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

Jury:

Reviewers:

Prof. Philippe Cudré-Mauroux	-	University of Fribourg, Switzerland
Prof. Marie Aude Aufaure	-	École Centrale Paris, France

Examiners:

Prof. Pierre Senellart	-	Telecom ParisTech, France
Dr. Stefan Dietze	-	Leibniz University, Germany

Supervisor:

Dr. Raphaël Troncy	-	EURECOM, France
Dr. Aline Senart	-	SAP, France

In the Name of God, Most Gracious, Most Merciful

Acknowledgments

Working as a PhD student in EURECOM and SAP was a great experience that would not be achieved without the help and support of many people, who I would like to acknowledge here.

First and foremost, I would like to thank my supervisors Dr. Raphaël Troncy and Dr. Aline Senart for their invaluable support and great guidance throughout my studies. I would like to express my gratitude to them for providing me with the freedom to pursue my research and the valuable feedback along the way. This work would not have been possible without their scientific knowledge, constructive advice and deep compassion.

I would like to thank my committee members, the reviewers Prof. Philippe Cudré-Mauroux and Prof. Marie Aude Aufaure, and furthermore the examiners Dr. Pierre Senellart and Dr. Stefan Dietze for their precious time, shared positive insight and guidance.

I owe my deepest gratitude to my parents, Dr. Abdel Mouti Assaf and Renad Al Fahoum and to my sisters Malak, Dima and Noor for their unwavering encouragement, devotion and love and for pushing me always to be the best. Last but not least, special thanks go to my friends and colleagues in SAP and EURECOM for their constant friendship, moral and infinite support.

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 most challenging task in most BI projects large and small. Self-service data provisioning aims at tackling this problem by providing intuitive datasets discovery, data 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 and so on. The increasing diversity of these datasets makes it difficult to annotate them with a fixed number of predefined 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 which facilitates data search and acquisition for business users.

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 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 by business users. Second, 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 implementing this framework 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 business 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.

Table of Contents

List of Figures

List of Tables

Listings

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 (WWW 2015), 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 2^{nd} Workshop on Linked Data Quality (LDQ), May 2015, Portoroz, Slovenia.

2. **Ahmad Assaf**, Raphaël Troncy and Aline Senart: **HDL - Towards A Harmonized Dataset Model for Open Data Portals**. In 2^{nd} 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 2^{nd} 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**. In International Workshop on Data Quality Management and Semantic Technologies (DQMST), 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 1^{st} 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 Metadata Schema
CRM	Customer Relationships Management
CSV	Comma Separated Values
DI	Data Integration
DMS	Data Management Systems
DW	Data Warehousing
EDW	Enterprise Data Warehouse
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 most 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 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 [**?**]. In addition to this heterogeneity, the amount of information in enterprise databases and on-line data stores expands exponentially each year. Enterprise Big Data is not 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 [**?**]. 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 [?, ?]. Recently, ontology-based data integration approaches have been suggested where ontologies are used to describe the data, queries and mappings between them [?]. A slightly different approach is the use of the Linked Data paradigm [?] 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] [?]. 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 [?]. 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 [?].

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 a portal. He oversees the creation of users, organizations and datasets. Administrators try to ensure a certain data quality level by

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

continuously checking for spam and manually enhancing dataset descriptions and annotations.

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

In our scenario, **Dan** 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. Dan receives a memo from his management to create a report comparing the number of car accidents that occurred in France for this 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, Dan is able to collect the data needed to create his report for the French side. Dan also issues an official request to the Department of Transport in UK to collect the data needed. However, Dan knows that the process takes a long time and his management needs the report within days. Dan is familiar with the Open Data movement and starts his journey searching through different data portals in the UK.

Paul is a Data Portal Administrator for the `data.gov.uk`. He continuously oversees the processes of acquiring, preparing and publishing datasets. Paul always tries to ensure that the data published is of high quality and contains sufficient attached metadata to easily enable search and discovery. Paul often receives complaints about inaccurate or spam datasets. He manually removes and fixes 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 categories 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 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 semantically similar but yet different. **Dan** needs powerful

[3] http://saplumira.com/

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. **Dan** 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 **Dan** 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 **Dan** 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 **Dan** to find them [**?**].

- 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, **Dan** has difficulties finding datasets with a specific geographical coverage as this information is missing from almost all of the examined datasets profiles.

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

[4] http://dbpedia.org

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 **Paul** need 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 **Dan** 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 **??**) to enable self-service data provisioning for internal and external data sources in the enterprise. The framework contributes to the three main challenges described above. In summary, the main contributions of this work are as follows:

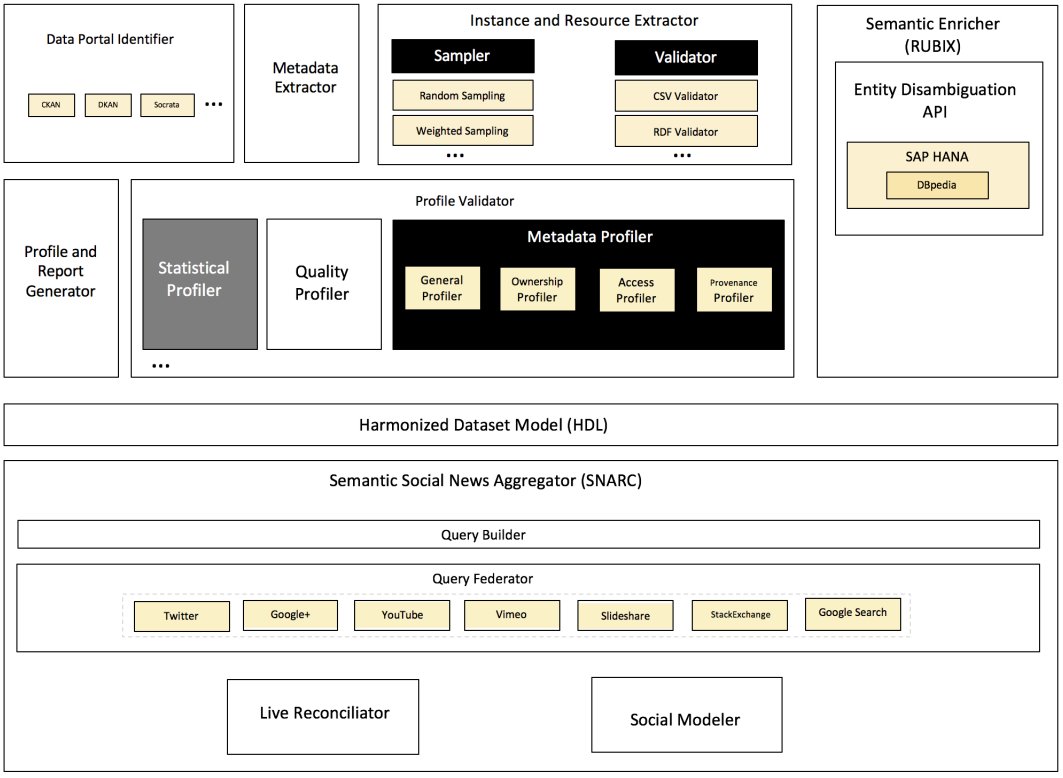

Figure 1.1: Architecture diagram for enabling self-service data provisioning

1.4.1 Contributions on Dataset Maintenance & Discovery

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

- 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 **??**). First, 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 **??**).
- We have analyzed the landscape of dataset profiling tools and discovered various gaps (see Section **??**). As a result, we proposed Roomba, a scalable automatic framework for extracting, validating, correcting and generating descriptive linked dataset profiles (see Section **??**). 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.

1.4.2 Contributions on Dataset Quality Control

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

- We proposed a linked data quality assessment framework focusing on the data's objective metrics. We have identified a total of 64 quality indicators that were mapped when suitable to four main categories (entity, dataset, links, models) corresponding to the core Linked Data publishing principles. (see Section **??**).
- Upon surveying the landscape of data quality tools, we noticed a lack in automatic tools to check the dataset quality metrics proposed in our framework (see Section **??**). As a result, we extended Roomba to perform a set of data quality checks on Linked datasets. Our extension covers most of the quality indicators proposed with focus on completeness, correctness, provenance and licensing (see Section **??**).

1.4.3 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 (see Section **??**).

- 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 presented these choices explicitly using the Fresnel vocabulary, so that any application could read this configuration file for deciding which properties of an entity is worth to enrich (see Section **??**).

- 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 (see Chapter **??**.

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 **??**, 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 [**?**, **?**, **?**, **?**, **?**, **?**].

- **Chapter ??** 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 ??** 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 and search and spam detection.

- **Chapter ??** 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. We show that the overall state of Linked Data portals needs more attention.

- **Chapter ??** 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 **??**, 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 [**?**, **?**]

- **Chapter ??** 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 ??** 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 ??** 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 ??

In Part **??**, 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 against various data portals.

In Chapter **??**, 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 **??**, 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 **??**, 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 **??**, 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 form (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 [**?**]. 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. [**?**] 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 difference between Semantic Web and other data-related technologies is that the Se-

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

mantic 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. Let us take for example, a use case where a website publishes a database about a specific product line, with descriptions and prices, while another publishes a database of product reviews. The Semantic Web standards make it easier to write an application to mesh those 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 sections:

2.1.1 Resource Description Framework (RDF)

Resource Description Framework (RDF) [**?**] 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) [**?**] 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.

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

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

a specific aspect, characteristic, attribute, or relation used to describe a resource [**?**]. 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:** a property of the subject and expresses the relationship between the subject and the object.
- **Object:** the value of the property. It can be a URI of a resource, a blank node or a literal.

Figure **??** 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.

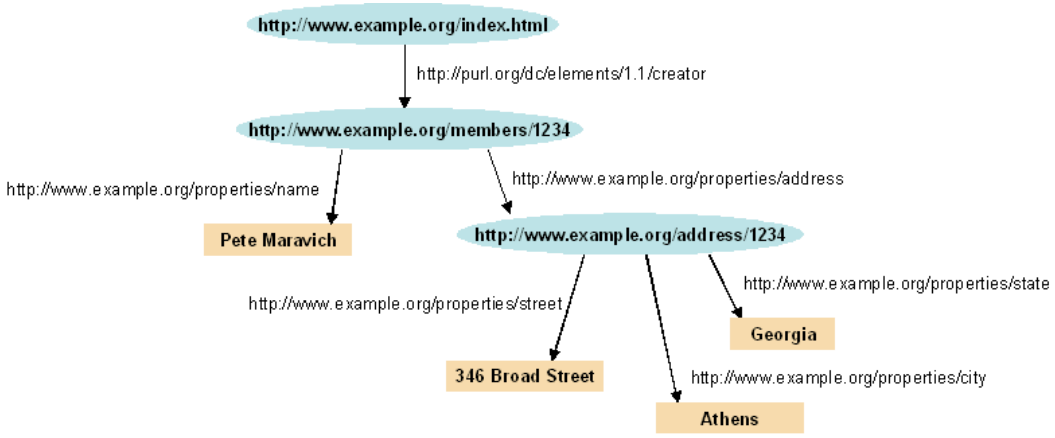

Figure 2.1: Example of RDF representation of an address

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.

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"

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

Cambridge Semantics [?]

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 [?]. 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) [?] provides a basic vocabulary to interpret RDF statements, describe taxonomies of classes and properties and define 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.

- Resources are instances of one or more class (*rdfs:class*). Classes are organized in a hierarchy using *rdfs:subClassOf* property.

- Properties 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) [**?**] 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) [**?**] 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 resolving 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" [**?**].

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

1. Use URIs as names for things
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)

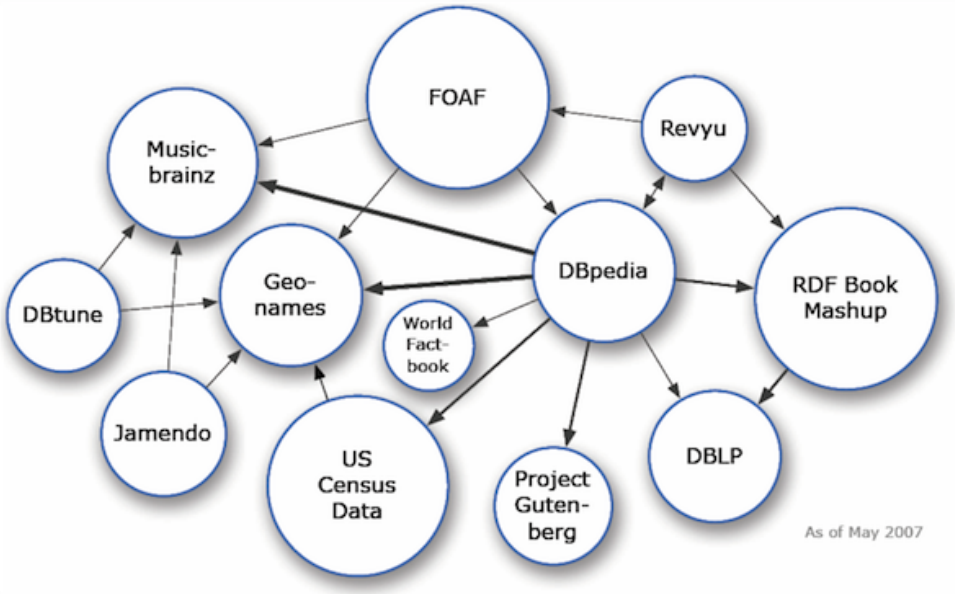

Figure 2.2: The LOD cloud as of May, 2007

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 **??**) to reach today thousands of datasets covering knowledge from various domains such as encyclopedic, government, geographic, entertainment and so on. The datasets have tripled in size from 2011 to 2014, with a significant growth of nearly 271% [**?**]. The latest version published in April 2014 contains 1014 linked datasets connected by 2909 linksets (see Figure **??**).

One of the most widely used datasets is DBpedia[6]. It is a structured knowledge extracted from multilingual versions of Wikipedia [**?**]. 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.

2.2 Open Data

Open Data (OD) is the data that can be easily discovered, accessed, reused and redistributed by anyone [**?**]. Open data has both legal and technical dimensions. It

[6]http://dbpedia.org

is placed in the public domain under liberal terms of use with minimal restrictions and is available in electronic formats that are non-proprietary and machine readable.

Businesses, citizens and governments are encouraged to publish, share and reuse data. Figure **??** shows the Open Data ecosystem described by [**?**]. 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.

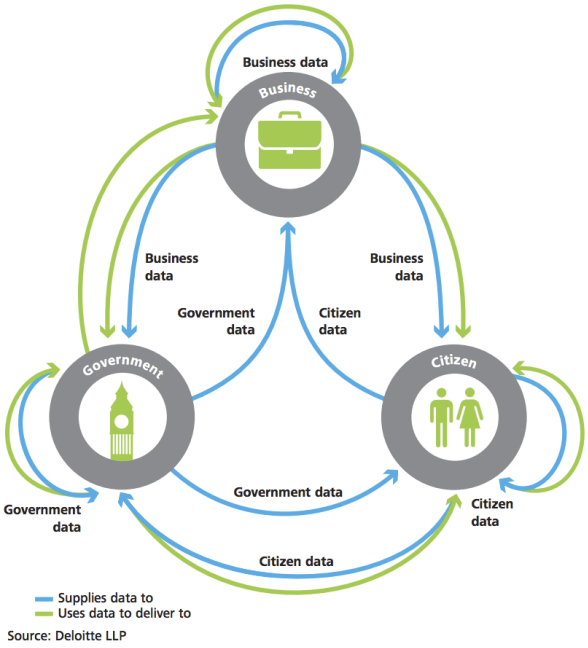

Figure 2.3: Open Data ecosystem

Linked Open Data refers to the semantically linked, machine-readable open data. Tim Berners-Lee [**?**] 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; allowing citizens to be better informed about policies, public spending and activities in the law making processes [**?**, **?**].

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 [**?**]. Despite the legal issues surrounding Open Data licenses [**?**], McKinsey [**?**] 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 **??** 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 [**?**]. Moreover, there are several reports and initiatives like Open Data Barometer[7], Open Data Monitor[8] and Global Open Data Index[9]that aim at analyzing and monitoring the adoption of Open Data across the world.

Going back to our scenario in **??**, Open Data will help our analyst **Dan** in:

- Having 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.

- Discovering 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, **Dan** could draw more accurate conclusions on the root cause of some of these accidents.

2.2.1 Open Licenses

Project Open Data [10] 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. Figure **??** shows the LOD cloud datasets licenses distribution. We notice that a considerable amount of datasets are still missing attached license information.

[7] http://barometer.opendataresearch.org/
[8] http://opendatamonitor.eu
[9] http://index.okfn.org/
[10] https://project-open-data.cio.gov

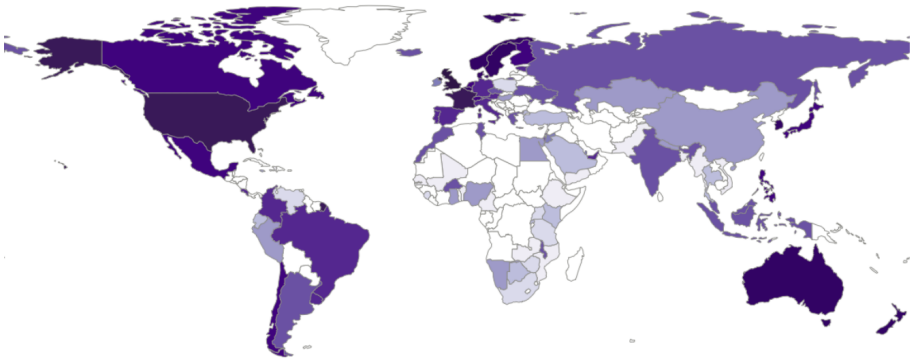

Figure 2.4: Heat map of Open Government Data adoption according to the Open Data Barometer 2015

The Open Definition [11] 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:

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

Despite the legal issues surrounding Linked Data licenses [**?**], 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 [**?**]. In [**?**], 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.

[11] http://opendefinition.org/

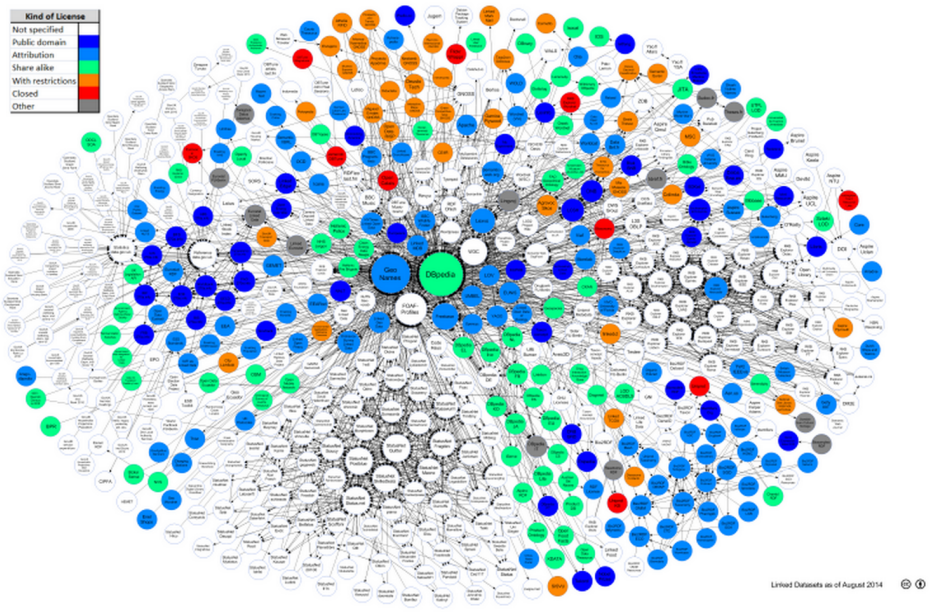

Figure 2.5: Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak colored by licensing types. http://www.cosasbuenas.es/blog/how-o-is-lod-2015

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 [**?**, **?**].

Data profiling is a vital task to monitor the quality of internal data in the enterprise. Halo BI report [**?**] 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 detecting irregularities to improve its quality. Data profiling includes typically several tasks:

- **Metadata profiling**: Provides general information on the dataset (dataset description, release and latest 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 [**?**].

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 [**?**] 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 [**?**]. Having rich metadata helps in enabling:

- **Data discovery, exploration and reuse**: In [**?**], 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 [**?**]. 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 Conclusion

In this chapter, we set up the grounds for the rest of this part of the thesis. We presented that basic concepts in semantic Web, open and linked data as well as data profiling and its subtasks.

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

Data portals can be public like `Datahub.io` and `publicdata.eu` or private 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[1] 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[2] is a standalone Drupal distribution that is used in various public data portals as well. In addition to these tradition data portals, there is a set of tools that allow exposing data directly as RESTful APIs like `thedatatank.com`.

3.1 Data Management Systems 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 Management Software. In this section, we present the results of our landscape survey of the most common data management systems and dataset models.

[1] http://ckan.org
[2] http://nucivic.com/dkan/

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 [**?**]. 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)[3] is a specialization of DCAT to describe public sector 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[4] to be the standard for describing datasets and catalogs in Europe.

3.1.3 Dataset Usage Vocabulary

The Dataset Usage Vocabulary (DUV) [**?**] focuses on capturing the experience of using datasets. Publishers often lack feedback on how their datasets are being used and consumers lack an effective method to communicate their experiences. DUV basically aims at filling these gaps by describing consumers experiences, citations and feedback about a dataset.

3.1.4 ADMS

The Asset Description Metadata Schema (ADMS) [**?**] 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.

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

3.1.5 VoID

VoID [?] 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.

3.1.6 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[5].

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[6] index is built on top of this metadata to enable search and filtering.

The CKAN data model[7] 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[8] provides plugins that allow CKAN to expose and consume metadata from other catalogs using DCAT as their model.

