*, 7(3), 2009.

[25] C. Bohm, F. Naumann, Z. Abedjan, D. Fenz, T. Grutze, D. Hefenbrock, M. Pohl, and D. Sonnabend. Profiling linked open data with ProLOD. In *26th International Conference on Data Engineering Workshops (ICDEW)*, 2010.

[26] Christoph Böhm, Gjergji Kasneci, and Felix Naumann. Latent Topics in Graph-structured Data. In *21st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2663–2666, Maui, Hawaii, USA, 2012.

[27] Christoph BöHm, Johannes Lorey, and Felix Naumann. Creating voiD Descriptions for Web-scale Data. *Journal of Web Semantics*, 9(3):339–345, 2011.

[28] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *ACM International Conference on Management of Data (SIGMOD)*, 2008.

[29] D Boyd and Kate Crawford. Six provocations for big data. *A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society*, 2011.

[30] Dan Brickley and R.V. Guha. RDF Schema 1.1. W3C Recommendation, 2014. http://www.w3.org/TR/rdf-schema.

[31] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In *7th International Conference on World Wide Web (WWW'98)*, 1998.

[32] Martin Brümmer, Ciro Baron, Ivan Ermilov, Markus Freudenberg, Dimitris Kontokostas, and Sebastian Hellmann. DataID: Towards Semantically Rich Metadata for Complex Datasets. In *10th International Conference on Semantic Systems*, 2014.

[33] Carlos Buil-Aranda and Aidan Hogan. SPARQL Web-Querying Infrastructure: Ready for Action? In *12th International Semantic Web Conference (ISWC)*, 2013.

[34] Soumen Chakrabarti, Byron E. Dom, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon Kleinberg. Mining the web's link structure. *Computer*, 1999.

[35] Didier Cherix, Ricardo Usbeck, Andreas Both, and Jens Lehmann. CROCUS: Cluster-based ontology data cleansing. In *2nd International Workshop on Semantic Web Enterprise Adoption and Best Practice*, 2014.

[36] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A Framework for Benchmarking Entity-annotation Systems. In *22nd World Wide Web Conference (WWW)*, 2013.

[37] Richard Cyganiak, Holger Stenzhorn, Renaud Delbru, Stefan Decker, and Giovanni Tummarello. Semantic Sitemaps: Efficient and Flexible Access to Datasets on the Semantic Web. In *5th European Semantic Web Conference (ESWC)*, pages 690–704, Tenerife, Spain, 2008.

[38] Richard Cyganiak, Jun Zhao, Michael Hausenblas, and Keith Alexander. Describing Linked Datasets with the VoID Vocabulary. W3C Note, 2011. http://www.w3.org/TR/void/.

[39] Altigran Soares da Silva, Denilson Barbosa, João M. B. Cavalcanti, and Marco A. S. Sevalho. Labeling Data Extracted from the Web. In *On The Move Confederated International Conferences*, pages 1099–1116, 2007.

[40] Mathieu d'Aquin and Enrico Motta. Watson, More Than a Semantic Web Search Engine. *Semantic Web Journal*, 2011.

[41] Tim Davies, Raed Sharif, and Jose Alonso. Open Data Barometer - Global Report. Technical report, World Wide Web Foundation, 2015. http://barometer.opendataresearch.org/.

[42] Jeremy Debattista, Christoph Lange, and Sören Auer. daQ, an Ontology for Dataset Quality Information. In *Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014)*, 2014.

[43] Jeremy Debattista, Santiago Londoño, Christoph Lange, and Sören Auer. LUZZU - A framework for linked data quality assessment. *CoRR*, abs/1412.3750, 2014.

[44] Renaud Delbru, Nickolai Toupikov, and Michele Catasta. Hierarchical link analysis for ranking web data. In *7th European Semantic Web Conference (ESWC)*, 2010.

[45] L Ding, Tim Finin, A Joshi, R Pan, and R Cost. Swoogle: A semantic web search and metadata engine. In *13st ACM International Conference on Information and Knowledge Management (CIKM)*, 2004.

[46] Diaz-Aviles Ernesto, Drumond Lucas, Schmidt-Thieme Lars, and Nejdl Wolfgang. Real-time top-n recommendation in social streams. In *6th ACM conference on Recommender systems - RecSys*, 2012.

[47] Bakshy Eytan, Rosenn Itamar, Marlow Cameron, and Adamic Lada. The role of social networks in information diffusion. In *21th International Conference on World Wide Web (WWW'12)*, 2012.