The Open Data Companion Kit[9] is a mobile application the provides a unified data access point for over 100 of open data portals. The application basically aims at CKAN-based portals providing a unique experience to mobile users.

[5] http://data.gov
[6] http://lucene.apache.org/solr/
[7] http://docs.ckan.org/en/ckan-1.8/domain-model.html
[8] https://github.com/ckan/ckanext-dcat
[9] http://www.socrata.com/open-data-field-guide/open-data-field-kit/

3.1.7 DKAN

DKAN[10] 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 datasets 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[11] is very similar to the CKAN one, containing information to describe datasets, resources, groups and tags.

3.1.8 Socrata

Socrata[12] 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.

3.1.9 Junar

Junar[13] 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.

3.1.10 INSPIRE metadata

The Infrastructure for Spatial Information in the European Community directive (INSPIRE)[14] aims at ensuring a compatible and usable spatial data infrastructure across the European Union.

[10] http://nucivic.com/dkan/
[11] http://docs.getdkan.com/dkan-documentation/dkan-developers/dataset-technical-field-reference/
[12] http://socrata.com
[13] http://junar.com/
[14] http://inspire.ec.europa.eu/index.cfm

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. For metadata schema, these include rules for the description of data sets, which could be adopted by open data publishers.

3.1.11 Schema.org

Schema.org[15] 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.

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.

3.1.12 Common Core Metadaa Schema (CCMS)

Project Open Data (POD)[16] 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)[17] 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 Model Classification

A dataset metadata model must 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 **??**, 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.

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

- **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.

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[18].

- **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 rights, e.g., Linked Data Rights[19], the Open Digital Rights Language (ODRL)[20].

- **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)[21] for people and relationships, vCard [**?**] for people and organizations and the Organization ontology [**?**] 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

[18] http://dublincore.org/documents/dcmi-terms/
[19] http://oeg-dev.dia.fi.upm.es/licensius/static/ldr/
[20] http://www.w3.org/ns/odrl/2/
[21] http://xmlns.com/foaf/spec/

lead to the development of various special vocabularies like the Open Provenance Model[22] and PROV-O [?]. DataID [?] 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[23] aims at establishing an infrastructure for spatial information. Mappings have been made between DCAT-AP and the INSPIRE metadata. CKAN provides as well a spatial extension[24] 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 [?] 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 [?]. 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

[22] http://open-biomed.sourceforge.net/opmv/
[23] http://inspire.ec.europa.eu/
[24] https://github.com/ckan/ckanext-spatial

daQ [?] that can be used to express datasets quality. The RDF Review Vocabulary[25] can also be used to express reviews and ratings about the dataset or its resources.

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

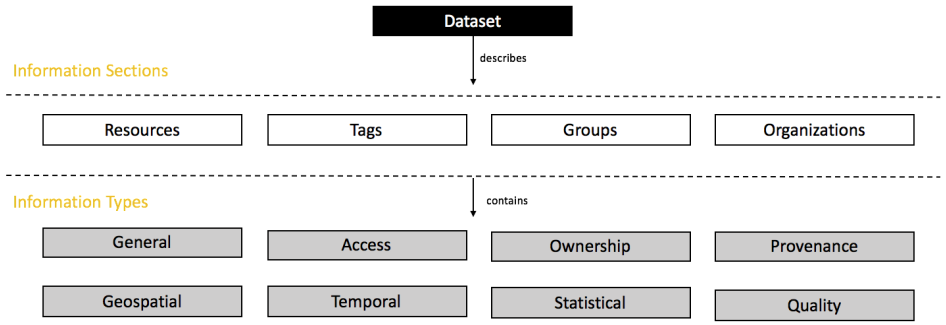

Figure 3.1: Information sections and groups across data models

3.3 Mapping Metadata Models

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 all the models and vocabularies specifications and documentations.
- Examine existing datasets using these models and vocabularies. Data Portals[26] 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

[25] http://vocab.org/review/
[26] http://dataportals.org

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	-	-

Table 3.1: Data models sections mapping

dataset's metadata. As a consequence, we had to investigate the Socrata Open Data API (SODA) source code[27] and check the different classes and interfaces.

The first task is to map the four main information sections (resources, tags, groups and organization) across those models. Table **??** 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 **??** presents the full mappings between the models across the information groups. Entries in the CKAN marked with ∗ are properties from CKAN extensions and are 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`.

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

Table 3.2: Harmonized Dataset Models Mappings

Data Model	CKAN	DKAN	POD	DCAT	VoID	Schema.org	Socrata
General Information	id			id		identifier	dcat:Dataset→ dct:identifier
	private			private		accessLevel	
	state			state			
	type			type			
	name			name			
	isopen						
	notes			notes		description	dcat:Dataset→ dct:description
	title			title		title	dcat:Dataset→ dct:title
	num_resources						
	num_tags						
						conformsTo	dcat:Dataset→ dct:conformsTo
						language	dcat:Dataset→ dct:language
						accuralPeriodicity	dcat:Dataset→ dct:accuralPeriodicity
access information	license_title			license_title		license	dcat:Distribution→ dct:license
	license_id						
	license_url						
	url			url		landingPage	dcat:Dataset→ dcat:landingPage
						rights	dcat:Distribution→ dct:rights
	attribution_text*						
provenance	version						
	revision_id						
	metadata_created			metadata_created			dcat:Distribution→ dct:created
	metadata_modified			metadata_modified		modified	dcat:Distribution→ dct:modified
	revision_timestamp			revision_timestamp			
						issued	dcat:Distribution→ dct:issued
						temporal	dcat:Dataset→ dct:temporal
ownership	maintainer			maintainer		contactPoint→ fn	dcat:Dataset→ dcat:contactPoint→ vcard:fn
	maintainer_email			maintainer_email		contactPoint→ hasEmail	dcat:Dataset→ dcat:contactPoint→ vcard:hasEmail
	owner_org						
	author						dcat:Dataset→ dct:creator→ foaf:Person:givenName
	author_email			author_email			dcat:Dataset→ dct:creator→ foaf:Person:mbox
						bureauCode	
						programCode	
	description						
						isPartOf	
						systemOfRecords	
						describedBy	
						describedByType	
GeoSpatial	spatial-text*					spatial	dcat:Dataset→ dct:spatial
	geographical_granularity*						

Data Model	VoID	Schema.org	Socrata
General Information	void:Dataset→ dct:description	Thing:description	description
	void:Dataset→ dc:title	Thing:name	name
	void:Dataset→ void:documents		
	void:Dataset→ dct:conformsTo		
	void:Dataset→ dct:language	CreativeWork:inLanguage	
	void:Dataset→ dct:accuralPeriodicity		
access information	void:Dataset→ dct:license		license→ name
			licenseId
		CreativeWork:license	license → termsLink
		Thing:url	
	void:Dataset→ dct:rights		
			attribution
			attributionLink
provenance		CreativeWork:version	
	void:Dataset→ dct:created	CreativeWork:dateCreated	
	void:Dataset→ dct:modified	CreativeWork:dateModified	
	void:Dataset→ dct:issued	CreativeWork:datePublished	
	void:Dataset→ dct:temporal	Dataset:temporal	
ownership		CreativeWork:producer→ Thing:name	owner→ displayName / owner→ ScreenName
		CreativeWork:producer→ Person:email	
		CreativeWork:sourceOrganization:LegalName	
	void:Dataset→ dct:creator→ foaf:Person:givenName	CreativeWork:author→ Thing:name	
	void:Dataset→ dct:creator→ foaf:Person:mbox	CreativeWork:author→ Person:email	
		CreativeWork:sourceOrganization→ Thing:description	
		CreativeWork:isPartOf	
		CreativeWork:hasPart	
GeoSpatial	void:Dataset→ dct:spatial	Dataset:spatial	
			bbox
			layers

Table 3.2 Harmonized Dataset Models Mappings

Data Model	CKAN	DKAN	POD	DCAT	VoID	Schema.org	Socrata
							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	
			accessURL	dcat:Distribution→ dcat:accessURL		Dataset:distribution→ DataDownload→ MediaObject:contentUrl	accessPoints
	webstore_url						
provenance	cache_last_updated						
	revision_timestamp	revision_timestamp					
	webstore_last_updated						

Table 3.2 Harmonized Dataset Models Mappings

Data Model	CKAN	DKAN	POD	DCAT	VoID	Schema.org	Socrata
	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 Towards A Harmonized Model (HDL)

Examining the different models and their mappings in Table **??**, 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. In this section, we present HDL, a harmonized dataset model that aims at filling this gap by taking the best from these models.

In addition to the core dataset metadata, HDL describes the four common sections of datasets described in Section **??** (see Figure **??**).

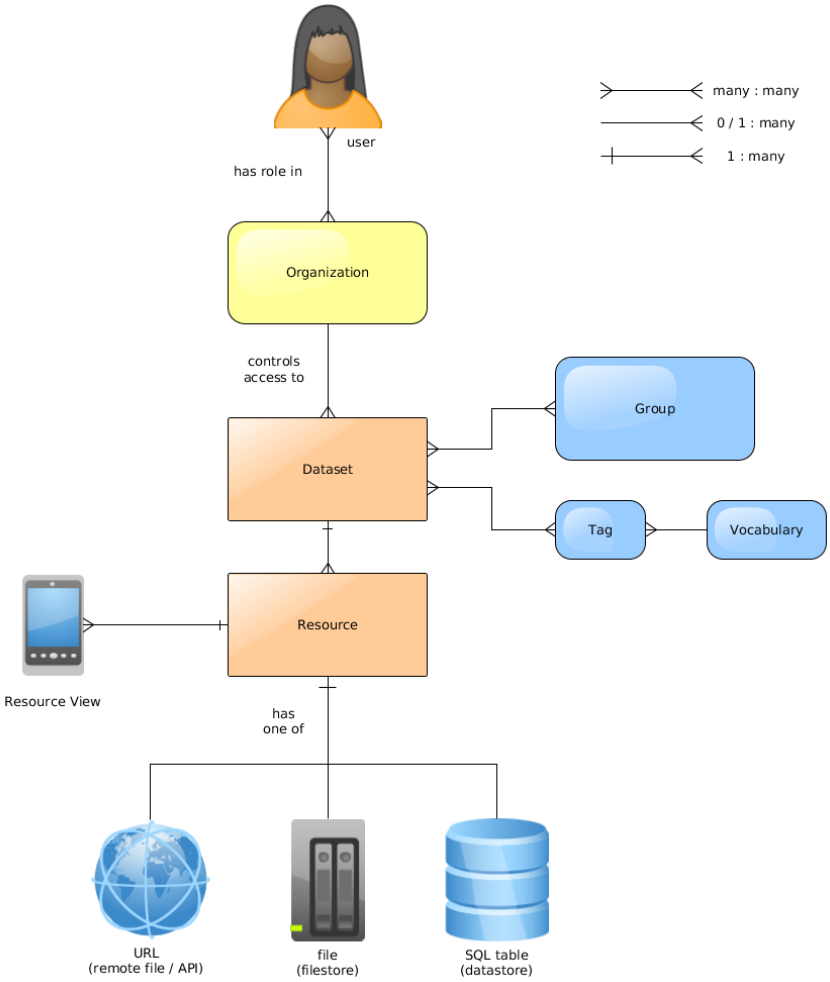

Figure 3.2: CKAN data model

Table **??** describes the required fields across all the sections of a dataset and its core metadata. For example, a dataset resource, group, organization as well as to

the dataset itself will have an `id`, `name`, etc.

Field	Label	Description	Required
id	Unique Identifier	A dataset unique identification	Yes
name	Name	Machine-readable name of the asset	Yes
title	Title	Human-readable name of the asset. Should be in plain English and include sufficient detail to facilitate search and discovery	Yes
description	Description	Human-readable description (e.g., an abstract) with sufficient detail to enable a user to quickly understand whether the asset is of interest	Yes
created	Creation Date	Date on which the dataset was created	Yes
modified	Last Modification Date	Most recent date on which the dataset was changed, updated or modified	Yes

Table 3.3: Common required metadata fields for all the datasets sections

Table **??** describes the authorship information that can be included in different sections. For example, a group has a required `administrator` field. A group administrator inherits all the fields mentioned in this table, meaning that he must have an `id, name, email` and an optional `role` within the organization.

Field	Label	Description	Required
id	Unique Identifier	A person unique identification	Yes
name	Name	Human-readable name of the person	Yes
email	E-mail	A valid electronic mail address for the person	Yes
role	Role	Human-readable name of the asset. Should be in plain English and include sufficient detail to facilitate search and discovery	No

Table 3.4: Metadata fields for ownership information

3.4.1 Resources

Resources are the main data containers of a dataset, they are a vital part of the dataset metadata as they are the facade on which users will interact with. Many of the core dataset metadata as we will see in Section **??** have an aggregate value of some resources fields. In addition to the common core metadata field described in Table **??**, Table **??** described the resources metadata fields.

Field	Label	Description	Required
type	Type	The human-readable format of the resource	Yes
download_url	Download URL	URL providing direct access to a resource, for example via API or a graphical interface	Yes
access_url	Access URL	URL providing indirect access to a resource. For example, the Web page on which the `download_url` is available at	Yes
format	Format	A human-readable description of the file format of a distribution	Yes
hash	Hash	Automatically generated unique md5 or sha-1 hash. Mainly used for indexing purposes.	Yes
state	State	The state of the current resource e.g. published, draft, under revision	Yes
access_level	Access Level	The degree to which this resource could be made publicly-available, e.g., public, restricted public, private	Yes
mimetype	MIME-type	Machine-readable file format that conforms to the IANA Media Types [28]	Yes
size	Size	Actual size (content-length) of the resource in bytes	Yes
described_by	Described By	URL to the data dictionary for the distribution found at the `download_url`	Yes
conforms_to	Conforms To	URI used to identify a standardized specification the distribution conforms to	No
rating	Rating	Normalized score of the resource rating by users	Yes
data_quality	Data Quality	The resource objective quality score	Yes
cache_url	Cache URL	A URL of the resource cached version (used for portals with build in cloud storage)	Yes
temporal_granularity	Temporal Granulairty	The detail levels associated with the temporal information of the dataset	If-Applicable
temporal_coverage_from	Temporal Coverage Starting Range	Start date of applicability for the data	If-Applicable
temporal_coverage_to	Temporal Coverage End Range	End date of applicability for the data	If-Applicable
spatial_text	Spatial Text	A textual information about the range of spatial applicability of a dataset. e.g., named place like London, United Kingdom.	If-Applicable
spatial_granularity	Spatial Granularity	The detail levels associated with the spatial coverage of the dataset	If-Applicable
bbox	Bounding Box	An area defined by two longitudes and two latitudes e.g., -0.489—51.28—0.236—51.686	If-Applicable
layers	Layers	A slice of the geographic coverage in a particular area. For example, on a road map roads, national parks, and rivers might be considered as different layers.	If-Applicable
cache_modified	Cache Modified	Most recent date on which the resource cache was changed, updated or modified	Yes
revision_id	Revision ID	Latest revision ID for the resource	Yes

Continued on next page

[28] http://www.iana.org/assignments/media-types/media-types.xhtml

Table 3.5 Metadata fields for resources information section

Field	Label	Description	Required
revision_timestamp	Revision Timestamp	Latest timestamp for the resource revision	Yes
license_id	License ID	The normalized license ID with which the resource has been published. If the license is open, the ID should conform to one available at https://github.com/okfn/licenses	Yes
license_title	License Title	The normalized human-readable title of the resource license. If the license is open, the title should conform to one available at https://github.com/okfn/licenses	Yes
license_url	License URL	The normalized URL of the resource license. If the license is open, the URL should conform to one available at https://github.com/okfn/licenses	Yes
attribution_text	Attribution Text	The attribution text that should be inserted based on the accompanying license guidelines if applicable.,The text is provided by the original author.	If-Applicable
attribution_link	Attribution Link	The attribution link to the original source if applicable	If-Applicable
rights	Rights	Information regarding access or restrictions based on privacy, security, or other policies. If the access is restricted, should also include information on how to ask for access information.	Yes

Table 3.5: Metadata fields for resources information section

3.4.2 Groups

In addition to the metadata fields in Table **??**, a group must also include information about an author in an `administrator` field. This means that he inherits all the fields mentioned in Table **??**. In addition to that, a group can be part of a larger group, thus a `subGroupOf` field is required when applicable to denote the `id` of the parent group.

3.4.3 Tags

One extra field is required in addition to those mentioned in Table **??** which is `vocabulary_id`. This fields represents a unique identifier referring to the vocabulary (if used) controlling the tag. For example, if a dataset defines a geographical coverage, then a possible tag vocabulary would be to add a `Country Code` field with values such as `en, fr, ar`, etc. This field is optional, however, its existence enforce restrictions and provide semantic grouping and clustering of datasets in portals.

3.4.4 Organization

Table ?? describes the required field to describe the organization information section in addition to those in Table ??. Those fields are mainly inspired by the Organization Ontology [?].

Field	Label	Description	Required
sub_organization_of	Sub Organization Of	Represents hierarchical containment of organizations by Indicating if an organization is a sub-part or child of another organization	If-Applicable
based_at	Based At	Indicates the site at which an organization is based. This does not restrict the possibility for an organization to be at multiple sites	Yes
has_site	Has Site	human-readable address for the company's site	
location	Location	Location description for the organization e.g. lat, long coordinates	Yes

Table 3.6: Metadata fields for organization information section

3.4.5 Core Metadata

In addition to the common metadata fields described in Table ??, Table ?? describes the core metadata fields of every dataset. In addition to those, two authorship related fields are also required: `maintainer` and `owner`. Both fields inherit the authorship properties described in Table ??.

Field	Label	Description	Required
access	Download URL	URL providing direct access to a dataset, for example via API or a graphical interface. The access method should aggregate all the dataset resources available.	Yes
access_url	Access URL	URL providing indirect access to a dataset. For example, the Web page on which the `download_url` is available at	Yes
state	State	The state of the current dataset e.g. published, draft, under revision	Yes
access_level	Access Level	The degree to which this dataset could be made publicly-available, e.g., public, restricted public, private	Yes
rating	Rating	Normalized score of the average resources rating	Yes
data_quality	Data Quality	The average quality score of the dataset resources	Yes
revision_id	Revision ID	Latest revision ID for the resource	Yes
revision_timestamp	Revision Timestamp	Latest timestamp for the resource revision	Yes
license_id	License ID	The normalised license ID(s) with which the dataset resources has been published. If the license is open, the ID should conform to one available at https://github.com/okfn/licenses	Yes

Continued on next page

Table 3.7 Dataset core metadata fields

Field	Label	Description	Required
license_title	License Title	The normalised human-readable title(s) of the dataset resources licenses. If the license is open, the title should conform to one available at https://github.com/okfn/licenses	Yes
license_url	License URL	The normalised URL of the license used. If the license is open, the URL should conform to one available at https://github.com/okfn/licenses	Yes
attribution_text	Attribution Text	The attribution text that should be inserted based on the accompanying license guidelines if applicable.,The text is provided by the original author.	If-Applicable
attribution_link	Attribution Link	The attribution link to the original source if applicable	If-Applicable
rights	Rights	An aggregate information regarding the dataset access or restrictions based on privacy, security, or other policies. If the access is restricted, should also include information on how to ask for access information.	Yes
language	Language	The aggregate set of languages used in the dataset resources	Yes
language_code	Language Code	The aggregate set of machine-readable language codes used in the dataset resources, e.g., `en, fr`	Yes
metadata_created	Metadata Creation Date	The creation date of the dataset metadata	Yes
metadata_modified	Metadata Modification Date	Most recent date on which the dataset metadata was changed, updated or modified	Yes
is_part_of	Is Part of	The unique identifier of a dataset of which the dataset is a subset	Yes
has_part	Has Part	The unique identifier of a dataset which is a part of the current dataset	Yes
number_of_resources	Number of Resources	Total number of resources for the dataset	Yes
number_of_tags	Number of Tags	Total number of tags for the dataset	Yes

Table 3.7: Dataset core metadata fields

3.4.6 Controlling Field Values

Various models control the set of values used to describe some of the model's properties. For example, CKAN model controls values for the `resource_type` property and restrict them to: file: `direct accessible bitstream`, `file.upload`, `api`, `visualization`, `code` and `documentation`. However, dataset publishers do not always conform to these predefined values and can add additional values. In order to know the set of values in these fields we examined the models of several CKAN datasets with a tool called Roomba. Roomba is a scalable automatic approach for

extracting, validating, correcting and generating descriptive linked dataset profiles (see Chapter **??**).

We created two main reports with Roomba. The first aims to list the file types specified for resources using the query string `resources>resource_type:resources>name` (see Listing **??**) and the second one to collect the list of `extras` values using the query string `extras>key:extras>value` (see Listing **??** and Listing **??**). 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[29] that contains 1653 datasets.

```
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 *extras* aggregation report for the LOD Cloud

```
access_constraints with total count of: 890
bbox-east-long with total count of: 890
bbox-west-long with total count of: 890
spatial with total count of: 890
spatial-data-service-type with total count of: 890
spatial-reference-system with total count of: 890
```

Listing 3.2: Excerpt of the *extras* field aggregation report for OpenAfrica portal

```
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 3.3: 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

[29] http://africaopendata.org/

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`).

At the moment, HDL does not control the metadata field values. However, restricting those values to a finite set as shown above pave the way to achieve better data harmonization across portals.

3.5 Summary

Data models vary across data portals. In this chapter, we surveyed the landscape of various models and vocabularies that described datasets on the web. As a result, 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) [**?**] which prohibits a semantically rich representation of complex datasets like DBpedia[30] that has multiple endpoints and thousands of dump files with content in several languages [**?**].

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.

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, a harmonized dataset model, that takes the best out of these models to ensure complete metadata coverage to enable data discovery, exploration and reuse.

[30] http://dbpedia.org

Chapter 4

Dataset Profiles Generation and Validation

4.1 Introduction

The heterogeneous nature of data sources reflects directly on the data quality as they often contain inconsistent as well as misinterpreted and incomplete metadata information. Moreover, the significant variation in size, formats and freshness of the data, makes it more difficult to find useful datasets without prior knowledge. This can be clearly noticed in the LOD Cloud where few datasets such as DBPedia [**?**], Freebase [**?**] and YAGO [**?**] 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].

Users explore datasets in data portals relying on the metadata information attached by either the dataset owner or the data portal administrator. This information is mainly in form of predefined tags such as *media, geography, life sciences* that are used for organization and clustering purposes. However, the increasing diversity of those datasets makes it harder to classify them in a fixed number of tags that are subjectively assigned without capturing the essence and breadth of the dataset [**?**]. Furthermore, the increasing number of datasets available makes the manual review and curation of metadata 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

[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

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 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, ignoring the tasks of statistical and topical profiling. We validate our framework against a manually created set of profiles and manually check the 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 [**?**]. 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 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 [**?**].

- **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 [**?**] 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) [?] and the Vocabulary of Interlinked Datasets (VoID) [?] are concerned with metadata about RDF datasets. There exist several tools aiming at exposing dataset metadata using these vocabularies. In [?], 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[7] 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 answers 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[8], 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 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[9] 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[10] 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.

[7] http://linkeddata.informatik.hu-berlin.de/LDSrcAss/datenquelle.php
[8] https://certificates.theodi.org/
[9] http://labs.data.gov/dashboard/
[10] http://validator.lod-cloud.net/

Statistical profiling: Calculating statistical information on datasets is vital to applications dealing with query optimization and answering, data cleansing, schema induction and data mining [?, ?, ?]. Semantic sitemaps [?] and RDFStats [?] are one of the first to deal with RDF data statistics and summaries. ExpLOD [?] creates statistics on the interlinking between datasets based on `owl:sameAs` links. In [?], the author introduces a tool that induces the actual schema of the data and gathers corresponding statistics accordingly. LODStats [?] is a stream-based approach that calculates more general dataset statistics. ProLOD++ [?] is a Web-based tool that allows LOD analysis via automatically computed hierarchical clustering [?]. Aether [?] generates VoID statistical descriptions of RDF datasets. It also provides a Web interface to view and compare VoID descriptions. LODOP [?] 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 [?] 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 [?] is a general entity annotation framework that provides machine processable output allowing efficient querying. In addition, there exist several entity annotation tools and frameworks [?] but none of those systems are designed specifically for dataset annotation. In [?], the authors created a semantic portal to manually annotate and publish metadata about both LOD and non-RDF datasets. In [?], 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 [?], the authors extracted dataset topics by exploiting the graph structure and ontological information, thus removing the dependency on textual labels. In [?], 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 [?], the authors used VoID descriptions to optimize query processing by determining relevant query-able datasets. In [?], the authors created an approximate index structure (QTree) and an algorithm for answering conjunctive queries over Linked Data. SchemEX [?] is a stream-based approach leveraging type and property information of RDF instances to create schema-level indexes.

Semantic search engines like Sindice [?], Swoogle [?] and Watson [?] help in entities

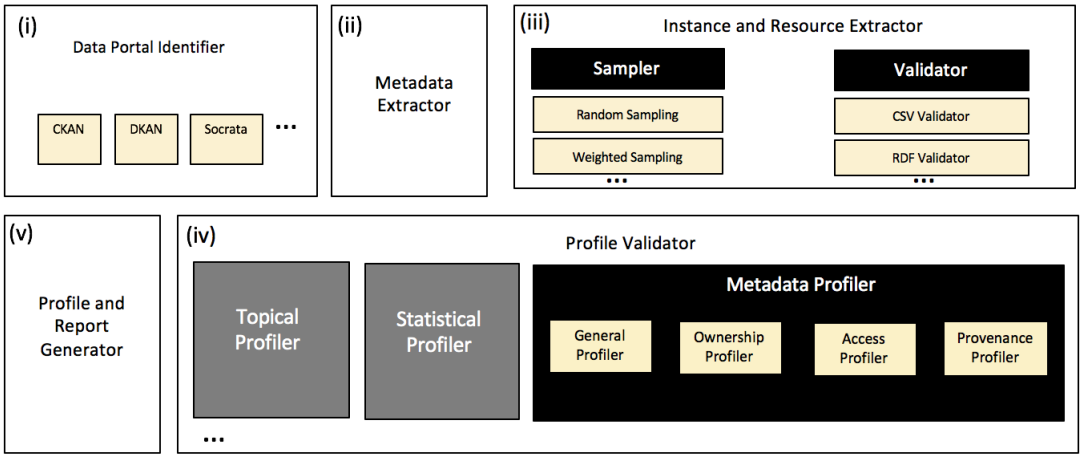

Figure 4.1: Processing pipeline for validating and generating dataset profiles

lookup but they are not designed specifically for dataset search. In [**?**], the authors utilized the sig.ma index [**?**] 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 **??** 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 (see Figure **??**) using Node.js and is available on the tools Github repository[11]. Roomba allows data portal administrators like **Dan** to:

- Fetch information about the portal's data management system

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

- Fetch all the information about datasets from a data portal
- Fetch all the groups information from a data portal
- Crawl, fetch and cache 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

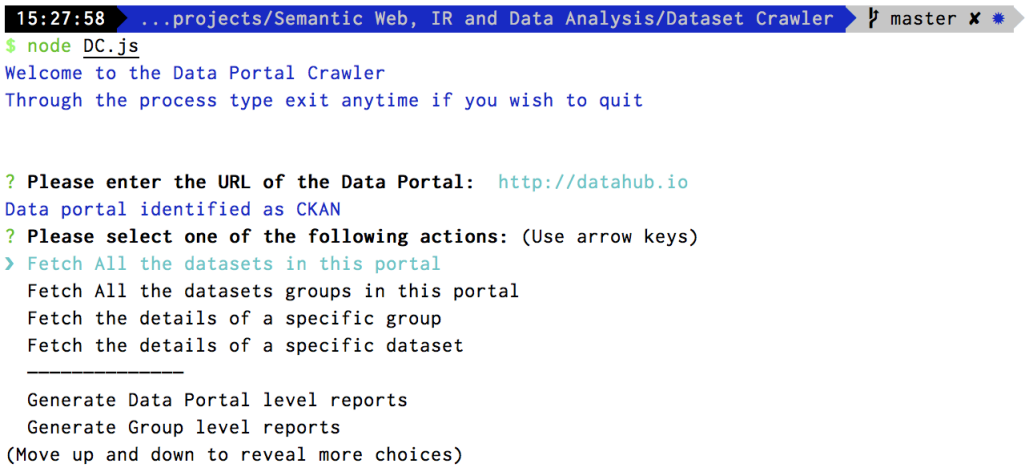

Figure 4.2: Screenshot for Roomba command line tool

Appendix **??** details the instructions for installing and running the framework. The various steps are explained in detail below.

4.4.1 Data Management System Identification

Data portals are considered to be data access points providing tools to facilitate data publishing, sharing, searching and visualization. Section **??** highlights the main data management systems powering those data portals and the various dataset models used. In addition to these traditional data management systems, there is a set of tools that allow exposing data directly as RESTful APIs like Datatank[12] and Database-to-API[13].

Roomba is 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 step. The

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

Data Portal Identifier (component (i)) relies 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. The Data Portal Identifier uses 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, the Data Portal Identifier checks the existence of certain DOM elements or properties. For example, CKAN-powered portals 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, the Data Portal Identifier issues a query to 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 **??**) must contain information about the dataset's title, description, maintainer email, update and creation date, etc.

Since Roomba operates on CKAN-based data portals, the Metadata Extractor (component (ii)) validates the extracted metadata against the CKAN standard

model[14] (see Listing **??**).


```
{
 "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, The Metadata Extractor performs 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, The Metadata Extractor can issue specific extraction jobs. For example, in CKAN-based portals, The Metadata Extractor is 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, the Instance and Resource Extractor (component (iii)) is 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), the extractor performs 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 MIME-type, name, description, format, valid dereferenceable URL, size, type and provenance. The validation process issues an HTTP request to the resource and automatically fills up various missing

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

information when possible, like the MIME-type 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[15] for CSV files and n3[16] 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 submodule is introduced to execute various sample-based strategies as they were found to generate accurate results even with comparably small sample size of 10% [**?**]. The sampling strategies introduced are:

- **Random Sampling**: Randomly selects resources instances.
- **Weighted Sampling**: Weighs each resource 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 that we offer by default. Strategies like those introduced in [**?**] 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 **??** describes an extension to measure the objective qualities of datasets).

The Profile Validator (component (iv)) identifies missing information and the ability to automatically correct them. Each set of metadata (general, access, ownership

[15] https://github.com/wdavidw/node-csv
[16] https://github.com/RubenVerborgh/N3.js

and 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, the Profile Validator issues an HTTP `HEAD` request in order to check if that URL is reachable. The Profile Validator also uses the information contained in a valid `content-header` response to extract, compare and correct some resources metadata values like `mimetype` and `size`.

Having valid license information is vital for organization looking to integrate external data. However, from our experiments, we found out that datasets' license information is often missing or noisy. The license names if found are not standardized. For example, `Creative Commons CCZero` can also be `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 **??**). In addition, we have also used the open source and knowledge license information[17] to normalize the license information and add extra metadata like the domain, maintainer and open data conformance. The Profile Validator uses this mapping file to valdiate and normalize datasets license information.

```
{
        "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

4.4.5 Profile and Report Generation

The validation process highlights the missing information and presents them in a human readable report (see appendix **??**). 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