[48] Besnik Fetahu, Stefan Dietze, Bernardo Pereira Nunes, Marco Antonio Casanova, Davide Taibi, and Wolfgang Nejdl. A Scalable Approach for Efficiently Generating Structured Dataset Topic Profiles. In *11th European Semantic Web Conference (ESWC)*, 2014.

[49] Tim Finin, Zareen Syed, James Mayfield, Paul Mcnamee, and Christine Piatko. Using Wikitology for Cross-Document Entity Coreference Resolution. In *AAAI Spring Symposium on Learning*, 2009.

[50] Annika Flemming. Quality Characteristics of Linked Data Publishing Datasources. Master's thesis, Humboldt-Universitt zu Berlin, 2010.

[51] Giorgos Flouris, Yannis Roussakis, and M Poveda-Villalón. Using provenance for quality assessment and repair in linked open data. In *2nd Joint Workshop on Knowledge Evolution and Ontology Dynamics (EvoDyn'12)*, 2012.

[52] Benedikt Forchhammer, Anja Jentzsch, and Felix Naumann. LODOP - Multi-Query Optimization for Linked Data Profiling Queries. In *International Workshop on Dataset PROFIling and fEderated Search for Linked Data (PROFILES)*, Heraklion, Greece, 2014.

[53] Philipp Frischmuth, Sören Auer, Sebastian Tramp, Jörg Unbehauen, Kai Holzweißig, and Carl-Martin Marquardt. Towards Linked Data based Enterprise Information Integration. In *Workshop on Semantic Web Enterprise Adoption and Best Practice Co-located with 12th International Semantic Web Conference (ISWC'13)*, 2013.

[54] Philipp Frischmuth, Jakub Klímek, Sören Auer, Sebastian Tramp, Jörg Unbehauen, Kai Holzweißig, and Carl-Martin Marquardt. Linked Data in Enterprise Information Integration. 2012.

[55] Matias Frosterus, Eero Hyvönen, and Joonas Laitio. Creating and Publishing Semantic Metadata about Linked and Open Datasets. In *Linking Government Data*. 2011.

[56] Matias Frosterus, Eero Hyvönen, and Joonas Laitio. DataFinland - A Semantic Portal for Open and Linked Datasets. In *8th Extended Semantic Web Conference (ESWC)*, pages 243–254, 2011.

[57] Christian Fürber and Martin Hepp. SWIQA - A Semantic Web information quality assessment framework. 2011.

[58] W3C OWL Working Group. OWL 2 Web Ontology Language. W3C Recommendation, 2012. http://www.w3.org/TR/owl2-overview.

[59] Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2), 1993.

[60] Christophe Guéret, Paul Groth, Claus Stadler, and Jens Lehmann. Assessing Linked Data Mappings Using Network Measures. In *9th European Semantic Web Conference (ESWC)*, 2012.

[61] Richard Hammell, Carl Bates, Harvey Lewis, Costi Perricos, Louise Brett, and David Branch. Open Data: Driving growth, ingenuity and innovation. Technical report, Deloitte LLP, 2012. http://www2.deloitte.com/content/dam/Deloitte/uk/Documents/deloitte-analytics/open-data-driving-growth-ingenuity-and-innovation.pdf.

[62] Patricia Harpring. *Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works*. Getty Research Institute, 2010.

[63] Andreas Harth, Katja Hose, Marcel Karnstedt, Axel Polleres, Kai-Uwe Sattler, and Jürgen Umbrich. Data Summaries for On-demand Queries over Linked Data. In *19th World Wide Web Conference (WWW)*, 2010.

[64] Andreas Harth, Sheila Kinsella, and Stefan Decker. Using naming authority to rank data and ontologies for web search. In *8th International Semantic Web Conference (ISWC)*, 2009.

[65] Olaf Hartig and Jun Zhao. Using web data provenance for quality assessment. In *8th International Semantic Web Conference (ISWC)*, 2009.

[66] Bernhard Haslhofer and Niko Popitsch. DSNotify: Detecting and Fixing Broken Links in Linked Data Sets. In *8th International Workshop on Web Semantics*, 2009.

[67] Oktie Hassanzadeh, Duan, Achille Fokoue, Anastasios Kementsietsidis, Kavitha Srinivas, and Michael J. Ward. Helix: Online Enterprise Data Analytics. In *20th International Conference Companion on World Wide Web (WWW'11)*, pages 225–228, 2011.

[68] Michael Hausenblas, Wolfgang Halb, Yves Raimond, Lee Feigenbaum, and Danny Ayer. SCOVO: Using Statistics on the Web of Data. In *ESWC*, 2009.

[69] Aidan Hogan, Andreas Harth, and Stefan Decker. ReConRank: A Scalable Ranking Method for Semantic Web Data with Context. In *2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.