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

and are publicly available[18].

```
=============================================================
                        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

Data portal administrators like **Paul** 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

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

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 CKAN-based data portals. The first is the Datahub targeting specifically the LOD cloud group. The current state of the LOD cloud report [**?**] indicates that the LOD cloud contains 1014 datasets. They were harvested via an LDSpider crawler [**?**] 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 as they contained more recent and accurate metadata. 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. We chose the Amsterdam data portal [19] as it is updated frequently and highly maintained. 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.

The experiments were executed on a 2.6 Ghz Intel Core i7 processor with 16GB of DDR3 memory machine. The approximate execution time alongside the summary of the datasets' properties are presented in Table **??**.

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.

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

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 **??**.

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 **??** 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 individual 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[20]. 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.

Figures **??** and **??** 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 **??** shows the top metadata fields errors for each metadata information 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

[20] 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	cache_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

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 ex-

ample, 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%) are metadata and one (0.029%) is example and documentation types.

To have more details about the resources URL types, we created a $key : objectmeta-fieldvalues$ group level report on the LOD cloud with `resources> format:title`. This aggregates the resources format information for each dataset. We observe that only 161 (62.16%) of the datasets valid URLs have SPARQL endpoints 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.

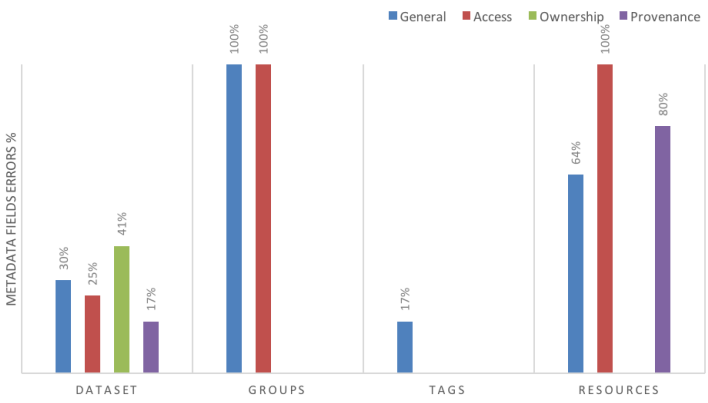

Figure 4.3: LOD Cloud error % by section

4.6.3 Ownership Information

Ownership information is divided into direct ownership (author and maintainer) and organization information. Four fields (66.66%) of the direct ownership information 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 MIME-type 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 related to ambiguous entries. To resolve that, we manually created a mapping file[21] standardizing the set of possible license names and urls using the open source and knowledge license information[22]. 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%.

4.7 Summary

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

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

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

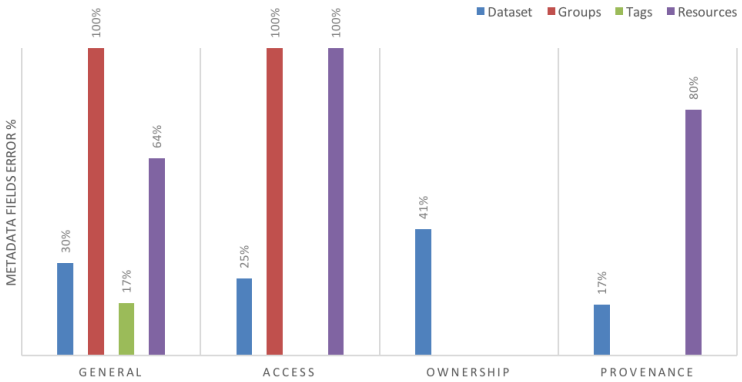

Figure 4.4: LOD Cloud error % by information type

in order to check the validity of the metadata provided and to generate descriptive and statistical information for a particular dataset or for an entire data portal.

It has been noticed that the issues surrounding metadata quality 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 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 [?]. 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 [?, ?, ?]. 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 [?], 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 [?]. 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 [?]. The dimensionality of data quality makes it dependent on the task and users requirements. For example, DBpedia [?] and YAGO [?] are knowledge bases containing data extracted from structured and semi-structured sources. They are used in a variety of applications e.g., annotation systems [?], exploratory search [?] and recommendation engines [?]. However, their data is not integrated into critical systems e.g., life critical (e.g., medical applications) or safety critical (e.g., aviation applications) as its data quality is found to be

insufficient.

In this chapter, we first propose a comprehensive objective framework to evaluate the quality of Linked Data sources. The framework is based on a refinement of the data quality principles described in [?] and surveyed in [?]. Some attributes have been grouped for more detailed quality assessments while we have also extended them by adding for each attribute a set of objective indicators. These indicators are measures that provide users with quality metrics measurable by tools regardless of the use case. For example, when measuring the quality of DBpedia dataset, an objective metric would be the availability of human or machine readable license information rather than the trustworthiness of the publishers.

Secondly, we survey the landscape of Linked Data quality tools to discover that they only cover a subset of the proposed objective quality indicators. As a result, we extend Roomba (see Chapter ??) with an extensible quality measurement tool. This tool helps on one hand data owners to rate the quality of their dataset and get some hints on possible improvements, and on the other hand data consumers to choose their data sources from a ranked set.

5.2 Data Quality Assessment

In [?], 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 (see Section ??) comes as an online and free questionnaire for data publishers focusing on certain characteristics about their data. The questions are classified into the following categories: general information (about dataset, publisher and type of release), legal information (e.g., rights to publish), licensing, privacy (e.g., whether individuals can be identified), practical information (e.g., how to reach the data), quality, reliability, technical information (e.g., format and type of data) and social information (e.g., contacts, communities, etc.). Based on the information

provided by the data publisher, a certificate is created with one of four different ratings.

Although ODI is a great initiative, the issued certificates are self-certified. ODI does not verify or review submissions but retains the right to revoke a certificate at any time. At the time of writing this paper, there was only 10,555 ODI certificates issued. The dynamicity of Linked Data makes it also very difficult to update the certificates manually, especially when these changes are frequent and affect multiple categories. There is clearly a need for automatic certification which can be supplemented with some manual input for categories that cannot be processed by machines.

The emerging critical need for large, distributed, heterogeneous, and complex structured datasets identified the necessity to establish industry cooperation between vendors of RDF and Graph database technologies in developing, endorsing, and publishing reliable and insightful benchmark results. The Linked Data Benchmark Council (LDBC)[1] aims to bridge the gap between the industry and the new trending stack of semantic technologies and their vendors. LDBC aims at promoting graph and RDF data management systems to be an accepted industrial solution. LDBC is not focused around measuring or assessing quality. However, it focuses on creating benchmarks to measure progress in scalability, storage, indexing and query optimization techniques to become the de facto standard for publishing performance results.

In [**?**], the authors propose a methodology for assessing Linked Data quality. It consists of three main steps: (1) requirement analysis, (2) quality assessment and (3) quality improvement. Considering the multidimensionality of data quality, the methodology requires users to provide the details of a use case or a scenario that describes the intended usage of the data. Moreover, quality issues identification is done with the help of a checklist. The user must have prior knowledge about the details of the data in order to fill this list. Tools implementing the proposed methodology should be able to generate comprehensive quality measures. However, they will require heavy manual intervention and deep knowledge on the data examined. These issues highly affect detecting quality issue on large scale.

Despite all the recent efforts in providing frameworks and tools for data quality in Linked Open Data, there is still no automatic framework for the objective assessment of Linked Data quality.

5.3 Objective Linked Data Quality Classification

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

[1] http://ldbc.eu/

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

- **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 [**?**], the authors identified 24 different Linked Data quality attributes. These attributes are a mix of objective and subjective measures that may not be derived automatically. In this paper, we refine these attributes into a condensed framework of 10 objective measures. Since these measures are rather abstract, we should rely on quality indicators that reflect data quality [**?**] 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 **??** 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 [?] 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 [?].

Table 5.1: Objective Linked Data quality framework

Quality Attribute	Quality Category	ID	Quality Indicator
Completeness	Dataset Level	1	Existence of supporting structured metadata [?]
		2	Supports multiple serializations [?]
		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 [?]
	Links Level	9	Existence of dereferencable links for the dataset [?, ?, ?]
	Model Level	10	Absence of disconnected graph clusters [?]
		11	Absence of omitted top concept [?]
		12	Has complete language coverage [?]
		13	Absence of unidirectional related concepts [?]
		14	Absence of missing labels [?]
		15	Absence of missing equivalent properties [?]
		16	Absence of missing inverse relationships [?]
		17	Absence of missing domain or range values in properties [?]
Availability	Dataset Level	18	Existence of an RDF dump that can be downloaded by users [?][?]
		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 [?]
		22	Existence of de-referenceable links to the full license information [?]
		23	Specifies permissions, copyrights and attributions [?]
Freshness	Dataset Level	24	Existence of timestamps that can keep track of its modifications [?]
Correctness	Dataset Level	25	Includes the correct MIME-type for the content [?]
		26	Includes the correct size for the content
		27	Absence of syntactic errors on the instance level [?]
	Links Level	28	Absence of syntactic errors [?]
		29	Use the HTTP URI scheme (avoid using URNs or DOIs) [?]
	Model Level	30	Contains marked top concepts [?]
		31	Absence of broader concepts for top concepts [?]
		32	Absence of missing or empty labels [?, ?]
		33	Absence of unprintable characters [?, ?] or extra white spaces in labels [?]
		34	Absence of incorrect data type for typed literals [?, ?]
		35	Absence of omitted or invalid languages tags [?, ?]
		36	Absence of terms without any associative or hierarchical relationships

Continued on next page

Table 5.1 Objective Linked Data quality framework

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

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 [?] and has documentation properties [?, ?]. 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 [?].
- Provides data in multiple serializations (N3, Turtle, etc.) [?].

- 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[2] or VOID[3].
- 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 [**?**].

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

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

5.3.3 Correctness

A dataset is considered to be correct if it includes the correct MIME-type and size for the content [**?**] and doesn't contain syntactic errors [**?**]. Links are considered to be correct if they lack syntactic errors and use the HTTP URI scheme (avoid using URNs or DOIs) [**?**]. 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) [**?**]. Moreover, if they don't contain incorrect data type for typed literals [**?**][**?**], no omitted or invalid languages tags [**?**, **?**], do not contain "orphan terms" (orphan terms are terms without any associative or

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

hierarchical relationships) and that labels are not empty, do not contain unprintable characters or extra white spaces [**?**, **?**, **?**].

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) [**?**, **?**], consistent preferred labels per language tag [**?**, **?**], atypical use of collections, containers and reification [**?**], wrong equivalent, symmetric or transitive relationships [**?**], consistent naming criteria in the model [**?**, **?**], overlapping labels in a given language for concepts in the same scheme [**?**] and membership violations for disjoint classes [**?**, **?**].

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 [**?**]. 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 [**?**].

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

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) [**?**]. A model is considered to be

comprehensible if there is no misuse of ontology annotations and that all the concepts are documented and annotated [?, ?].

5.3.9 Coherence

Coherence is the ability to interpret data as expected by the publisher or vocabulary maintainer [?]. 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 [?], blank nodes [?], deprecated classes or properties [?], relations and mappings clashes [?], invalid inverse-functional values [?], cyclic hierarchical relations [?, ?, ?], solely transitive related concepts [?], redefinitions of existing vocabularies [?] and valueless associative relations [?].

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 [?].

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

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

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

5.4.2 Modeling Quality

Reusing existing ontologies is a common practice that Linked Data publishers are always trying to adopt. However, ontologies and vocabularies development is often a long error-prone process especially when many contributors are working consecutively or collaboratively [?]. This can introduce deficiencies such as redundant concepts or conflicting relationships [?]. 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 [?] uses supervised machine learning techniques to learn concepts from user-provided examples. CROCUS [?] 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[10] [?] scans SKOS vocabularies to provide reports on vocabulary resources and relations that are problematic. PoolParty checker[11] is an online service based on qSKOS. Skosify [?] 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 [?] evaluates OWL ontologies against a rules catalog and provides the user with a set of guidelines to solve them. ASKOSI[12] 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 [?, ?] the authors perform quality checking on OWL ontologies using integrity constraints involving the Unique Name Assumption (UNA) and the Closed World Assumption (CWA). Pellet[13] provides reasoning services for OWL ontologies. It incorporates a

[10] https://github.com/cmader/qSKOS
[11] http://www.poolparty.biz/
[12] http://www.w3.org/2001/sw/wiki/ASKOSI
[13] http://clarkparsia.com/pellet

number of heuristics to detect and repair quality issues among disjoint properties, negative property assertions and reflexive, irreflexive, symmetric, and anti-symmetric properties. Eyeball[14] 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[15] provides validation for many issues highlighted in [**?**] like misplaced, undefined or deprecated classes or properties.

5.4.3 Dataset Quality

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

5.4.3.1 Manual Ranking Approaches

Sieve [**?**] is a framework for expressing quality assessment and fusion methods. It is implemented as a component of the Linked Data Integration Framework (LDIF)[16]. Sieve leverages the LDIF provenance metadata as quality indicators to produce quality assessment scores. However, despite its nice features, it is only targeted to perform data fusion based on user-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 [**?**] is a framework providing policies or formulas controlling information quality assessment. It is composed of three layers: data acquisition, query and ontology layers. It uses query templates based on the SPARQL Inferencing Notation (SPIN)[17] to express quality requirements. The queries are built to compute weighted and unweighted quality scores. At the end of the assessment, it uses vocabulary elements to annotate important values of properties and classes, assigning inferred quality scores to ontology elements and classifying the 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 [**?**] the authors highlight the fact that the RDFification process of some data can

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

be more challenging than others, leading to errors in the Linked Data provisioning process that needs manual intervention. This can be more visible in datasets that have been semi-automatically translated to RDF from their primary source (the best example for this case is DBpedia [?]). 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[18].

TripleCheckMate [?] 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 [?] 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 [?]. 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) [?]. Any additional quality metrics added to the framework should extend it.

RDFUnit[19] is a tool centered around the definition of data quality integrity constraints [?]. 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 [?] 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

[18] https://www.mturk.com/
[19] http://github.com/AKSW/RDFUnit

links).

Quality Assessment of Data Sources (Flemming's Data Quality Assessment Tool)[20] calculates data quality scores based on manual user input. The user should assign weights to the 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 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 [**?**] is the Open Refine[21] 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[22] 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[23] 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. Tradi-

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

tional link analysis has proven to be an effective way to measure the quality of Web documents search. Algorithms like PageRank [?] and HITS [?] 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 [?][?]. However, the basic assumption that links are equivalent does not suit the heterogeneous nature of links in the Linked Open Data. Thus, the previous approaches fall short to provide reliable rankings as the types of the links can have a direct impact on the ranking computation [?]. The first adaption of PageRank for Semantic Web resources was the Ontology Rank algorithm implemented in the Swoogle search engine [?]. 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 [?] rankings are computed at query time based on two levels of granularity: resources and context graphs. DING [?] 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[24][?] 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 [?] is a fully automated approach which takes a set of RDF triples as an input and analyzes it to extract topological measures (links quality). However, the authors depend only on five metrics to determine the quality of data (i.e.degree, clustering coefficient, centrality, sameAs chains and descriptive richness through sameAs).

Provenance-based Approaches

Provenance-based assessment methods are an important step towards transparency of data quality in the Semantic Web. In [?][25] the authors use a provenance model as an assessment method to evaluate the timeliness of Web data. Their model identifies types of "provenance elements" and the relationships between them. Provenance elements are classified into three types: actors, executions and artifacts. The assessment procedure is divided into three steps: 1) Creating provenance graph based on the defined model 2) Annotating the graph with impact values 3) Calculating the information quality score. In [?] the authors describe a set of provenance-based assessment metrics to support quality assessment and repair in Linked Open Data. They rely on both data and metadata and use indicators like the source reputation,

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

freshness and plausibility. In [?] the authors introduce the notion of naming authority which connects an identifier with the source to establish a connection to its provenance. They construct a naming authority graph that acts as input to derive PageRank scores for the data sources.

Entity-based Approaches

Sindice [?] uses a set of techniques to rank Web data. They use a combination of query dependent and query independent rankings implemented in the Semantic Information Retrieval Engine (SIREn)[26] to produce a final entity rank. Their query dependent approach rates individual entities by aggregating the the score of the matching terms with a term frequency - inverse subject frequency (tf-isf) algorithm. Their query independent ranking is done using hierarchical links analysis algorithms [?]. 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 [?][27] the authors present their findings to measure the discoverability of SPARQL endpoints by analyzing how they are located and the metadata used to describe them. In addition to that, they also analyze endpoints interoperability by identifying features of SPARQL 1.0 and SPARQL 1.1 that are supported. The authors tackled the endpoints efficiency by testing the time taken to answer generic, content-agnostic SPARQL queries over HTTP.

5.5 An Objective Quality Assessment Framework

Looking at the list of objective quality indicators, we found out that a large amount of those indicators can be examined automatically from attached datasets metadata found in data portals. As a result, we have chosen to extend Roomba as it performs the preprocessing steps needed to objectively measure datasets quality.

In our framework, we have presented 30 objective quality indicators related to dataset and links quality. The remainder 34 indicators are related to the entities and models quality and cannot be checked through the attached metadata. Excluding security related quality indicators as LOD cloud group members should not restrict access to their datasets, the Roomba quality extension is able to assess and score 23

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

of them (82%).

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

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 dereferenceable 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 dereferenceable
21	Check if the dataset contains valid `license_id` and `license_title`
22	Check if the `license_url` is dereferenecable
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 dereferenceable (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, 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 **??**.

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 **??**.

After running Roomba and examining the results on the selected datasets and groups, we found out that our framework provides 100% correct results on the in-

Dataset ID	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

dividual dataset level. Roomba's aggregation have been evaluated in [**?**], 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[28]. These profiles cover the indicators in Table **??**. 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.

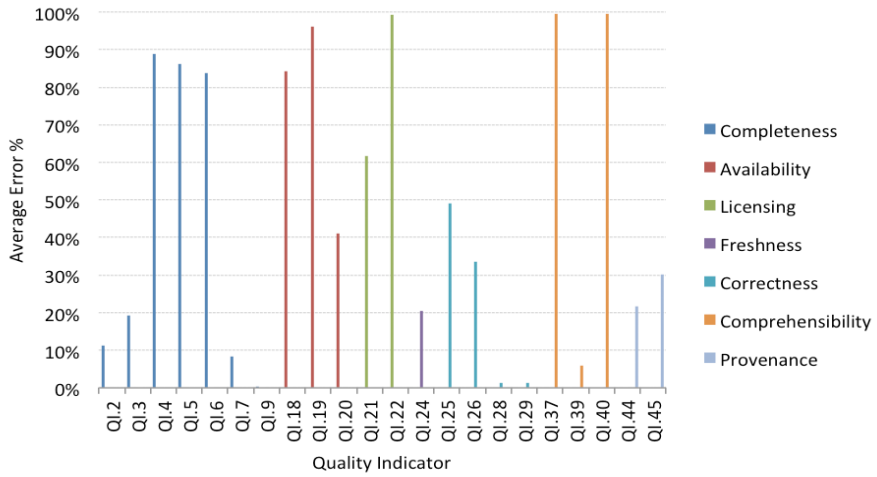

Figure 5.1: Average Error % per quality indicator for LOD group

5.5.3 Experiments and Analysis

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

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

resource extractor in order to cache the metadata files for these datasets locally and ran the quality assessment process which took around two hours on a 2.6 Ghz Intel Core i7 processor with 16GB of DDR3 memory machine.

We found out that licensing, availability and comprehensibility had the worst quality measures scores: 19.59%, 26.22% and 31.62% respectively. On the other hand, the LOD cloud datasets have good quality scores for freshness, correctness and provenance as most of the datasets have an average of 75% for each one of those measures.

Figure **??** 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 Quality Extension vs. state of the art

Looking at Section **??** 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 [**?**]. PoolParty covers also a wide set of those indicators but it targets SKOS vocabularies only. However, we notice a lack in automatic tools to check the dataset quality especially in its completeness, licensing and provenance measures. Table summarizes the automatic dataset quality approaches that have implemented tools (full circle denotes full quality indicator assessment, while half circle denoted partial assessment). As can be seen in Table **??** 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[29]. We have also used the open source and knowledge license information[30] to normalize license information and add extra metadata like the domain, maintainer and open data conformance.

The quality report is currently generated in a JSON format. Leveraging quality vocabularies like the Data Quality Vocabulary (DQV) [**?**] allows us to expose this data in a machine readable format so that they can be automatically consumed by various applications.

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

[30] 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

5.7 Summary

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

Conclusion of Part ??

In this part, we presented the various parts required to automatically assess and build harmonized dataset profiles.

First, we surveyed the landscape of various models and vocabularies that described datasets on the web. We have identified four main sections that should be included in the model and classified the information to be included into eight types. We proposed HDL, a harmonized dataset model, that takes the best out of these models and extends them to ensure complete metadata coverage to enable data discovery, exploration and reuse.

Second, we detail the gaps in the current tools for automatic validation and generation of dataset profiles. Afterwards, we propose Roomba to tackle these gaps and show the results of running it on various data portals.

Last, we cover the quality dimension from HDL. We propose an objective assessment framework by identifying quality indicators that can be automatically measured by tools. We further survey the landscape of quality tools and discover various shortcomings. As a result, we extend Roomba and cover 82% of the proposed quality indicators.

Going back to our scenario, our data portal administrator **Paul** will be able to use HDL as a basis to extend and present the datasets he controls. Moreover, he can use HDL and the proposed mappings as a basis to extend Roomba to support various dataset models like DKAN or Socrata.

Roomba with its quality extension helps **Paul** to have a detailed overview on the health and quality of the datasets. He can use it to automatically fix some issues, and notify the datasets owners of the other issues to be manually fixed. He will be able to identify spam datasets resulting in higher data quality.

Dan on the other will be able to have access to cleaner, richer set of datasets. He will be able to examine detailed attributes of the datasets. This will help **Dan** to make more infomred decisions on which dataset to use in his report.

Part II

Towards Enriched Enterprise Data

Overview of Part ??

In Part **??**, 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 **??**, 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 **??**, we identify the need for an enterprise knowledge base. We detail the challenges and design decisions to import DBpedia into SAP HANA. We further present a set of tools that enable entity disambiguation, entity properties rankings and semantic enrichment on top of DBpedia. We also enhance an in-house schema matching tool called AMC with a set of matchers that show that using Linked Data to map cell values with instances and column headers with types improves significantly the quality of the matching results and therefore should lead to more informed business decisions.

In Chapter **??**, 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 [**?**]. 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 [**?**], the authors argue that SOA is well-suited for transactional processing rather than an approach for data integration.

- Ontologies can be used as a rich format to describe queries and data mappings between schemas and sources. However, developing ontologies require specific skills and it is difficult to provide a complete model that captures the dynamics of the enterprise.

- Linked Data paradigm is a slightly different approach from the ontology-based by exploiting Semantic Web technologies like RDF to represent enterprise taxonomies. The LD approach allows terms to be easily reused and extended.

Data integrated from various resources should be loaded into a central repository often referred to as a Data Warehouse (DW).

A *Data Warehouse* is a large repository where integrated data from different resources reside for the purpose of analysis. Feeding data into the warehouse is done using the Extract-Transform-Load (ETL) process: First the data is extracted from the operational source systems (ERP, CRM, etc.) and then the transformation process is applied in order to unify the data into the warehouse format. Finally the loading is applied to import the data to the warehouse.

6.2 Business Intelligence

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

6.2.1 Multidimensional Model

The traditional relational model is efficient in performing "online" transactions. However, it has clear shortcomings when the objective is to analyze large scale data. The multidimensional model is designed specifically to support data analysis by presenting data as facts with associated numerical values. The multidimensional model has the following fundamental concepts:

- **Dimensions**: Textual data used for labeling, selection, filtering and grouping of data at various levels of details. A dimension is organized into a 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. Figure **??** shows an example of a hierarchical geography dimension.

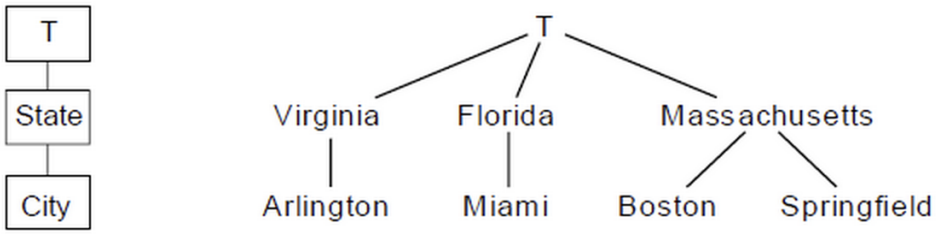

Figure 6.1: Example of a hierarchical geography dimension

- **Measures**: A measure has two components, a numerical property and a formula (usually an aggregation function such as sum or average). Measures generally represent the properties of a chosen fact.

- **Facts**: Facts are the objects that present the subject of the analysis. They are mostly defined by their combination of dimension values. a fact has a certain granularity which is determined by the levels from which its dimension values are drawn.

- **Cubes**: A cube is a multidimensional data structure for capturing and analyzing data. It generalizes the tabular spreadsheet such as there can be any number of dimensions (in contrast to only two in the tabular spreadsheets).

- **Pivot Tables**: A pivot table is a two dimensional table of data with associated subtotals and totals. It may also allow the user to use hierarchies to drill down or roll up. It can be also nested into several dimensions on one axis or pivoted such as the dimensions can be rotated (swapping x and y).

6.2.1.1 Relational Representation

There are two principal ways of representing dimensions:

- **Star Schema**: A star schema has one table for each dimension. This table has a key column and one column for each level of the dimension. Furthermore, a star schema has a Fact Table that hold a row for each multidimensional fact and has a column for each dimension. The primary key in the dimension tables is typically a surrogate key (ID). This results in better storage, prevention of key-reuse problems and more efficient query processing.

- **Snowflake Schema**: Very similar to the star schema. However, it contains several dimension tables for each dimension. This results in removing the redundancy found in star schemas. As a result, querying the schema is now harder since several joins must be applied resulting in longer processing time to compute the results.

6.2.1.2 Analysis and Querying

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

The support for analysis and querying on cubes is done using these operations:

- **Rolling up**: Rolling up causes the data view to go up to a higher cross grained view

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

6.2.2 SAP BI Application Suite

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

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

6.3 SAP High Performance Analytic Appliance (HANA)

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

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

platform that is best suited for performing real-time analytics, and developing and deploying real-time applications. At the core of this real-time data platform is the SAP HANA database (see Figure **??**) which leverages the cheap price of memory chips and does the computation operations all in the memory instead of disk. For BI and Real-time analytics HANA specializes on:

- **Data Warehousing**: Provides real-time data warehousing which allows businesses to rapidly access their Enterprise Data Warehouse (EDW).
- **Operational Reporting**: Provides real-time insights and Business Intelligence from transaction systems such as ERP.
- **Predictive and text analysis on Big Data**: Provides the ability to perform predictive and text analysis on large volumes of data in real-time. With its text search/analysis capabilities SAP HANA also provides a robust way to leverage unstructured data.

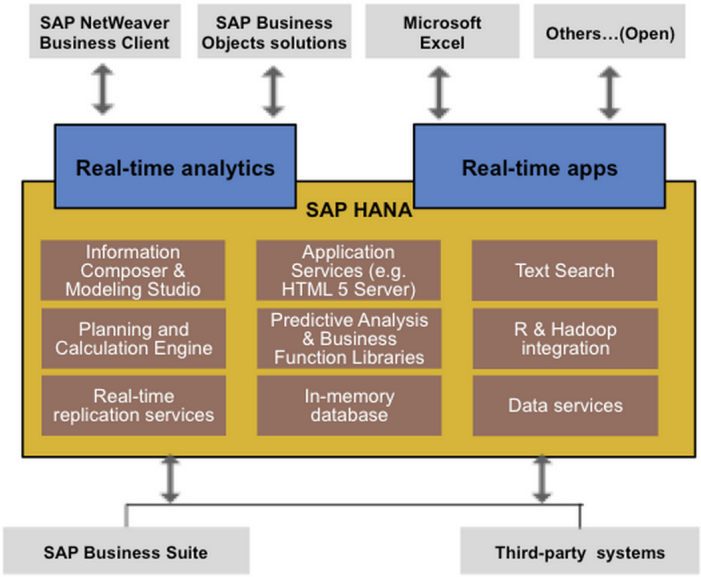

Figure 6.2: SAP's High Performance Analytic Appliance (HANA) with the SAP BI suite

HANA has both columns and rows stores for data storage, the user specifies on which data store he wishes to put his data. Row stores are fit for traditional transaction systems (traditional databases) when transactions are done on row level. However BI queries or analytical queries are done on subsets of columns as the database does not need to access all the elements in a row in order to fetch the required data.

HANA has mainly three views on data:

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

6.3.1 HANA XS-Engine

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

6.3.2 Active Information Store (AIS)

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

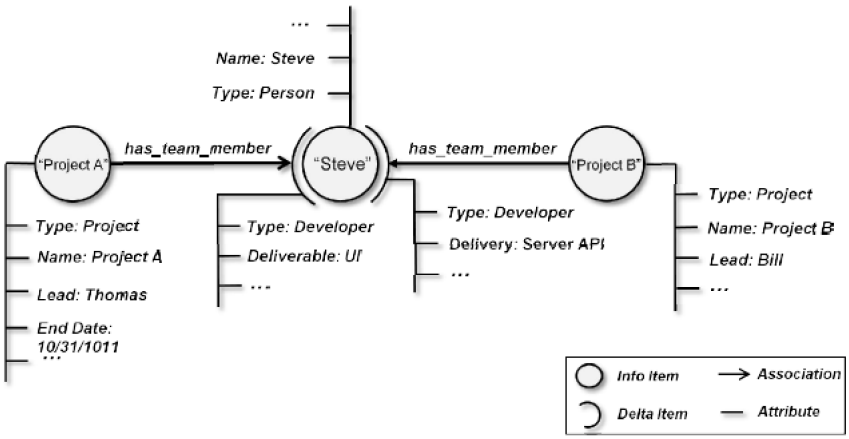

Figure 6.3: AIS data model

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

6.4 Social Web

The social web (Web 2.0) is about websites and services designed and developed to support and foster social interactions [**?**]. 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 the work of 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 allows 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 "hashtag" 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+.
- **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.

[2] http://twitter.com
[3] https://plus.google.com
[4] http://stackexchange.com
[5] http://youtube.com

- **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. The website also supports documents, videos and webinars and acts as an educational and e-learning hub.

[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 [**?**].

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 [**?**].

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. 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 the provision of data integration services [**?**]. In this chapter, we present our work in using DBpedia as an internal knowledge base. We further present a set of services that we implemented on top of DBpedia allowing entity disambiguation and enhancing 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 relevant properties for an entity. 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 as an illustration. It is a crowd-sourced community effort to extract information from Wikipedia and present it in structured accessible formats [**?**].

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 that hold the main core information of DBpedia:

- **Mapping-based Types**: Contains types assignment 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.

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

DBpedia is modeled as an RDF graph. A natural way to import DBpedia into SAP HANA would have been to its graph engine, namely AIS. To do that, we would need to map the RDF triples to the AIS data model (see Section **??**). This requires 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

[1] http://dbpedia.org

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.
- 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 DBpedia. As a result, we decided to import DBpedia into HANA's column store. Table **??** shows the tables structure used.

7.1.1 Entity Disambiguation with DBpedia in SAP HANA

After successfully, importing DBpedia into HANA's column store, we need to create a service that is able to disambiguate a query string (full documentation of the API is found in appendix **??**). HANA has a built-in (fuzzy) text search function that can be used. However, relying on string matching only is not sufficient. To better rank the search results, we combined a link-based rank approach that takes into account the number of incoming associations as shown in Equation **??**.

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"

While the number of outgoing associations can vary between entities,it is rather considered as 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 thus reflecting its popularity.

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

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 **??** and its affect on the overall score for query "apple"

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

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 that 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 our entity disambiguation API in Section **??**.

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. Datasets 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 [?] but literature mainly covers automatic or semi-automatic labeling of anonymous datasets through Web extraction. Examples include [?] that automatically labels news articles with a tree structure analysis or [?] 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. [?] goes a step further by launching speculative queries to standard Web search engines to enlarge the set of potential candidate labels. More recently, [?] 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 [?] present techniques to automatically infer a semantic model on tabular data by getting top candidates from Wikitology [?] 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 [?]. In the Helix project [?], 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[2]. 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) [3] 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 datasets. Moreover, Open Refine has some strict assumptions over the input of spreadsheets which make it difficult to identify primitive and complex data types.

Open Refine makes use of a modular web application framework similar to OSGi called Butterfly[4]. The server-side written in Java maintains states of the data (undo/redo history, long-running processes, etc.) while the client-side implemented in

[2] http://www.freebase.com/
[3] http://openrefine.org/
[4] http://code.google.com/p/simile-butterfly/

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.

The AutoMapping Core (AMC) [**?**] 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.).

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 Intelligence reporting tools. We first map cell values with instances and column headers with types from popular datasets 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 that is built on top of the entity disambiguation API in Section **??**. Figure **??** gives an outline of these activities.

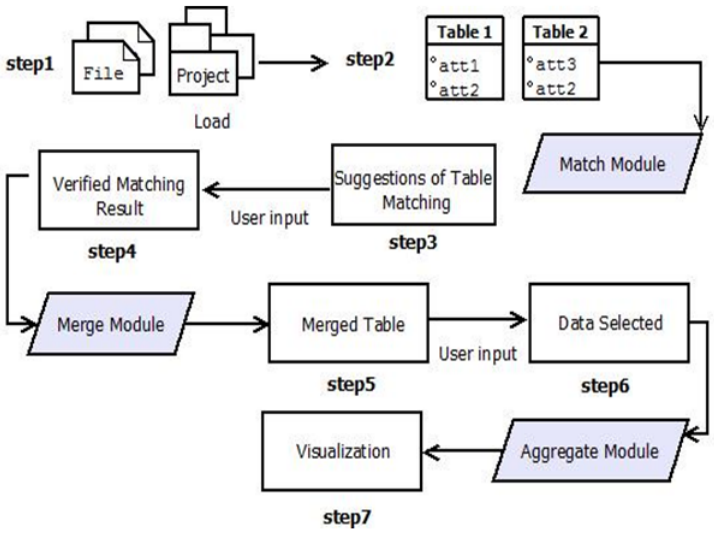

Figure 7.1: RUBIX Activity Workflow

The datasets 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 datasets. These projects are imported into the internal data structure (called schema) of the AMC [**?**] (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 **??**.

Figure 7.2: Screenshot showing the results mapping view in RUBIX

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 the union of the two projects where the matched columns of 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 (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 disambiguation, 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 datasets, 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 **??** 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 [**?**][**?**]. 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 Vector-based Similarity

The first algorithm that we implemented is based on vector algebra. Let v be the vector of ranked candidate types returned by the disambiguation API 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) [**?**]. 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, $C_{sr1},\ C_{sr2}, \dots,\ C_{srn}$ are the sum of scores for each corresponding r occurrence in the source column, and $C_{tr1},\ C_{tr2},\ \dots,\ C_{trn}$ 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:

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

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:

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

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 datasets, 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 **??**.

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 7.4: 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 **??**.

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

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 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 **??**.

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

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

We notice that by plugging the Spearman method, the number of matches and similarity results have decreased (-4). After Several experiments we have found that this method does not work well with noisy datasets. 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 datasets 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 datasets. 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 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 **??**.

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 7.8: 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 **??**.

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 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 properties are more "important".

In this section we provide a method enabling business users to select what properties should be used when depicting the summary of an entity. For example, when our analyst **Dan** wishes to enrich his reports with external data, he is overwhelmed by the number of dimensions he can add. We reverse engineered the Google Knowledge graph panel (see Figure **??**) to find out what are the most "important" properties for an entity according to Google that can be used to enrich business reports. We compare these results with a survey we conducted on 152 users.

7.3.1 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 [**?**]. 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 (since Freebase is the knowledge base behind the graph panel) 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[5].

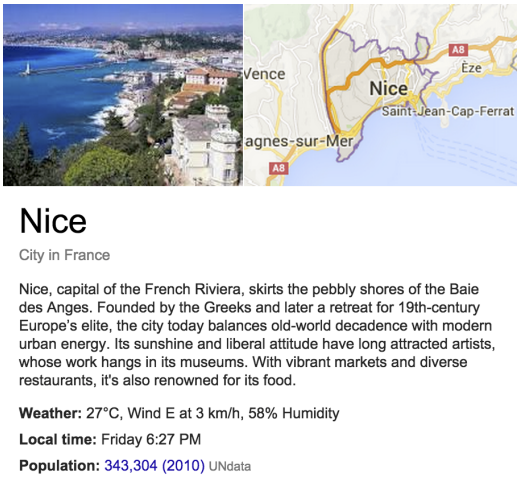

Figure 7.3: Google knowledge graph panel for the city of Nice, France

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
```