[70] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. Weaving the pedantic web. 2010.

[71] Aidan Hogan, JüRgen Umbrich, Andreas Harth, Richard Cyganiak, Axel Polleres, and Stefan Decker. An empirical survey of Linked Data conformance. *Journal of Web Semantics*, 2012.

[72] Renato Iannella and James McKinney. vCard Ontology - for describing People and Organizations. W3C Interest Group Note, 2014. http://www.w3.org/TR/vcard-rdf.

[73] Halo Business Inteligence. Data Quality - Why you should care about the cleanliness of your data. Technical report, 2013.

[74] Antoine Isaac and Ed Summers. SKOS Simple Knowledge Organization System Primer. W3C Working Group Note, 2009.

[75] Robert Isele, Jürgen Umbrich, Christian Bizer, and Andreas Harth. LDspider: An Open-source Crawling Framework for the Web of Linked Data. In *9th International Semantic Web Conference (ISWC), Posters & Demos Track*, 2010.

[76] Cantador Iván and Bellogín Alejandro. Semantic contextualisation of social tag-based profiles and item recommendations. In *12th Internationl Conference on E-Commerce and Web Technolgoies*, 2011.

[77] Prateek Jain, Pascal Hitzler, Krzysztof Janowicz, and Chitra Venkatramani. There's No Money in Linked Data, 2013. http://knoesis.wright.edu/faculty/pascal/pub/nomoneylod.pdf.

[78] Anja Jentzsch. Profiling the Web of Data. In *13th International Semantic Web Conference (ISWC), Doctoral Consortium*, Trentino, Italy, 2014.

[79] Tobias Käfer, Ahmed Abdelrahman, Jürgen Umbrich, Patrick O'Byrne, and Aidan Hogan. Observing Linked Data Dynamics. In *10th European Semantic Web Conference (ESWC)*, 2013.

[80] Beverly K. Kahn, Diane M. Strong, and Richard Y. Wang. Information quality benchmarks: product and service performance. *Communications of the ACM*, 2002.

[81] C.Maria Keet, María del Carmen Suárez-Figueroa, and María Poveda-Villalón. The Current Landscape of Pitfalls in Ontologies. In *International Conference on Knowledge Engineering and Ontology Development (KEOD)*, 2013.

[82] Shahan Khatchadourian and Mariano P. Consens. ExpLOD: Summary-based Exploration of Interlinking and RDF Usage in the Linked Open Data Cloud. In *7th Extended Semantic Web Conference (ESWC)*, pages 272–287, Heraklion, Greece, 2010.

[83] Ralph Kimball, Laura Reeves, Warren Thornthwaite, Margy Ross, and Warren Thornwaite. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses with CD Rom.* John Wiley & Sons, Inc., 1st edition, 1998.

[84] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *ACM Journal*, 1999.

[85] Mathias Konrath, Thomas Gottron, Steffen Staab, and Ansgar Scherp. SchemEX - Efficient Construction of a Data Catalogue by Stream-based Indexing of Linked Data. *Journal of Web Semantics*, 16, 2012.

[86] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. Test-driven Evaluation of Linked Data Quality. In *23rd International Conference on World Wide Web (WWW'14)*, 2014.

[87] Dimitris Kontokostas, Amrapali Zaveri, Sören Auer, and Jens Lehmann. TripleCheckMate: A Tool for Crowdsourcing the Quality Assessment of Linked Data. *4th Conference on Knowledge Engineering and Semantic Web*, 2013.

[88] Kovács-Láng. Global Terrestrial Observing System. Technical report, GTOS Central and Eastern European Terrestrial Data Management and Accessibility Workshop, 2000.

[89] Charles J. Kowalski. On the Effects of Non-Normality on the Distribution of the Sample Product-Moment Correlation Coefficient. *Journal of the Royal Statistical Society*, 1972.

[90] S. Lalithsena, P. Hitzler, A. Sheth, and P. Jain. Automatic Domain Identification for Linked Open Data. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pages 205–212, 2013.

[91] Andreas Langegger and Wolfram Woss. RDFStats - An Extensible RDF Statistics Generator and Library. In *20th International Workshop on Database and Expert Systems Application (DEXA)*, pages 79–83, 2009.

[92] Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.

[93] Steve LaValle, Eric Lesser, Rebecca Shockley, Michael S. Hopkins, and Nina Kruschwitz. Big Data, Analytics and the Path From Insights to Value. *MIT Sloan Management Review*, 2011.

[94] Timothy Lebo, Satya Sahoo, and Deborah McGuinness. PROV-O: The PROV Ontology. W3C Recommendation, 2013. http://www.w3.org/TR/prov-o.