For each of these concepts (e.g., `Band`, `Organization`, `ArchitecturalStructure`), we retrieve n instances (in our experiment, n was equal to 100 random instances). For example, for the concept `Band` we retrieved: `!Action_Pact!`, `12_Stones`, `20_Fingers`, etc. For each of these instances we issue a search query to Google containing the in-

[5] https://github.com/ahmadassaf/KBE/blob/master/results/dbpediaConcepts.json

stance 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 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 **??** 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 **??**. We finally observe that this experiment is only valid for the English Google.com search results since GKP varies according to top level names.

7.3.2 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.

7.3.2.1 User survey

We set up a survey[6] 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 selected 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[7] show the ranking of the top properties for each entity. We only keep the properties having received at least 10% votes for comparing 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.

7.3.2.2 Comparison with Google Knowledge Graph

The results of the Google Knowledge Panel (GKP) extraction[8] clearly show a long tail distribution of the properties depicted by Google, with a top N properties (N

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

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

[8] https://github.com/ahmadassaf/KBE/blob/master/results/survey.json

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 **??** 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 conclusions 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 7.9: 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.

7.3.2.3 Modeling the preferred properties with Fresnel

Fresnel[9] 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` [**?**].PROV-O[10] is a vocabulary to describe semantically rich metadata with focus on providing detailed provenance, license and access information. We use those two vocabularies to explicitly represent what properties should be depicted when displaying an entity[11]. This dataset can now be re-used as a configuration for any consuming application (see Appendix **??** for a snippet of the generated Fresnel file).

7.4 Summary

In this chapter, we presented an entity disambiguation API built on top of SAP HANA. We used this service in a framework to enable mashup of potentially noisy enterprise and external data. The API is used to annotate business reports 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,

[9] http://www.w3.org/2005/04/fresnel-info/
[10] http://www.w3.org/TR/prov-o/
[11] https://github.com/ahmadassaf/KBE/blob/master/results/results.n3

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 [**?**] 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 [**?**].

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 both the Internet and Intranets. 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" representing 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.

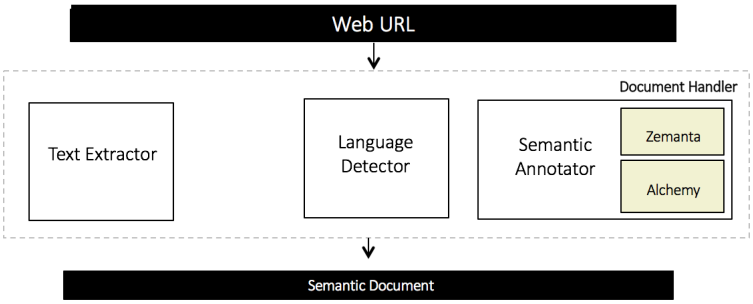

Figure 8.1: SNARC's Document Handler

The Semantic Model is created by the Document Handler (see Figure **??**) 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.

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.

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

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 entities 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 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].

[2] http://developer.zemanta.com/docs/suggest/
[3] http://www.alchemyapi.com/api/categ/categs.html
[4] http://www.dmoz.org/desc/Top

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. Figure **??** shows the different steps in order to retrieve a set of social results.

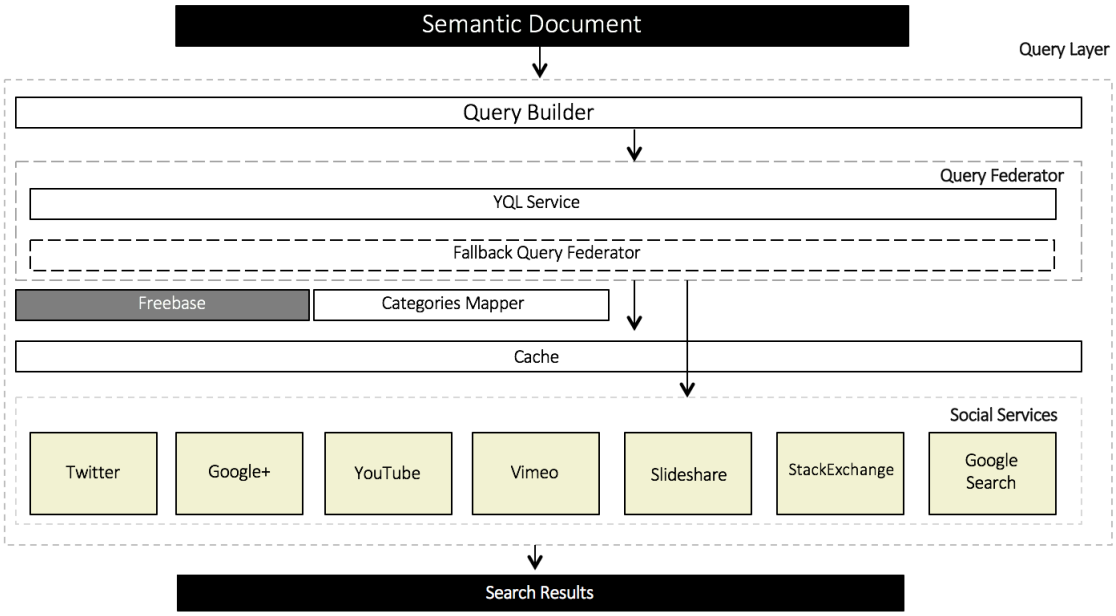

Figure 8.2: SNARC's Query Layer

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

[5] http://www.bbc.co.uk/news/world-europe-22889060

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 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

[6] http://stackexchange.com/sites
[7] http://developer.yahoo.com/yql/

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 Alchemy's 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 are unified and wrapped in a single social model. Figure **??** shows the different steps needed to produce the final parsed results that will be pushed back to the front-end.

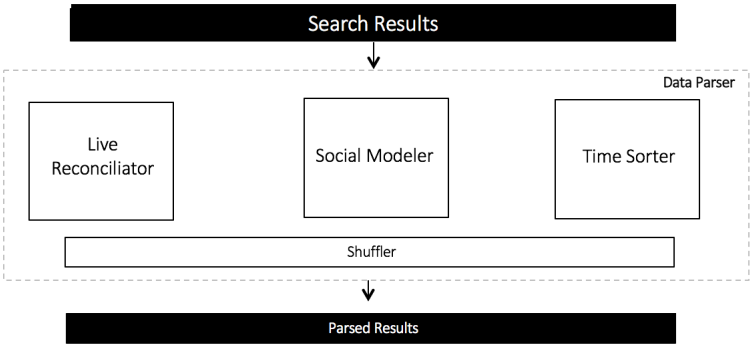

Figure 8.3: SNARC's Data Parser

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

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

keywords that they think describe best the content. This contrasts with taxonomies that over-impose hierarchical categorization of content [**?**]. 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 like [**?**, **?**, **?**, **?**] 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 (service name and type), author information (author's name and profile link) and general post information (title, thumbnail, link, embed code and time). Listing **??** shows the model for some of the social networks about an article for understanding critical CSS[10].

```
[service] => twitter
[type] => micropost
[time] => 1 hour ago
[title] => RT @ProjectPeachUK: Need a practical guided tour of our html5
    live web? Book your tour now via twitter or our live webapp http://t.
    co/DgtAB",25)
[link] => 641623665579270144
[author] => TheOnlyHTML5God
[thumbnail] => http://pbs.twimg.com/profile_images/593400198786842624/
    UNw7ITcE_normal.jpg
[program] => Twitter Web Client
```

Listing 8.1: Social Modeler output snippet for different social networks

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.

[10] http://smashingmagazine.com/2015/08/understanding-critical-css/

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 Figure **??**). This implementation offers more flexibility to users by loading related social news anytime on any webpage or application. The results are visualized using 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.

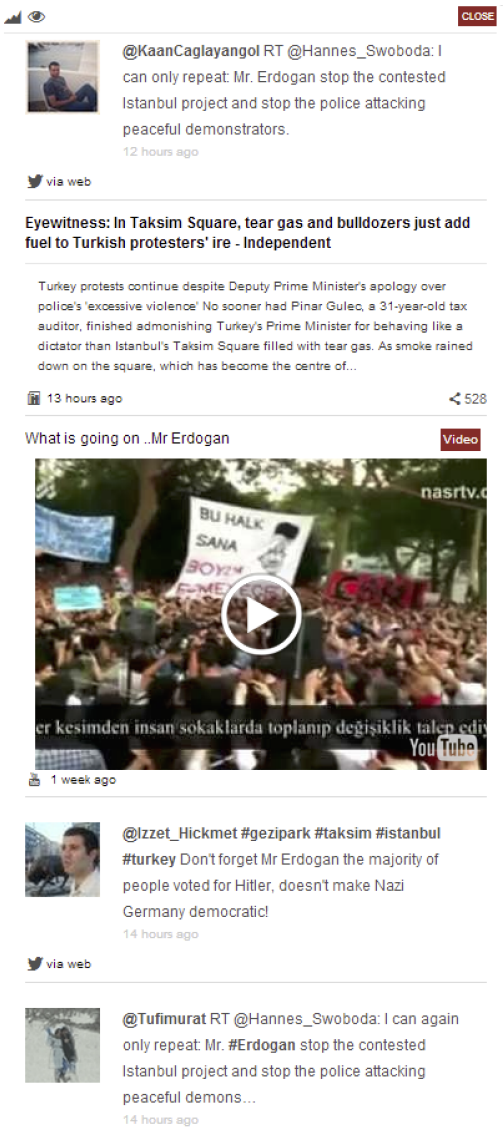

Figure 8.4: SNARC's User Interface - The Google Chrome Extension

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 ??

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 **Dan** to be able to find and match various reports he is working on. Moreover, he will be able to augment extra measures and dimensions to his reports using DBpedia. In addition, **Dan** will be able to monitor relevant social feeds. This will allow him to either embed social snippets directly in his reports or to 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 Scenario Flashback

Going back to our scenario defined in Section **??**, we have defined two main personae. The first is a data analyst called **Dan** who works with the Ministry of Transport in France. He receives a memo from his management to create a report comparing the number of car accidents that occurred in France for this 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.

The second is a data portal administrator called **Paul**. He is affiliated with the British Open Data portal (`data.gov.uk`). His daily job includes acquiring, preparing and publishing relevant datasets on the portal. He always strives to maintain a spam-free, high-quality portal.

9.2 Achievements

This thesis thoroughly describes the different steps aiming at realizing the vision of enabling self service data provisioning in the enterprise (see Figure **??**). The work presented is beneficial to both our personae introduced. The contributions made are:

Contributions for Data Portals Administrators

Our data portal administrator **Paul** is always looking to expand his portals in terms of the number of datasets hosted, without compromising in their portal's data quality. In Chapter **??** (component **B** in Figure **??**), we surveyed the landscape of various models and vocabularies that described datasets on the web. We found a shortcoming when it comes to having a complete descriptive dataset model taking into account access, license and provenance information. As a result, we proposed

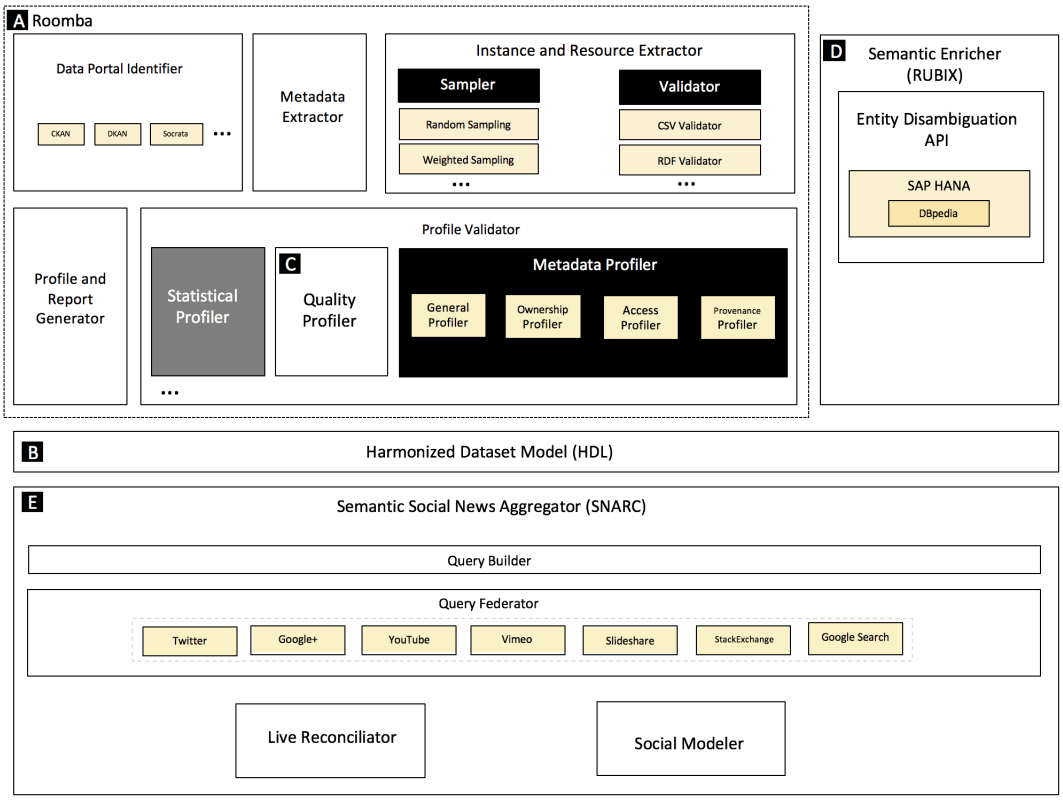

Figure 9.1: Annoteted architecture diagram for enabling self-service data provisioning

a Harmonized Dataset Model (HDL) that **Paul** will use as a basis to extend and present the datasets he controls. **Paul** now also knows what are the major dataset models out there, and what kind of metadata data owners need to fully represent their dataset. The mappings proposed in Section **??** will allow him to easily integrate data from various data management systems into his own.

In Chapter **??** (component **A** in Figure **??**), we proposed Roomba, an automatic dataset profiles generation and validation tool that can be easily extended to perform various profiling tasks. Out of the box, **Paul** can use Roomba to automatically fix datasets metadata issues, and notify the datasets owners of the other issues to be manually fixed.

In Chapter **??** (component **C** in Figure **??**), we proposed a comprehensive objective quality framework applied to the Linked Open Data. Moreover, after surveying the landscape of existing data quality tools, we identified several gaps and 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. **Paul** will be able now to identify spam and low quality datasets. In addition to that, data available in his portal will now have rich semantic information attached to it. For example,

temporal and spatial information extracted will be assigned into the corresponding fields in HDL. As an exemplary result, various datasets will be easily identifiable to cover various parts of the UK.

Contributions for Data Analysts

Our data analyst **Dan** believes that "more data beats better algorithms" and is always hunting for high quality data to produce accurate reports to the management team. By examining the rich datasets metadata presented in HDL he will be able to make fast decisions whether the dataset examined is suitable or not. He will also have vital information about the licensing and limitations for using this data internally. He will also have assurances on the dataset quality, which will help choose the best candidates out of ranked list.

Dan will be able to have direct access to rich and high-quality dataset descriptions generated by Roomba. Moreover, the topical profilers in Roomba will be able to identify occurrences of alcohol related terms like "wine" in various datasets. Query expansion methods can be used to relate alcohol to wine allowing him to find the datasets he wants.

In Chapter **??** (component **D** in Figure **??**), we presented an entity disambiguation API built on top of SAP HANA. This API is used in RUBIX, a framework we proposed to enable mashup of potentially noisy enterprise and external data. **Dan** now has access to various datasets that he found matching his query to the portal administered by **Paul**. He will be also able to use the schema matching services to find and merge those datasets in his reports.

Having imported those dataset into Lumira, he will be also able to use the internal knowledge base to apply various semantic enrichments on this data. Figure **??** shows a possible integration of such service in SAP Lumira where **Dan**

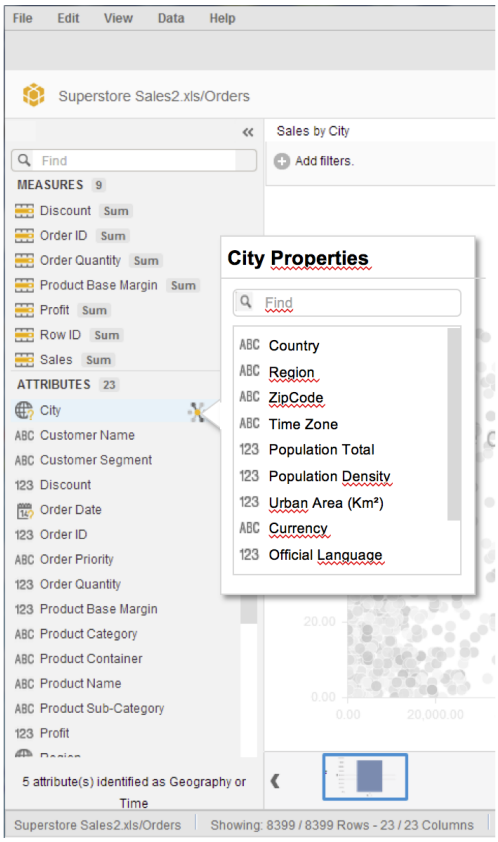

Figure 9.2: UI Prototype of semantic data enrichment in SAP Lumira

is presented with a ranked set of properties retrieved by the algorithm proposed in Section **??**.

In Chapter **??** (component **E** in Figure **??**), we proposed SNARC, a semantic social news aggregation service that allows the user to explore relevant news from internal or external sources. **Dan** is also a modern person, who is always trying to fresh information and believes in the wisdom of the crowd. Having SNARC services integrated with Lumira, he is also able to see a feed of relevant social media items that can be of interest to him. He actually follows a link in some tweet that he saw and was able to find relevant pieces of pointers that he would like to investigate further.

In summary, the contributions above pave the way to build a set of smart services to enable analysts easily find relevant pieces of information and administrators fight spam and be able to maintain high quality data portals. The work presented in this thesis goes beyond the fact that attaching metadata to datasets is vital, but propose a set of services that can automatically achieve that in seamless manner.

9.3 Perspectives

This thesis could be extended in the following directions:

- **Data Profile Representation**

  The proposed Harmonized Dataset Model (HDL) is currently available as a hierarchical JSON file. An enhancement would be to refine HDL and present it as a fully fledged OWL ontology. In addition, HDL can be extended to propose also a set of enumerations as values to ensure a unified fine-grained representation of a dataset. Moreover, while we presented the mappings between various models in a table structure, presenting those mappings in a machine readable format will allow various tools like Roomba to use it.

- **Automatic Dataset Profiling**

  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. There are various extensions to our tool Roomba that can help in automatically building and enhancing dataset profiles. An example of these extension would be the integration of statistical and topical profilers allowing the generation of full comprehensive profiles. We would also like to extend Roomba to be able to run over other data portal types like DKAN or Socrata. This extension can be done by leveraging the data models mappings we proposed. In addition to all that, a possible enhancement will be ability to correct the rest of the metadata either automatically or through intuitive manually-driven interfaces.

- **Objective Linked Data Quality**

  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. In this thesis, we managed to narrow down the set of quality issues surrounding Linked Data to those who can be objectively measured and assessed by automatic tools. Our proposed tool covers 85% of the quality indicators proposed. A possible extension would be to integrate tools assessing models quality in addition to syntactic checkers with Roomba. This will provide a complete coverage of the proposed quality indicators. Moreover, there are currently no weights assigned to the quality indicators. A valid contribution would be to suggest weights to those indicators which will result in a more objective quality calculation process.

- **Enterprise Data Integration**

  A vital component to Data Integration in the enterprise is the existence of enterprise knowledge bases. Integrating additional linked open data sources of semantic types such as YAGO and evaluate our matching results against instance-based ontology alignment benchmarks such as OAEI[1] or ISLab[2] are possible future directions. Moreover, our work can be generalized to data classification. The same way the AMC helps identifying the best matches for two datasets, we plan to use it for identifying the best statistical classifiers for a sole dataset, based on normalized scores.

[1] http://oaei.ontologymatching.org/2011/instance/index.html
[2] http://islab.dico.unimi.it/iimb/

Part III

Résumé de Thèse

Chapter 10

Introduction

La Business Intelligence (ou BI) a toujours été porté par la transformation des données brutes en données intelligibles. En donnant un sens aux données, il est plus facile de conduire des changements dans les organisations. Un aspect clé pour apporter du sens aux données est d'attacher des informations précises en lien avec les données représentées. Cet aspect est étudié dans le domaine de la Sémantique.

La Business Intelligence ainsi que les nouveaux outils de visualisation agile concentrent une grande partie de leurs caractéristiques de vente sur des représentations visuelles attrayantes et uniques. La principale tâche, la plus ardue, est de préparer les données pour ces outils de visualisation dans la plupart des projets de BI, petits ou grands. Ainsi, le but final de la BI est de simplifier la prise de décision en éliminant toute la lourdeur liée à la gestion des systèmes d'informations et des nombreuses données collectées. Traditionnellement, les approches BI sont gérées avec une version centralisée et une séparation distincte entre le monde du système d'information et le monde de l'entreprise décisionnelle. La recherche des données par l'utilisateur concerné apporte une solution pour rapprocher ces deux mondes et faciliter la capture et intégration d'une manière intuitive pour l'utilisateur final.

10.1 Contexte et Motivation

Les entreprises utilisent un large panel de systèmes d'information hétérogènes dans leurs activités telles que la planification des ressources d'entreprise (ERP), de gestion des relations client (CRM) et de systèmes de chaine d'approvisionnement (SCM). Un système d'information distribué contient plusieurs systèmes utilisant différentes briques technologies et plusieurs standards différents [**?**]. En plus de cette hétérogénéité, la quantité d'information stockée dans les bases de données de l'entreprise et sur les différents fournisseurs de stockage sur Internet augmente de façon exponentielle chaque année. Ce qu'on appelle "Enterprise Big Data" n'est pas seulement corrélé à un grand espace de stockage, mais aussi a une grande variété dans les formats de fichiers associés. L'information est également souvent stockées dans des formats non structurés et inconnus.

L'intégration des données est un challenge. Elle nécessite la combinaison de données provenant de différentes sources pour fournir à l'utilisateur une vue unifiée de ces données [**?**]. Dans les grandes entreprises, c'est une tache coûteuse en ressources

et en temps. Diverses approches ont été proposées pour résoudre les problématiques d'intégration. Ces approches ont été principalement basées sur XML comme syntaxe de représentation de données, les web-services en tant que protocole d'échange de données et sur une architecture orientée service (SOA) comme approche holistique de l'architecture de systèmes distribués et de la communication. Cependant, il a été constaté que ces technologies ne sont pas suffisantes pour résoudre les problèmes d'intégration dans les grandes entreprises [**?**, **?**]. Récemment, des approches d'intégration de données basés sur l'ontologie ont été suggérées, pour lesquels les ontologies sont utilisées pour décrire les données, faire des requêtes dessus et les synchroniser [**?**]. Une approche légèrement différente est l'utilisation du concept de "Linked Data" [**?**] pour l'intégration de données d'entreprise. Des entreprises comme Google et Microsoft utilisent ce paradigme pour l'intégration de données liées à leurs systèmes d'information, et elles vont encore plus loin en utilisant la technique pour construire des bases de connaissances complète (comme le Google Knowledge Graph alimenté en partie par Freebase[1]) qui agissent comme un point central de leurs données structurées.

Les données sont plus utiles quand elles sont ouvertes et largement diffusées dans des formats partageables pour ensuite être le point de départ d'analyses plus poussées. La qualité et la quantité de connaissance structurée disponible sur le web rendent désormais possible pour les entreprises de parcourir cette énorme quantité de données publiques et de l'intégrer dans leurs systèmes d'information de nouvelle génération. Un exemple de ce type de données externe est l'initiative du Linked Open Data (LOD) disponible sur le cloud. En 2007, il y avait 12 bases de données référencées. Aujourd'hui, le catalogue contient plus de 1000 bases contenant plus de 82 milliards d'entrees[2] [**?**]. Les données sont fournies aussi bien par des organismes publics que privés, et couvrent un ensemble diversifié de domaines allant des sciences de la vie aux médias en passant par les données du gouvernement. Le cloud LOD est potentiellement une mine d'or pour les organisations et les individus qui cherchent à tirer parti de sources de données externes afin de produire des rapports plus étoffés [**?**]. Ces données externes peuvent être accessibles via des portails de données publiques comme `datahub.io` et `publicdata.eu` ou privés comme `quandl.com` et `enigma.io`. L'analyse de ce nouveau type de données dans le contexte des données d'entreprise permet d'en tirer des analyses nouvelles ou plus poussées, ce qui permet de détecter et explorer de nouvelles opportunités de marchés [**?**].

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

10.2 Scénario d'utilisation

Pour permettre une intégration efficace des données à grande échelle, il ya quelques efforts nécessaires de en plusieurs points. Dans cette thèse, nous abordons les enjeux et les défis de deux points de vues :

- **Analyste de données :** Un analyste de données est un professionnel expérimenté qui est en mesure de recueillir et d'acquérir des données provenant de multiples sources de données, filtrer et nettoyer les données, interpréter et analyser les résultats et fournir des rapports en cours.
- **Administrateur du portail de données :** Un administrateur du portail de données surveille la bonne tenue du portail. Il supervise la création des utilisateurs, des organisations et des ensembles de données. Les administrateurs tentent d'assurer un niveau de qualité de certaines données en vérifiant en permanence le spam, et en amélioration manuellement les descriptions et annotations des ensembles de données.

Tout au long de cette thèse, nous allons présenter un scénario de cas d'utilisation impliquant les deux personnes pour illustrer les défis et les solutions que nous fournissons.

Dans notre scénario, **Dan** est un analyste de données en collaboration avec le ministère chargé des Transports en France. Son outil de prédilection pour les calculs, la manipulation et la visualisation de données est SAP Lumira[3], un outil de visualisation de données en libre-service qui le rend facile pour importer des données provenant de sources multiples, effectuer une analyse BI visuelle à l'aide de tableaux de bord intuitifs, des cartes interactives, des graphiques, et des infographies. Dan reçoit une note de sa direction pour créer un rapport comparant le nombre d'accidents de voiture qui ont eu lieu en France pour cette année, à son homologue du Royaume-Uni (UK). En outre, il est demandé de mettre en évidence les accidents liés à la consommation illégale d'alcool dans les deux pays.

Après avoir examiné les dossiers du ministère, Dan est en mesure de recueillir les données nécessaires pour créer son rapport pour la partie française. Dan publie également une demande officielle au ministère des Transports au Royaume-Uni pour collecter les données nécessaires. Cependant, Dan sait que le processus prend beaucoup de temps alors qu'il doit rendre le rapport sous quelques jours. Dan est familier avec le mouvement Open Data et commence son voyage à travers différents portails de recherche de données au Royaume-Uni.

Paul est un administrateur du portail de données pour le `data.gov.uk`. Il supervise en permanence les processus d'acquisition, et il prépare et publie des ensembles

[3] http://saplumira.com/

de données. Paul essaie toujours de veiller à ce que les données publiées soient de haute qualité et contiennent des métadonnées attachées suffisantes pour permettre une recherche et découverte simple. Paul reçoit souvent des plaintes au sujet des ensembles de données inexacts ou de spam. Il supprime manuellement et corrige les erreurs tout en gardant les canaux de communication ouverts avec les différents départements lui fournissant des données.

10.3 Problématiques de Recherche

Dans le scénario présenté ci-dessus, les fournisseurs de données (administrateurs du portail) et les utilisateurs (analystes de données) ont besoin de solutions pragmatiques qui les aident dans leurs tâches. Pour permettre cela, il y a quelques problématiques de recherche difficiles qui doivent être abordées. Ces défis sont organisés en trois catégories comme suit :

10.3.1 Intégration et Amélioration des Données

- La diversité des sources de données en entreprise soulève de grandes problématiques. Il y a de nombreux formats de fichiers différents, des protocoles d'accès ainsi que des langages de requête variés. Les sources de données possèdent leur propre modèle de données avec différentes façons de représenter et stocker les données. Ces données à travers ces sources peuvent être incomplètes, incompatibles, ou redondantes. Aussi, elles peuvent sembler sémantiquement similaire mais pourtant différentes.**Paul** a besoin d'outils puissants pour cartographier et organiser les données afin d'avoir une vue unifiée pour ces structures de données hétérogènes et complexes.

- L'ajout de métadonnées et d'informations sémantiques peut être délicat. Une entité n'est généralement pas associée à un type générique unique dans la base de connaissances, mais plutôt à un ensemble de types spécifiques qui peuvent être pertinents ou non compte tenu du contexte. **Paul** se doit de trouver le type de l'entité la plus pertinente dans un contexte donné.

- Les entités jouent un rôle clé dans les bases de connaissances en général et dans le Web des données en particulier. Les entités comme celles de DBpedia, sont généralement décrites avec beaucoup de propriétés. Cependant, il est difficile pour **Dan** d'évaluer celles qui sont plus "importantes" que d'autres pour des tâches particulières telles que la visualisation des principales caractéristiques d'une entité.

- Les réseaux sociaux ne sont pas seulement un rassemblement des utilisateurs d'Internet en groupes d'intérêts communs. Ils aident aussi les gens à s'informer

des dépêches, à contribuer aux débats en ligne ou apprendre des autres. Ils sont en train de transformer l'utilisation du Web en se positionnant comme point d'entrée principal dans la recherche, la navigation et l'analyse du comportement d'achat. Cependant, l'intégration des informations de ces réseaux sociaux peut être difficile à **Paul** en raison de la grande quantité de données disponibles. Il est plus difficile de repérer ce qui est pertinent en temps opportun.

10.3.2 Maintenance et Découverte des Données

- Même si les ensembles de données populaires comme DBPedia[4] et Freebase sont bien connus et largement utilisés, il existe d'autres sources de données plus cachées qui ne sont pas utilisées. En effet, ces ensembles de données peuvent être utiles pour les domaines spécialisés, mais sans maintenance d'annuaire précis sur les sujets, il est difficile pour les analystes de données comme **Dan** de les trouver [**?**].

- La quantité croissante de données nécessite des métadonnées riches pour atteindre son plein potentiel. Ces métadonnées permettent la découverte de données, la compréhension, l'intégration et la maintenance de ces données. Malgré les différents modèles et des vocabulaires décrivant les ensembles de données, la possibilité d'avoir une compréhension de l'ensemble des données en inspectant seulement les métadonnées peut être limitée. Par exemple, **Dan** a des difficultés à trouver des ensembles de données avec une couverture géographique spécifique, car cette information est manquante à partir de presque tous les profils de jeux de données examinés.

- Les utilisateurs, les organisations et les gouvernements sont habilités à publier des ensembles de données. Toutefois, les administrateurs du portail de données comme **Paul** ont besoin de vérifier en permanence et manuellement la qualité de ces publications pour détecter le spam et maintenir une haute qualité de service.

10.3.3 Qualité des Données

Les données liées (Linked Data) utilisent de l'information structurée décrite par des modèles, des ontologies et des vocabulaires, et elles contiennent des liens et des destinations pour effectuer des requêtes. L'ensemble de ces informations rend la cohérence, et donc la qualité des donnes difficile à maintenir. Malgré le fait que la tendance et la demande est très forte autour de la qualité des données publiques liées (Linked Open Data), très peu d'initiatives sont initiées pour essayer de normaliser et de formaliser des processus. Ainsi, il n'existe pas de scores ou de certificats permettant d'aider les

[4] http://dbpedia.org

décisionnaires quant à la pertinence et qualité des données. Les administrateurs de portail de données comme **Paul** ont besoin d'avoir une vision globale de la qualité de leurs portails. Ils voudraient ainsi intégrer ces paramètres dans les profils de leur source de données existantes. D'autre part, les analystes de données et les utilisateurs comme **Dan** veulent savoir à l'avance si l'ensemble de données est d'un certain niveau de qualité pour être utilisé dans leurs rapports.

10.4 Contributions de la Thèse

Dans cette thèse, nous proposons un ensemble d'outils pour permettre l'accès à des données en libre-service pour des sources internes et externes à l'entreprise. Les outils contribuent aux trois principales problématiques décrites ci-dessus. En résumé, les principales contributions de ce travail sont les suivantes :

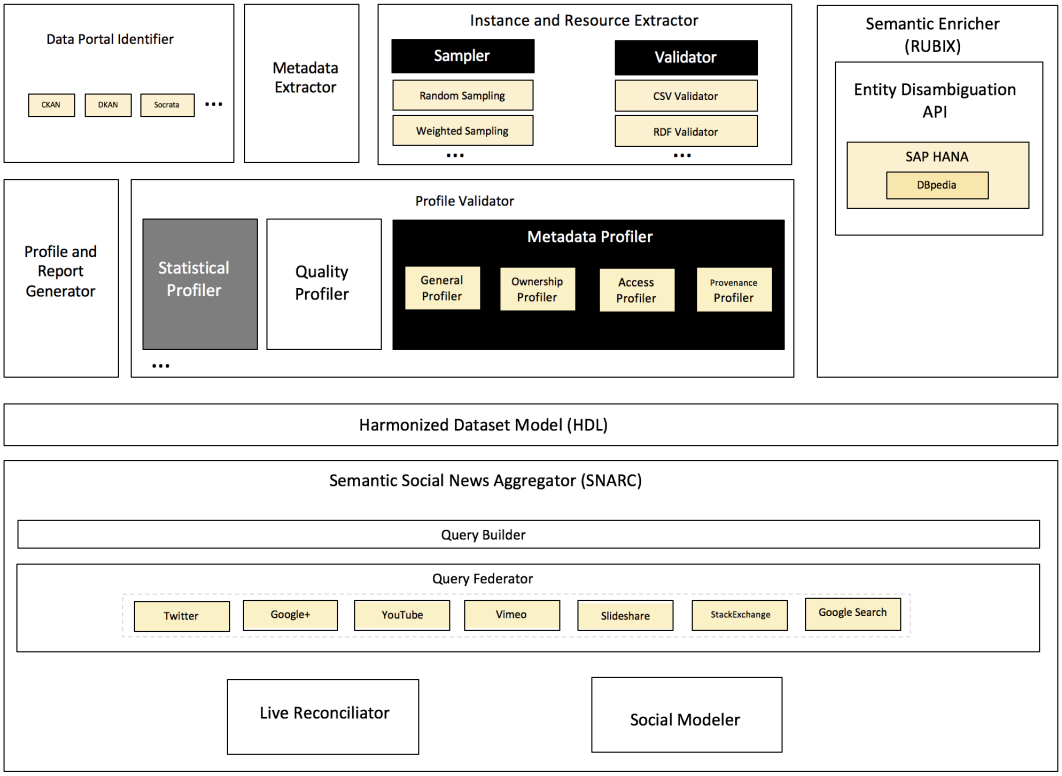

Figure 10.1: Schéma de l'architecture des données pour permettre l'acces en libre-service

10.4.1 Contributions sur la Maintenance et Découverte des Données

En ce qui concerne cet aspect de notre recherche, nous avons accompli les tâches suivantes :

- Nous avons étudiés les différents modèles et vocabulaires qui décrivent des ensembles de données sur le web. Alors que l'utilisation d'un vocabulaire ou d'un modèle commun est la clé dans la communication, nous avons identifié le besoin d'un modèle de métadonnées harmonisé contenant suffisamment d'informations afin que les consommateurs puissent facilement comprendre et analyser les données. Premièrement, nous avons mis en place un ensemble de correspondances entre chacune des propriétés des modèles étudiés. Ceci a conduit à la conception de HDL, un modèle de données harmonisée, qui prend le meilleur sur les modèles étudiés et les étend pour assurer une couverture complète de métadonnées lors de la découverte, exploration et la réutilisation des données.

- Nous avons analysé l'ensemble des outils de profilage de source de données et découvert de sérieuses lacunes. En conséquence, nous avons proposé Roomba, un outil évolutif pour automatiquement extraire, valider, créer et générer des profils de métadonnées décrivant les données liées. Roomba applique plusieurs techniques afin de vérifier la validité des métadonnées fournies et pour générer des informations descriptives et statistiques sur un ensemble de données particulier ou alors pour un portail de données complet.

10.4.2 Contributions sur la Qualité des Données

Concernant nos contributions sur l'évaluation de la qualité des données liées (Linked Data), nous avons accompli les tâches suivantes :

- Nous avons proposé un cadre d'évaluation de la qualité des données liées en se concentrant sur des mesures objectives. Nous avons identifié un total de 64 indicateurs de qualité qui ont été répartis en quatre catégories principales (entité, source de données, liens, modèles) correspondant aux principes de la publication des "Linked Data".

- Sur l'étude des différents outils de qualité de données, nous avons remarqué un manque d'automatisation pour vérifier les mesures proposées dans notre approche. En conséquence, nous avons étendu Roomba pour effectuer une série de contrôles de qualité des données sur les ensembles de données liés. Notre extension couvre la plupart des indicateurs de qualité proposés avec un accent sur l'exhaustivité, l'exactitude, la provenance et les licences utilisées.

10.4.3 Contributions sur l'intégration et l'amélioration des Données

En ce qui concerne cet aspect de notre recherche, nous avons accompli les tâches suivantes :

- Nous avons créé un framework appelé RUBIX qui permet d'utiliser des données provenant de sources peu structurées et privées à des sources publiques et clairement structurées. Le framework exploite des bases de connaissances de référence pour annoter des données avec des concepts sémantiques (métadonnées). Un des avantages de ces métadonnées est d'améliorer le processus d'identification de données similaire à partir de sources hétérogènes au sein d'une entreprise.

- Les métadonnées attachées par RUBIX peuvent encore être utilisée pour enrichir les ensembles de données existants. Toutefois, les concepts sont souvent représentés avec un grand ensemble de propriétés. Pour recommander les plus "importantes" propriétés d'un concept, nous avons analysé les choix faits par Google avec de la retro-ingénierie lors de la création de graphiques et présentés ces choix en utilisant le vocabulaire de Fresnel, de sorte que toute application peut lire ce vocabulaire pour décider les propriétés intéressantes d'une entité.

- L'agrégation des informations des nouvelles sociales n'est pas une tâche facile. Nous fournissons une interface de programmation (API) qui permet l'agrégation des réseaux sociaux appelé SNARC. Nous avons conçu un exemple d'application web tirant parti des capacités de SNARC pour permettre aux utilisateurs de découvrir instantanément les nouvelles pertinentes des réseaux sociaux.

10.5 Vers un Profil de Données Complet

10.5.1 Profils et Modèles d'ensembles de Données

L'utilité des données ouvertes (Open Data) se reconnait à partir du moment où on en a besoin. Les fournisseurs de données se doivent de fournir des données faciles à retrouver. Les portails de données sont spécifiquement conçus à cet effet. Ils rendent facile pour les individus et les organisations le stockage, la publication et la découverte des ensembles de données.

Les portails de données (ou des catalogues de données) sont les points d'entrée pour découvrir les jeux de données publiés. Ils sont les garants des jeux de données et de métadonnées qu'ils hébergent, tout en fournissant un ensemble de services de découverte et d'intégration complémentaires.

Les portails de données peuvent être public comme `Datahub.io` et `publicdata.eu` ou privé comme `quandl.com` et `enigma.io`. Les portails privés exploitent des données consolidées manuellement provenant de diverses sources et les exposent à

des utilisateurs soit librement, soit par le biais de licences. De la même manière, dans certains portails de données publiques, les administrateurs examinent manuellement les informations des bases de données, les valide, les corrige et leur attache des informations de métadonnées approprié. Cette information est principalement sous la forme de tag prédéfinis telles que *médias, géographie, sciences de la vie* pour des raisons d'organisation et de répartition.

Il existe plusieurs systèmes de gestion de données (DMS) qui agrémente les portails publics. CKAN[5] est le leader open-source de portail de données, propulsant des sites web comme DataHub, les données publiques de l'Europe et des données ouvertes du gouvernement américain. Modelé sur CKAN, DKAN[6] est une distribution Drupal qui est utilisée dans différents portails de données publiques. En plus de ces portails de données, il y a un ensemble d'outils qui permettent d'exposer directement les données via des API RESTful, tel que `thedatatank.com`.