[95] Jens Lehmann and Soeren Sonnenburg. DL-Learner: Learning Concepts in Description Logics. *Journal of Machine Learning Research*, 2009.

[96] Maurizio Lenzerini. Data Integration: A Theoretical Perspective. In *21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 233–246, 2002.

[97] Maurizio Lenzerini. Data Integration: A Theoretical Perspective. In *21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2002.

[98] Jure Leskovec and Christos Faloutsos. Sampling from Large Graphs. In *12thth ACM International Conference on Knowledge Discovery and Data Mining (KDD'12)*, 2006.

[99] Huiying Li. Data Profiling for Semantic Web Data. In *International Conference on Web Information Systems and Mining (WISM)*, pages 472–479, 2012.

[100] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and Searching Web Tables Using Entities, Types and Relationships. *VLDB Endowment*, pages 1338–1347, 2010.

[101] Joseph Juran. M. and A. Blanton Godfrey. *Juran's quality handbook*. McGraw Hill, 1999.

[102] Fadi Maali and John Erickson. Data Catalog Vocabulary (DCAT). W3C Recommendation, 2014. http://www.w3.org/TR/vocab-dcat/.

[103] Christian Mader, Bernhard Haslhofer, and Antoine Isaac. Finding quality issues in SKOS vocabularies. *Theory and Practice of Digital Libraries*, 2012.

[104] Eetu Mäkelä. Aether - Generating and Viewing Extended VoID Statistical Descriptions of RDF Datasets. In *11^{th} European Semantic Web Conference (ESWC), Demo Track*, Heraklion, Greece, 2014.

[105] James Manyika and Almasi Doshi Elizabeth. Open data: Unlocking innovation and performance with liquid information. Technical report, McKinsey Business Technology Office, 2013.

[106] Nicolas Marie, Fabien Gandon, Myriam Ribière, and Florentin Rodio. Discovery Hub: On-the-fly Linked Data Exploratory Search. In *The 9^{th} International Conference on Semantic Systems*, 2013.

[107] Pablo Mendes, Hannes Mühleisen, and Christian Bizer. Sieve: linked data quality assessment and fusion. 2012.

[108] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *7^{th} International Conference on Semantic Systems*, 2011.

[109] Nandana Mihindukulasooriya, Raul Garcia-Castro, and Miguel Esteban Gutiérrez. Linked Data Platform as a novel approach for Enterprise Application Integration. In *4^{th} International Workshop on Consuming Linked Data (COLD'13)*, 2013.

[110] Peter Mika. *Social Networks and the Semantic Web*, volume 5 of *Semantic Web and Beyond*. Springer, 2007.

[111] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference. W3C Recommendation, 2009. http://www.w3.org/TR/skos-reference/.

[112] Renée J. Miller and Periklis Andritsos. Schema Discovery. *IEEE Data Engineering Bulletin*, 26:40–45, 2003.

[113] Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta. What Should I Link to? Identifying Relevant Sources and Classes for Data Linking. In *Joint International Semantic Technology Conference (JIST)*, 2011.

[114] Tommaso Di Noia, Roberto Mirizzi, Vito Ostuni Claudio, Davide Romito, and Markus Zanker. Linked Open Data to Support Content-based Recommender Systems. In *8^{th} International Conference on Semantic Systems - I-SEMANTICS '12*, 2012.

[115] Lawrence Page, Sergey Brin, Motwani Rajeev, and Winograd Terry. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, 1998.

[116] E. Peukert, J. Eberius, and E. Rahm. AMC - A framework for modelling and comparing matching systems as matching processes. In *IEEE 27^{th} International Conference on Data Engineering (ICDE'11)*, 2011.

[117] Eric Peukert, Julian Eberius, and Erhard Rahm. A Self-Configuring Schema Matching System. In *IEEE 28^{th} International Conference on Data Engineering (ICDE'12)*, 2012.

[118] Archer Phil and Shukair Gofran. Asset Description Metadata Schema (ADMS). W3C Working Group Note, 2013. http://www.w3.org/TR/vocab-adms.

[119] Emmanuel Pietriga, Christian Bizer, David Karger, and Ryan Lee. Fresnel: A Browser-Independent Presentation Vocabulary for RDF. In *5^{th} International Semantic Web Conference (ISWC'06)*, pages 158–171, 2006.

[120] Joshua Porter. *Designing for the Social Web.* New Riders, 2008.

[121] Mara Poveda-Villalón, MariCarmen Suárez-Figueroa, and Asunción Gmez-Pérez. Validating Ontologies with OOPS! In *18^{th} International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, 2012.