Un ensemble de données représentant un modèle de métadonnées doit contenir suffisamment d'informations afin que les consommateurs puissent facilement comprendre et traiter les données qui sont décrites. Après avoir analysé les modèles d'ensembles de données les plus importants, nous déterminons qu'un ensemble de données contient quatre sections principales :

- **Ressources**:Les données brutes peuvent être téléchargé ou accessible directement via des requêtes. Les ressources peuvent être fournies dans différents formats tels que JSON, XML ou RDF.

- **Tags**:Une description plus poussée à propos du contenu et de la structure du jeu de données. Cela peut aller de la représentation textuelle simple à des termes sémantiques riches. Les tags sont la base de pour définir une recherche et découverte de données.

- **Groupes**:Les groupes agissent comme des unités organisationnelles qui partagent une sémantique commune. Ils peuvent être considérés comme un ensemble de jeu de données basés sur des catégories ou des thèmes communs.

- **Organisations**:Les organisations sont une autre façon d'organiser les ensembles de données. Cependant, ils diffèrent des groupes car ils ne partagent pas des propriétés ou une sémantique commune, mais reposent uniquement sur l'association de l'ensemble de données à un tiers d'administration spécifique.

Après une étude poussée des différents modèles de données, nous avons regroupé les informations de métadonnées en huit types principaux. Chaque section décrite ci-dessus doit contenir un ou plusieurs de ces types. Par exemple, les ressources ont

[5] http://ckan.org
[6] http://nucivic.com/dkan/

les types suivants : général, accès, droits, provenance alors que les tags sont : général et provenance seulement. Les huit types d'information sont :

- **Informations Générales** :L'information de base à propos de l'ensemble de données (i.e, titre, description, ID). Le vocabulaire le plus couramment utilisé pour décrire cette information est Dublin Core[7].

- **Informations d'Accès** : Informations sur l'accès et l'utilisation de l'ensemble de données (par exemple, l'URL, le titre de la licence et son URL associée). En plus des propriétés décrites dans les modèles ci-dessus, il y a plusieurs vocabulaires spécialement conçus pour décrire les droits d'accès aux données, par exemple, Linked Data Rights[8], l'Open Digital Rights Language (ODRL)[9].

- **Informations de Droits** : Informations de référence sur l'ensemble de données (par exemple, auteur, responsable et organisation). Les vocabulaires communs utilisés pour exposer des informations de propriété sont Friend-of-Friend (FOAF)[10] pour les personnes et les relations, vCard [**?**] pour les personnes et les organisations et l'Organization ontology [**?**] conçu spécifiquement pour décrire les structures organisationnelles.

- **Informations de Provenance** : Informations temporelle et historique sur la création et mise à jour de l'ensemble de données, en plus des informations sur les versions (par exemple, la données de création, les données sur la mise à jour des métadonnées, dernière version). Le renseignement de la provenance de l'information varie à travers le modèle étudié. Cependant, son importance a conduit à l'élaboration de divers vocabulaires spéciaux comme l'Open Provenance Model[11] et le PROV-O [**?**]. DataID [**?**] est un effort pour fournir des métadonnées sémantiquement riche en mettant l'accent sur la source de l'information, la licence associée et les informations d'accès.

- **L'information Géospatiale** : Information reflétant la couverture géographique de l'ensemble de données représenté avec des coordonnées ou des polygones. Il existe plusieurs modèles et extensions spécifiquement conçus pour exprimer des informations géographiques supplémentaires. La directive "Infrastructure for Spatial Information in the European Community" (INSPIRE)[12] vise à établir une infrastructure d'information spatiale. Des connections ont été établis entre l'EDSC-AP et les métadonnées de INSPIRE. CKAN fournit ainsi une extension

[7] http://dublincore.org/documents/dcmi-terms/
[8] http://oeg-dev.dia.fi.upm.es/licensius/static/ldr/
[9] http://www.w3.org/ns/odrl/2/
[10] http://xmlns.com/foaf/spec/
[11] http://open-biomed.sourceforge.net/opmv/
[12] http://inspire.ec.europa.eu/

spatiale[13] afin d'ajouter des informations géospatiales. Il permet d'importer des métadonnées géospatiales à partir d'autres ressources et fonctionne avec diverses normes (par exemple, ISO 19139) et formats (par exemple, GeoJSON).

- **Information temporelle** : Information reflétant la couverture temporelle de l'ensemble de données (par exemple, de date à date). Il y a eu un travail remarquable sur l'extension CKAN pour inclure des informations temporelles. `govdata.de` est un portail Open Data en Allemagne qui étend le modèle de données CKAN avec des informations comme `temporal_granularity`, `temporal_coverage_to` and `temporal_granularity_from`.

- **Informations statistiques** : des informations statistiques sur les types de données et leur redondance dans les ensembles de données (par exemple, la distribution des propriétés, nombre d'entités et triplets RDF). Cette information est particulièrement utile pour explorer un ensemble de données, car il donne un aperçu détaillé sur les données brutes. VoID est le seul modèle qui fournit des informations statistiques sur un ensemble de données. VoID définit des propriétés pour exprimer différentes caractéristiques statistiques sur un ensemble de données comme le nombre total de triplets, le nombre total d'entités, le nombre total de classes distinctes, etc. Cependant, il existe d'autres vocabulaires tels que SCOVO [**?**] qui peut modéliser et publier des données statistiques sur les ensembles de données.

- **Information de qualité**:L'information qui indique la qualité de l'ensemble de données sur les niveaux de métadonnées. En plus de cela, un ensemble de données devrait inclure une échelle qui mesure le respect des normes de publication des standards des Linked Data [**?**]. Une information de qualité est seulement exprimée dans les métadonnées de POD. Cependant, `govdata.de` étend le modèle de CKAN pour y inclure un champ `ratings_average`. En outre, il existe plusieurs autres vocabulaires comme daQ [**?**] qui peuvent être utilisés pour exprimer la qualité des ensembles de données. Le RDF Review Vocabulary[14] peut également être utilisé pour exprimer les commentaires et évaluations sur l'ensemble de données et de ses ressources.

Alors que l'utilisation d'un vocabulaire ou d'un modèle commun est la clé de la communication, nous avons identifié le besoin d'un modèle harmonisée de métadonnées contenant suffisamment d'informations pour que les consommateurs puissent facilement comprendre et appréhender les données. Pour créer les correspondances entre les différents modèles, nous avons procédé en plusieurs étapes :

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

- Examiner tous les modèles, les spécifications des vocabulaires et les documentations.

- Examiner les ensembles de données existantes en utilisant ces modèles et vocabulaires. Data Portal[15] fournit une liste complète des portails ouverts de données du monde entier. Ca a été notre point de départ pour trouver les portails utilisant CKAN ou DKAN comme système de gestion de données (DMS). Socrata, par exemple, maintient une liste des portails Open Data utilisant leur logiciel sur leur page d'accueil comme http://pencolorado.org et http://data.maryland.gov.

- Examiner le code source de certains portails. Ce fut particulièrement le cas pour Socrata, car leur API renvoie les données brutes sérialisées en JSON plutôt que les métadonnées de l'ensemble de données. En conséquence, nous avons dû étudier le code source de l'API Socrata Open Data (SODA)[16] pour récupérer les différentes classes et interfaces.

De notre étude, nous avons constaté qu'une bonne intégration des données de l'Open Data dans les entreprises nécessite une amélioration des ensembles de données pour inclure les informations suivantes :

- **informations d'accès** : un ensemble de données est inutile s'il ne contient pas de mécanismes pour récupérer les données, ou de point central pour effectuer des requêtes.

- **informations de licence** : les entreprises sont toujours préoccupées par les implications juridiques de l'utilisation du contenu externe. En conséquence, les ensembles de données devraient inclure à la fois des informations compréhensible par les individus mais aussi automatiquement par des logiciels, comme les informations de permissions, droits d'auteur et attributions.

- **information de provenance** : en fonction de la licence du jeu de données, les données pourraient ne pas être légalement utilisable s'il n'y a pas d'information décrivant les auteurs et les modifications effectuées. Les modèles actuels ne respectent pas ces contraintes, limitant de fait l'utilisation de nombreux ensembles de données.

Nous avons identifié quatre sections principales qui devraient être incluses dans le modèle : les ressources, les groupes, les tags et les organisations. En outre, nous avons défini huit types pour classifier l'information. Notre principale contribution est

[15] http://dataportals.org
[16] https://github.com/socrata/soda-java/tree/master/src/main/java/com/socrata/model

la définition de correspondances entre chacune des propriétés de ces modèles. Ceci a conduit à la conception de HDL, un modèle de données harmonisée, qui prend le meilleur des modèles pour assurer une couverture complète de métadonnées facilitant la découverte, exploration et la réutilisation de données.

10.5.2 Génération et Validation de Profils de Données

La nature hétérogène des sources de données influe directement sur la qualité des données, car elles contiennent souvent des incohérences ainsi que des métadonnées incomplètes et mal interprétées. En outre, la variation significative de la taille, de formats et la fraîcheur des données rend plus difficile la recherche d'ensemble de données utiles sans connaissance préalable. On remarque très bien ce point dans le cloud de Linked Open Data où quelques ensembles de données tels que DBPedia [**?**], Freebase [**?**] et YAGO [**?**] sont favorisés par rapport à des ensembles de données moins populaires, mais qui sont plus adaptes avec un domaine spécifique pour les tâches à accomplir. Par exemple, pour construire des systèmes de recommandation à partir d'une bibliothèque numérique universitaire dans le cloud LOD, les ensembles de données populaires comme le Semantic Web Dog Food[17], DBLP[18] ou Yovisto[19] peuvent être favorisés par rapport à des ensembles de données moins connus, mais plus spécifiques comme VIAF[20] qui relie les fichiers d'autorité de 20 bibliothèques nationales, une liste des sujets des titres des bibliothèques publiques en Espagne[21] ou les recherche de thèse françaises[22].

Les utilisateurs explorent des ensembles de données dans des portails en se basant sur les métadonnées proposées par le propriétaire de l'ensemble de données ou l'administrateur du portail de données. Cette information est principalement disponible sous la forme de tags prédéfinis tels que *médias, géographie, sciences de la vie* qui sont utilisés à des fins d'organisation et de regroupement. Cependant, la diversité croissante de ces ensembles de données rend plus difficile le classement dans un nombre restreint de tags, qui sont subjectivement attribués sans jamais capturer l'essence et l'étendue de l'ensemble de données [**?**]. En outre, l'augmentation du nombre d'ensembles de données disponible rend l'analyse manuelle et la conservation des métadonnées insoutenable, même quand cette tache est confiée à des communautés.

Roomba est un outil que nous proposons pour adresser les problématique de validation et de génération automatique de profils d'ensemble de données descriptifs. C'est un framework extensible qui consiste en une exécution structurée combinant des techniques d'identification de portails de données, de récupération de jeux de

[17] http://datahub.io/dataset/semantic-web-Dog-alimentaire
[18] http://datahub.io/dataset/dblp
[19] http://datahub.io/dataset/yovisto
[20] http://datahub.io/dataset/viaf
[21] http://datahub.io/dataset/lista-encabezamientos-materia
[22] http://datahub.io/dataset/thesesfr

données, et d'un ensemble de modules assemblable en combinant plusieurs tâches. Le framework valide les métadonnées attachées à l'ensemble de données en les confrontant à un ensemble de données agrégées. Les champs des métadonnées sont automatiquement corrigés quand c'est possible (par exemple, ajout de l'URL d'une licence manquante). En outre, un rapport décrivant tous les problèmes qui ne peuvent être automatiquement résolus est créé pour être envoyé par courriel au responsable de l'ensemble de données. Il existe différents outils de profilage statistiques pour les données relationnelles et les Linked Data. L'architecture du framework permet de facilement intégrer ces outils pour approfondir le profilage. Cependant, dans cette section, nous nous concentrons sur le profilage des métadonnées d'un jeu de données en ignorant les tâches de profilage statistique. Nous validons notre framework à l'aide d'un ensemble de profils créés manuellement et vérifions manuellement la précision des résultats par rapport à ceux fournis par des portails bases sur CKAN.

Roomba est proposé comme un outil en ligne de commande (CLI) en utilisant Node.js et est disponible sur la plateforme GitHub[23]. Roomba permet aux administrateurs de portail de données comme **Dan** de:

- Récupérer des informations sur le système de gestion des données du portail
- Récupérer toutes les informations sur les ensembles de données à partir d'un portail
- Récupérer toutes les informations de groupe à partir d'un portail de données
- Parcourir, chercher et mettre en cache des jeux de données (un ensemble de données spécifiques, des ensembles de données d'un groupe, des ensembles de données sur l'ensemble du portail)
- Exécuter un programme d'agrégation sur un groupe spécifique ou sur le portail entier
- Analyser un ensemble de données spécifique, un ensemble du groupe ou le portail entier

La Figure **??** montre les principales étapes :

- **Identification de système de gestion de données**: L'identificateur de portail repose sur plusieurs techniques de récupération lors de la phase d'identification. Elle comprend une combinaison d'inspection de l'URL, une revue des métadonnées, et une inspection du Document Object Model (DOM).

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

- **Extraction des métadonnées** : Après avoir identifié la plateforme sous-jacente du portail, le module d'extraction des métadonnées envoie des requêtes a l'API afin de récupérer l'ensemble des métadonnées puis les entrepose dans un système de cache. Selon la plateforme utilisée, l'extracteur peut lancer des taches spécifiques. Par exemple, dans les portails basés CKAN, l'extracteur est capable de parcourir et extraire les métadonnées d'un ensemble de données spécifique, tous les ensembles de données d'un groupe donné (par exemple, LOD cloud) ou tous les jeux de données du portail.

- **Extraction des ressources et des instances**: Des métadonnées extraites, le module d'extraction de ressources est capable d'identifier toutes les ressources associées a un jeu de données. Ils peuvent avoir différents types comme une instance SPARQL, API, fichier, visualisation, etc. Cependant, avant d'extraire l'intégralité des ressources d'un serveur, et considérant que l'ensemble des données contient potentiellement de grandes quantités de ressources et que la puissance de calcul est limitée pour certains serveur, un sous-module permet de récupérer des échantillons. Différentes stratégies à base d'échantillons existent, et peuvent produire des résultats précis, même avec des petits échantillons d'environ 10% des données [**?**].

- **Validation du profil** : Le module de validation du profil (composant (iv)) identifie les informations manquantes et a la capacité de les corriger automatiquement. Chaque ensemble de métadonnées (général, accès, possession et provenance) est validé et corrigé automatiquement lorsque c'est possible. Chaque profilage vérifie les champs de métadonnées qu'il peut remplir. Le processus de validation vérifie si chaque champ est défini et si la valeur attribuée est valide.

  Il existe de nombreuses validations pour différents domaines. Par exemple, les adresses électroniques et les URL doivent être validées pour garantir que la valeur entrée est syntaxiquement correct. En plus de cela, pour les URL, le module de validation de profil émet une requête HTTP `HEAD` afin de vérifier si cette URL est accessible. Le module utilise également les informations du `content-header` d'une réponse valide pour extraire, comparer et corriger certaines valeurs de métadonnées des ressources comme `mimetype` et `size`.

- **Génération du profil et de rapports** : Le processus de validation met en évidence les informations manquantes et les présente dans un rapport qui peut être automatiquement envoyé par email au mainteneur si son adresse est renseignée dans les métadonnées. En plus du rapport généré, les profils améliorés sont publiés dans un format JSON en utilisant le modèle de données de CKAN et sont publiquement accessible[24].

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

L'état actuel du rapport de cloud LOD [?] indique que le cloud LOD contient 1014 jeux de données. Ils ont été récoltés via un robot de LDSpider [?] à partir de 560 000 URI. Roomba récupère les ensembles de données hébergés dans les portails dont les métadonnées sont pertinentes. Nous nous sommes appuyés sur les informations fournies par l'API DataHub CKAN. En examinant les tags disponibles, nous avons trouvé deux groupes candidats. Le premier, marqué avec "lodcloud", contient 259 ensembles de données, tandis que le second, marqué avec "lod" retourne seulement 75 ensembles de données. Après avoir examiné manuellement les deux listes, nous avons découvert que les jeux de données groupés avec la balise "lodcloud" sont les bons car ils contenaient des métadonnées plus récente et précise. Pour aller plus loin, nous avons utilisé d'autres portails CKAN. Nous avons utilisé `dataportals.org`, qui contient une liste complète des portails Open Data à travers le monde. Nous avons choisi le portail de données d'Amsterdam[25], car il est fréquemment mis à jour et bien maintenu. Le portail a été commandé en 2012 par le Amsterdam Economic Board Open Data Exchange (ODE), et couvre un large éventail de domaines d'information (énergie, économie, éducation, développement urbain, etc.) sur Amsterdam.

Dans notre évaluation, nous nous sommes concentrés sur deux aspects: i) *la pertinence du profilage* qui évalue manuellement la validité des erreurs générées dans le rapport, et ii) *l'exhaustivité du profilage* qui évalue si les profils couvrent toutes les erreurs des métadonnées des ensembles de données.

Notre évaluation a montré que Roomba propose une pertinence et une exhaustivité accrue pour les propriétés examinées. En conséquence, nous avons exécuté Roomba sur le cloud LOD hébergé dans le DataHub. Nous avons découvert que de nombreux ensemble de données nécessite des corrections. La plupart d'entre eux n'ont pas d'information d'accès et leurs ressources souffrent de faible disponibilité. Ces deux mesures sont d'une grande importance pour les entreprises qui cherchent à intégrer et utiliser des données externes. Nous avons découvert que l'information la plus erronée est l'information de possession, puisque cette information est manquante ou indéterminée pour 41% des ensembles de données. Les ressources de jeux de données ont les métadonnées les plus pauvres : 64% des métadonnées générale, toutes les informations d'accès et 80% de l'information sur la provenance des valeurs est manquante ou indéterminée. Nous avons également montré que le processus de correction automatique peut effectivement améliorer la qualité de certaines informations. Nous croyons qu'il y a un besoin d'avoir un effort communautaire pour corriger manuellement les informations manquantes importantes comme des informations de possession (mainteneur, auteur, email de contact).

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

10.5.3 Èvaluation Objective de la Qualité des Données Associées

Nous entrons dans une ère où l'ouverture devient le standard. Les gouvernements, les universités, les organisations et même les individus publient d'énormes quantités de données ouvertes. Cette ouverture doit être accompagnée d'un certain niveau de confiance, ainsi que des garanties sur la qualité des données. Le Linked Open Data est une mine d'or pour ceux qui essaient de tirer parti de sources de données externes afin de produire des décisions plus éclairées [**?**]. Cependant, l'hétérogénéité des sources impacte directement la qualité des données, car les sources contiennent souvent des informations mal interprétées et incomplètes.

La qualité des données est un domaine d'étude minutieux avec de nombreux points d'études et des frameworks pour en apprécier ses dimensions [**?**, **?**, **?**]. Les principes de qualité des données reposent généralement sur de nombreux indicateurs subjectifs qui sont complexes à mesurer. La qualité des données est réellement appréciée lors de son utilisation [**?**], donc est directement liée à la capacité de satisfaire les besoins des utilisateurs.

Les documents web qui sont par nature non structurés et interconnectes nécessitent des mesures de la qualité et des techniques d'évaluation différentes des ensembles de données traditionnels. Par exemple, l'importance et la qualité des documents web peuvent être subjectivement calculés par des algorithmes tels que le Page Rank [**?**]. Malgré le fait que la qualité des Linked Open Data est un sujet actuellement très demandé, très peu d'efforts sont en cours pour essayer de normaliser, suivre et formaliser les frameworks pour fournir des évaluations et des certificats qui aideront les consommateurs de données dans leurs tâches d'intégration.

L'évaluation de la qualité des données est un processus qui évalue si un morceau de donnée est conforme aux besoin dans un cas d'utilisation précis [**?**]. La dimension de la qualité des données est liée aux exigences qu'en ont les utilisateurs. Par exemple, DBpedia [**?**] et YAGO [**?**] sont des bases de connaissances contenant des données extraites de sources structurées et semi-structurées. Ils sont utilisés dans une variété d'applications comme par exemple, des systèmes d'annotation [**?**], des systèmes d'exploration et de recherche [**?**] et des moteurs de recommandation [**?**]. Cependant, leurs données ne sont pas intégrées dans des systèmes critiques : applications médicales, applications aéronautiques, etc. La qualité des données est jugée insuffisante.

L'idée de base à propos des Linked Data est qu'au plus elle est interconnectée avec d'autres ensembles de données, au plus son intérêt grandit. Tim Berners-Lee a défini quatre grands principes pour la publication de données qui peuvent assurer un certain niveau d'uniformité qui reflète directement leur utilisabilité [**?**]:

- Utiliser des URIs pour identifier les ressources.

- Utiliser des URIs HTTP pour accéder de manière uniformisée aux ressources.
- Fournir des représentations en utilisant des langages et des protocoles standards (RDF, SPARQL).
- Inclure des liens pour permettre de découvrir de nouvelles ressources.

Fort de ces principes, nous regroupons la qualité des attributs en quatre catégories principales :

- **Qualité des entités**: indicateurs de qualité qui mettent l'accent sur les données au niveau de l'instance.
- **Qualité de l'ensemble de données**: indicateurs de qualité au niveau de l'ensemble de données.
- **Qualité du modèle sémantique**: indicateurs de qualité qui mettent l'accent sur les modèles sémantiques, les vocabulaires et ontologies.
- **Qualité du processus de liaison**: indicateurs de qualité qui mettent l'accent sur les liens entrants et sortants entre les ensembles de données.

In [**?**], les auteurs ont identifié 24 attributs de qualité de données liées différentes. Ces attributs sont un mélange de mesures objectives et subjectives qui ne peuvent pas être dérivées automatiquement. Dans cet article, nous affinons ces attributs dans un cadre condensé de 10 mesures objectives. Ètant donné que ces mesures sont plutôt abstraites, nous devrions compter sur les indicateurs de qualité qui reflètent la qualité des données [**?**] et les utiliser pour automatiser le calcul de la qualité des ensembles de données.