[122] Daniel Preotiuc-Pietro, Sina Samangooei, Trevor Cohn, Nicholas Gibbins, and Mahesan Niranjan. Trendminer: An architecture for real time analysis of social

media text. In *6th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2012.

[123] NISO Press. Understanding Metadata. Technical report, National Information Standards Organization, 2004.

[124] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Recommendation, 2008. http://www.w3.org/TR/rdf-sparql-query/.

[125] D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. Automatic Web News Extraction Using Tree Edit Distance. In *13th International World Wide Web Conference (WWW'04)*, pages 502–601, 2004.

[126] Dave Reynolds. The Organization Ontology. W3C Recommendation, 2014. http://www.w3.org/TR/vocab-org.

[127] Edna Ruckhaus, Oriana Baldizan, and Maria-Esther Vidal. Analyzing Linked Data Quality with LiQuate. In *11th European Semantic Web Conference (ESWC)*, 2014.

[128] Olivia Parr Rud. *Business Intelligence Success Factors: Tools for Aligning Your Business in the Global Economy*. John Wiley & Sons, 2009.

[129] Massimiliano Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the Linked Data Best Practices in Different Topical Domains. In *13th International Semantic Web Conference (ISWC)*, 2014.

[130] Cambridge Semantics. RDF-101. http://www.cambridgesemantics.com/semantic-university/rdf-101. Accessed: 2013-09-07.

[131] Evren Sirin, Michael Smith, and Evan Wallace. Opening, Closing Worlds - On Integrity Constraints. In *5th OWLED Workshop on OWL: Experiences and Directions*, 2008.

[132] Dagobert Soergel. Thesauri and ontologies in digital libraries. In *2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, 2002.

[133] Thomas Steiner and Stefan Mirea. SEKI@home or Crowdsourcing an Open Knowledge Graph. In *1st International Workshop on Knowledge Extraction & Consolidation from Social Media (KECSM'12)*, Boston, USA, 2012.

[134] Umberto Straccia and Raphaël Troncy. oMAP: Combining Classifiers for Aligning Automatically OWL Ontologies. In *6th International Conference on Web Information Systems Engineering*, 2005.

[135] Fabian Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *16th International World Wide Web Conference (WWW)*, 2007.

[136] Osma Suominen and Eero Hyvönen. Improving the quality of SKOS vocabularies with skosify. In *The 18th International Conference on Knowledge Engineering and Knowledge Management*, 2012.

[137] Osma Suominen and Christian Mader. Assessing and Improving the Quality of SKOS Vocabularies. *Journal on Data Semantics*, 2013.

[138] Zareen Syed, Tim Finin, Varish Mulwad, and Anupam Joshi. Exploiting a Web of Semantic Data for Interpreting Tables. In *2nd Web Science Conference*, 2010.

[139] Jiao Tao, Li Ding, and Deborah McGuinness. Instance Data Evaluation for Semantic Web-Based Knowledge Management Systems. In *42nd Hawaii International Conference on System Sciences, HICSS'09*, pages 1–10, 2009.

[140] Nickolai Toupikov, J Umbrich, and Renaud Delbru. DING! Dataset ranking using formal descriptions. In *2nd International Workshop on Linked Data on the Web (LDOW)*, 2009.

[141] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig.ma: Live views on the Web of data. *Journal of Web Semantics*, 8(4), 2010.

[142] Giovanni Tummarello, Renaud Delbru, and Eyal Oren. Sindice.com: Weaving the open linked data. In *6th International Semantic Web Conference (ISWC)*, 2007.

[143] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga-Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. GERBIL - General Entity Annotation Benchmark Framework. In *24th World Wide Web Conference (WWW)*, 2015.

[144] Zanardi Valentina and Capra L. Social ranking: uncovering relevant content using tag-based recommender systems. In *2nd ACM conference on Recommender systems - RecSys*, 2008.

[145] Mateja Verlic. LODGrefine - LOD-enabled Google Refine in Action. In *8th International Conference on Semantic Systems - I-SEMANTICS '12*, 2012.

[146] Graham Vickery. Review of Recent Studies on PSI-use and Related Market Developments. Technical report, EC DG Information Society, 2011.

[147] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-Based Integration of Information - A Survey of Existing Approaches. In *IJCAI Workshop: Ontologies and Information*, pages 108–117, 2001.

[148] Jiying Wang and Frederick Lochovsky. Data Extraction and Label Assignment for Web Databases. In *12th International World Wide Web Conference (WWW'03)*, pages 187–196, 2003.

[149] Richard Y. Wang and Diane M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 1996.

[150] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality Assessment Methodologies for Linked Open Data. *Semantic Web Journal*, 2012.