Les indicateurs de qualité sont pondérés. Ces poids donnent la possibilité de définir plusieurs degrés d'importance. Par exemple, un ensemble de données contenant les gens peuvent avoir plus d'une personne portant le même nom et il n'est donc pas toujours vrai que deux entités dans un jeu de données ne devraient pas avoir le même label. En conséquence, le poids de cet indicateur de qualité sera mis à zéro et ne n'affectera pas la note globale de qualité pour la mesure de la cohérence.

Les indicateurs indépendants pour la qualité de l'entité sont essentiellement subjectifs: par exemple, le niveau de détail sur les objets représentés, leur portée etc. Les entités sont régies par le modèle sous-jacent, et nous avons donc regroupé les indicateurs avec ceux de la qualité de modélisation.

Les mesures et l'objectif des indicateurs de qualité ont été recueillis en :

- Transformant et consolidant les indicateurs de qualité objectifs à partir d'une série de questions [**?**].

- Analysant et comparant les différentes approches et outils sur la qualité des données.

- Examinant les propriétés des modèles de Linked Data les plus courants à partir de l'étude conduite dans [**?**].

Nous avons étendu Roomba avec 7 sous-modules qui vérifient les différents indicateurs de qualité de l'ensemble de données. Certains indicateurs doivent être examinés sur un ensemble fini. Puisque Roomba fonctionne sur des portails bases sur CKAN, nous avons construit notre extension de qualité pour calculer les scores à partir du modèle standard de CKAN.

Roomba couvre 82% des indicateurs de qualité objectifs proposés. Sur la base de nos expérimentations avec Roomba sur le cloud LOD, nous avons découvert que la majorité des ensembles de données ont besoin de corrections. La plupart d'entre eux ne renseignent que très mal la provenance, licence d'utilisation, et indicateurs de qualité.

10.6 Vers des Données d'entreprise Agrémentées

10.6.1 Intégration des Données dans l'entreprise

Les entreprises effectuent traditionnellement des analyses et des rapports à partir de bases relationnelles. Les données d'entreprise disponibles pour les décideurs étaient généralement la gestion des clients ou les données de planification et ressources d'entreprise (ERP). Cependant, les informations en provenance des réseaux sociaux, les blogs, les données des capteurs, ou les données publiées par les gouvernements et organisations internationales sont de plus en plus accessibles [**?**].

La qualité et la quantité de connaissance structurée disponible rendent désormais possible pour les entreprises de parcourir cette énorme quantité de données publiques et de l'intégrer dans leurs systèmes d'information. L'analyse de ces nouvelles données dans le contexte des entreprises devrait leur apporter de nouvelles approche pour mieux appréhender l'exploration et le ciblage de nouveaux marches [**?**].

Ces nouvelles sources distribuées, cependant, posent des défis énormes. Elles ont différents formats de fichiers, protocoles d'accès, ou encore différents langages de requête. Elles possèdent leur propre modèle de données avec différentes façons de représenter et stocker les données. Les données à travers ces sources peuvent être dupliquées, incohérentes ou être sémantiquement similaire et pourtant différentes. L'intégration et la définition d'une vue unifiée pour ces structures de données hétérogènes et complexes nécessitent donc des outils puissants pour cartographier et organiser les données.

Ètablir des bases de connaissances de données dans l'entreprise peut faciliter la fourniture de services d'intégration de données [**?**]. Dans cette section, nous présentons notre travail en utilisant DBpedia comme une base de connaissances interne. Nous présentons un ensemble de services que nous avons mis en place audessus de DBpedia permettant une amélioration des schémas de correspondance et une distinction des entités. Ces services permettent aux utilisateurs de combiner semi automatiquement des données potentiellement mal formées disponible dans des silos hétérogènes. Les données sémantiquement liées sont identifiées et des suggestions sont proposées aux utilisateurs. En fonction des besoins de l'utilisateur, les données sont agrégées et peuvent être visualisées directement ou exportées vers des outils de reporting de Business Intelligence. Enfin, nous procédons à une retro-ingénierie du système de Knowledge Graph de Google pour en retirer les propriétés les plus pertinentes pour une entité. Nous comparons ces résultats à l'aide un sondage mené sur 152 utilisateurs et nous montrons comment nous pouvons représenter ces connaissances en utilisant le vocabulaire de Fresnel.

La correspondance entre les schémas est généralement utilisée dans les intégrations inter-entreprises, les correspondances de méta-modèle, ainsi que dans les extractions de données (lors de processus ETL). Pour les non-spécialistes en système d'information, la manière de comparer les données financières sur différentes années par exemple, est de copier et coller les données d'une feuille de calcul Excel dans une autre, créant ainsi des redondances et potentiellement des erreurs de copier-coller. En utilisant les techniques de correspondance entre schémas, il est possible de proposer un processus semi-automatise, comme par exemple déterminer des colonnes similaires et les proposer à l'utilisateur lors de l'intégration. Cette intégration peut être faite avec des outils d'intelligence décisionnelle appropriés qui proposent des visualisations.

Un des problèmes dans l'intégration est la qualité des données. Les colonnes peuvent contenir des données mal formées ou incorrectes. Il peut aussi ne pas avoir d'information appropriée dans les en-têtes de colonne, limitant de fait l'identification lors de l'intégration. Un certain nombre d'approches exploitent les similitudes d'entêtes ou les similitudes des types de données de ces colonnes. Nous avons proposé une nouvelle approche qui exploite un typage sémantique riche fourni par notre outil de détermination d'entité.

10.6.1.1 Rapprochement des Données

La réconciliation permet de différencier les entités, à savoir les cellules correspondant à des entités typées dans le cadre de tableau de données. Google Refine supporte la réconciliation avec Freebase, mais nécessite une confirmation de l'utilisateur. Pour les bases de données de moyennes et grandes tailles, cela peut être très chronophage. Pour réconcilier les données, nous identifions dans un premier temps les colonnes

candidates à la réconciliation en sautant les colonnes contenant des valeurs ou des dates numériques. Nous utilisons ensuite l'API de différentiation sur les colonnes cibles et sources afin d'en déterminer une liste de type possible. Les résultats sont mis en cache afin d'être récupérés par nos algorithmes de comparaison sur la similarité.

10.6.1.2 Identification des Colonnes Non-nommées et Non Typées

L'AMC a la capacité de combiner les résultats de différents algorithmes d'unification. Son algorithme par défaut se base sur les en-têtes des colonnes pour produire un score de similarité entre les schémas des éléments. Il a été prouvé que la combinaison de différents algorithmes augmente considérablement la qualité des résultats correspondant [**?**][**?**]. Toutefois, lorsque les en-têtes sont manquantes ou ambiguës, l'AMC ne peut qu'exploiter des algorithmes au niveau de l"intersection et l'inclusion basé sur les données de la colonne. Nous avons donc mis en place trois nouveaux algorithmes de similarité qui exploitent les types riches extraits des Linked Data en vue d'améliorer les résultats des colonnes non-nommées et non typées. Ils sont présentés ci-dessous.

- **Similarité cosinus** : On compare le vecteur de résultat des types potentiels à partir de la colonne source avec le vecteur de résultat de la colonne cible. Les similitudes s entre les paires de colonnes peuvent être calculées en utilisant la valeur absolue de la fonction cosinus de similitude.

- **Pearson Product-Moment Correlation Coefficient (PPMCC)** : Le deuxième algorithme que nous avons implémenté est PPMCC, une mesure statistique de l'indépendance linéaire entre deux variables (x, y) [**?**]. L'entrée pour PPMC se compose de deux tableaux qui représentent les valeurs des colonnes source et cible, où la colonne source est la colonne avec le plus grand ensemble de types riches trouvé.

- **Corrélation de Spearman** : Le dernier algorithme que nous avons implémenté pour identifier les colonnes non-nommées et non typées est le coefficient de corrélation de Spearman. Il applique une transformation sur la position des données d'entrée et calcule le PMCC sur les données triées. Dans nos expérimentations, nous avons utilisé l'ordre naturel avec des stratégies par défaut pour le arbitrer les valeurs NaN et sans réponse précise. L'algorithme de classement est toutefois configurable et peut être amélioré en utilisant des mesures plus sophistiquées.

10.6.1.3 Ètiquetage des Colonnes

Nous avons montré dans la section précédente comment faire correspondre des colonnes non-nommées et non typées. L'étiquetage de la colonne est cependant bénéfique car les résultats de nos précédents algorithmes peuvent être combinés avec des techniques

traditionnelles de correspondance au niveau de l'en-tête à des fins d'améliorations sur la qualité des correspondances.

Les typages forts extraits de Freebase sont indépendants les uns des autres. Nous avons besoin de trouver une méthode qui permette de déterminer un score normalisé pour chaque type dans l'ensemble en équilibrant la proportion de scores élevés avec celles des plus faibles à l'aide de l'intervalle de Wilson avec un paramètre de Bernoulli.

10.6.1.4 Gestion des Valeurs typées

Jusqu'à présent, nous avons utilisé plusieurs méthodes pour identifier les similitudes entre des valeurs de type de chaine de caractères, mais nous devons aussi pouvoir déterminer le type d'autres valeurs numériques telles que les dates, l'argent, la distance, etc. A cet effet, nous avons mis en place un identificateur de type de base qui peut reconnaître dates, argent, valeurs numériques, ou des chiffres utilisés comme identifiants. Cela nous aidera à mieux déterminer les similitudes entre données. Le réglage des algorithmes de combinaison AMC peut être d'une grande importance à cette étape. Par exemple, l'attribution de pondérations à différents identificateurs et peaufiner la configuration peut donner des résultats plus précis.

10.6.2 Propriétés Importantes pour les Entités

Les entités sont généralement décrites avec beaucoup de propriétés. Cependant, toutes les propriétés n'ont la même importance. Certaines propriétés sont considérées comme cruciales pour identifier une correspondance, alors que d'autres sont généralement choisies pour rapidement fournir un résumé des faits principaux attachés à une entité. Contrairement aux entités, il est difficile d'évaluer les propriétés qui sont les plus "importantes".

Le Web Scraping est une technique pour extraire des données à partir de pages web. Nous voulons capturer les propriétés décrites dans le Google Knowledge Panel (GKP) qui sont injectées dans les pages de résultats de recherche [**?**]. Nous avons développé une application Node.js qui interroge tous les concepts DBPedia qui ont au moins une instance qui est `owl:sameAs` avec une ressource Freebase (car Freebase est la base de connaissances derrière le GKP) afin d'augmenter la probabilité que la page de résultats du moteur de recherche (SERP) pour cette ressource contiendra un GKP. Nous supposons dans nos expérimentations que les propriétés affichées pour une entité sont dépendants du type et du contexte (pays, temps, requête) qui peuvent affecter les résultats. En outre, nous filtrons les concepts génériques en excluant ceux qui sont une sous-classe directe de `owl:Thing` car ils vont déclencher des requêtes ambigües. Nous avons obtenu une liste de 352 concepts[26].

[26] https://github.com/ahmadassaf/KBE/blob/master/results/dbpediaConcepts.json

Algorithm 2 Retro-enginerie de l'algorithme du Google Knowledge Panel

```
1: INITIALIZE equivalentClasses(DBpedia, Freebase) AS vectorClasses
2: Upload vectorClasses for querying processing
3: Set n AS number-of-instances-to-query
4: for each conceptType ∈ vectorClasses do
5:    SELECT n instances
6:    listInstances ← SELECT-SPARQL(conceptType, n)
7:    for each instance ∈ listInstances do
8:       CALL http://www.google.com/search?q=instance
9:       if knowledgePanel exists then
10:         SCRAP GOOGLE KNOWLEDGE PANEL
11:      else
12:         CALL http://www.google.com/search?q=instance + conceptType
13:         SCRAP GOOGLE KNOWLEDGE PANEL
14:      end if
15:      gkpProperties ← GetData(DOM, EXIST(GKP))
16:   end for
17:   COMPUTE occurrences for each prop ∈ gkpProperties
18: end for
19: gkpProperties
```

Fresnel[27] est un vocabulaire pour afficher des données RDF. Il spécifie *quelles* informations contenues dans un graphe RDF devraient être présentée avec le concept de base `fresnel:Lens` [**?**]. PROV-O[28] est un vocabulaire pour décrire les métadonnées sémantiquement riches en mettant l'accent sur les détails de la provenance, la licence et l'accès. Nous utilisons ces deux vocabulaires pour représenter explicitement les propriétés devant être représentées lors de l'affichage d'une entité[29]. Cet ensemble de données peut désormais être réutilisé comme une configuration.

10.6.3 Agrégateur de Nouvelles Sociale sémantique

Avec les progrès rapides de l'Internet, les médias sociaux deviennent de plus en plus prévalent dans nos vies quotidiennes. La nature omniprésente des appareils compatibles Web, en particulier les téléphones mobiles, permet aux utilisateurs de participer et d'interagir en de nombreuses formes différentes comme avec le partage de photos et vidéos, les forums, groupes de discussion, blogs, micro-blogs, services de favoris, et les services basés sur la localisation. Les réseaux sociaux ne rassemblent pas seulement les utilisateurs d'Internet en groupes d'intérêts communs, ils aident aussi les gens á suivre les informations, contribuer aux débats en ligne ou apprendre des autres. Ils sont en train de transformer l'utilisation du Web en termes de comportement, da recherche, da navigation et de comportement d'achat [**?**].

[27] http://www.w3.org/2005/04/fresnel-info/
[28] http://www.w3.org/TR/prov-o/
[29] https://github.com/ahmadassaf/KBE/blob/master/results/results.n3

Un scénario qui arrive souvent en lisant un article intéressant, regardant une belle vidéo ou participant à une discussion dans un forum est l'intérêt croissant de chercher des informations supplémentaires en lien avec le sujet initial. Pour ce faire, les utilisateurs peuvent utiliser Twitter, Google+ ou YouTube. Ils peuvent essayer plusieurs fois avec plusieurs mots-clés pour obtenir les résultats souhaités. En fin de compte, ils pourraient se retrouver avec plusieurs onglets du navigateur ouverts et se laisser distraire par la surcharge d'information de toutes ces ressources. La même chose arrive en entreprises lorsque les utilisateurs sont intéressés par des informations fournies par les applications web d'entreprise. Dans cette section, nous présentons SNARC, un agrégateur de nouvelles sociale sémantique qui tire parti des riches données que les réseaux sociaux permettent de construire avec un expérience interactive sur l'Internet et les intranets. Le service récupère les nouvelles liées à la page en cours à partir des plates-formes populaires comme Twitter, Google+, YouTube, Vimeo, Slideshare, StackExchange et le Web. Comme une possible implémentation, nous avons créé une extension Google Chrome qui enrichit l'expérience des utilisateurs en augmentant l'information contextuelle liée à des entités sur la page elle-même, ainsi que l'affichage nouvelles sociale connexe sur une barre latérale flottante.

Le back-end de SNARC se compose de trois éléments principaux : un gestionnaire de document qui crée un "modèle sémantique" représentant une ressource Web, une couche de requête responsable de la diffusion des requêtes sur les services sociaux supportes et un analyseur de données qui traite le résultat des requêtes, les enveloppe dans un modèle social commun et génère la sortie désirée.

10.7 Réalisations

Cette thèse décrit en détail les différentes étapes visant à réaliser la mise à disposition de données en libre-service dans l'entreprise. Le travail présenté est bénéfique pour nos deux personnages introduits. Les contributions sont :

10.7.1 Contributions pour les Administrateurs de Portail de Données

Notre administrateur de portail de données **Paul** cherche toujours à élargir le nombre d'ensemble de de données hébergées, sans compromettre la qualité des celles-ci. Dans la section **??**, nous avons étudié différents modèles et vocabulaires qui décrivaient les ensembles de données sur le web. Nous avons trouvé une faiblesse lorsqu'on a besoin de travailler avec un modèle de données descriptif complet, avec notamment des informations d'accès, de licence et de provenance. En conséquence, nous avons proposé un modèle pour données harmonisé (HDL) que **Paul** peut utiliser comme base de travail pour ensuite étendre et présenter les ensembles de données qu'il contrôle.

Paul connait aussi les principaux modèles de données et les types de métadonnées devant être ajoute pour valoriser au mieux ses jeux de données. Les correspondances proposées par l'approche lui permettent d'intégrer facilement des données provenant de divers systèmes de gestion des données dans son propre portail.

Dans la section **??**, nous avons proposé Roomba, un outil de génération automatique de profils de données et de validation qui peut être facilement étendu pour effectuer diverses tâches de profilage. En natif, **Paul** peut utiliser Roomba pour réparer automatiquement des problèmes de métadonnées, et aviser les responsables des données des problèmes à fixer manuellement.

Dans la section **??**, nous avons proposé un cadre global de la qualité objective appliquée au Linked Open Data. En outre, après avoir étudié les outils de qualité de données existantes, nous avons identifié plusieurs faiblesses et la nécessité d'un framework d'évaluation pour mesurer la qualité au niveau de l'ensemble de données. En conséquence, nous avons proposé une extension à Roomba qui couvre 82% des préconisations de qualité objective.**Paul** sera maintenant en mesure d'identifier des données poubelles ou de faible qualité. En plus de cela, les données disponibles dans son portail auront désormais une sémantique riche attachée. Par exemple, l'information temporelle et spatiale extraite sera remplie dans les champs correspondants à HDL. Dans un cas concret, divers ensembles de données seront facilement identifiables pour couvrir différentes parties du Royaume-Uni.

10.7.2 Contributions pour les Analystes de Données

Notre analyste de données **Dan** croit que "plus de données bat les meilleurs algorithmes" et est toujours à la chasse aux données de haute qualité pour fournir des rapports précis à l'équipe de gestion. En examinant les métadonnées riches des ensembles de données présentées dans HDL, il sera en mesure de prendre des décisions rapides pour inclure ou exclure un ensemble de données. Il aura également des informations critiques sur les licences, donc sur les possibilités de réutilisation de ces données en interne. Il aura également des indicateurs sur la qualité du jeu de données, qui va l'aider à choisir les meilleurs jeux de données parmi l'ensemble des données disponibles.

Dan sera en mesure d'avoir un accès direct à des descriptions de jeux de données riches et de haute qualité générées par Roomba. En outre, les profileurs intégrés dans Roomba seront en mesure d'identifier les occurrences de termes liés à l'alcool, comme "vin" dans divers ensembles de données. Les méthodes d'expansion de requêtes peuvent être utilisées pour relier l'alcool au vin, lui permettant ainsi de trouver les ensembles de données qu'il recherche.

Dans la section **??**, nous avons présenté une API pour vérifier des entités construite sur SAP HANA. Cette API est utilisée dans RUBIX, un framework que nous avons

proposé pour permettre de connecter des données potentiellement mal formée avec des données externes. **Dan** a maintenant accès à divers ensembles de données qu'il a trouvé en requêtant le portail administré par **Paul**. Il sera également en mesure d'utiliser les schémas des services pour trouver et intégrer ces ensembles de données dans ses rapports.

Après avoir importé les données dans Lumira, il sera également en mesure d'utiliser la base de connaissances interne pour appliquer divers enrichissements sémantiques sur ces données.

Dans la section **??**, nous avons proposé SNARC, un service sémantique d'agrégation de nouvelles sociales qui permet à l'utilisateur d'explorer des nouvelles pertinentes provenant de sources internes ou externes. **Dan** est aussi un homme moderne, qui cherche toujours des informations fraîches et croit en la sagesse de la communauté. Avec les services SNARC intégrés à Lumira, il est aussi capable de voir un flux d'articles de médias sociaux pertinents qui peuvent être d'intérêt pour lui. Il peut suivre des tweets lies à ce qu'il a vu et creuser plus profondément pour trouver des sources fiables d'informations.

En résumé, les contributions apportent de nouvelles briques pour construire un ensemble de services intelligents afin que les analystes trouvent facilement des informations pertinentes. Les administrateurs peuvent plus facilement lutter contre les sources de données non fiables et sont en mesure de maintenir des portails de données de haute qualité. Le travail présenté dans cette thèse va au-delà du simple ajout de métadonnées aux ensembles de données : c'est un ensemble de services qui peuvent automatiquement réaliser ces taches de manière transparente.

10.8 Perspectives

Cette thèse pourrait être étendue dans les directions suivantes :

10.8.1 Représentation du profil des données

Le modèle proposé sur le modèle de données harmonisé (HDL) est actuellement disponible sous forme de fichier JSON hiérarchise. Une amélioration serait d'affiner HDL et de le présenter comme une ontologie OWL à part entière. De plus, le modèle HDL peut être étendu pour proposer des énumérations comme des valeurs pour assurer une représentation fine et unifiée d'un ensemble de données. En outre, alors que nous avons présenté les correspondances entre les différents modèles dans un tableau, la présentation de ces données dans un format lisible par des machines permettrait à divers outils comme Roomba de l'utiliser.

10.8.2 Profilage Automatique des Données

Il a été remarqué que les questions entourant la qualité des métadonnées affectent directement la recherche dans les ensembles de données, alors que les portails reposent sur ces informations pour alimenter leur index de recherche. Il existe diverses extensions à notre outil Roomba qui peut aider dans la construction et l'amélioration automatique de profils de données. Un exemple de ces extensions serait l'intégration des profileurs statistiques et topiques permettant la génération de profils complets. Nous tenons également à étendre Roomba pour le faire fonctionner avec divers portails tels que DKAN ou Socrata. Cette extension peut être faite en s'appuyant sur les modèles de données que nous avons proposés. En plus de tout cela, une amélioration possible serait la capacité à corriger le reste des métadonnées soit automatiquement, soit par le biais d'interfaces intuitives.

10.8.3 Qualité Objective des Linked Data

Assurer la qualité des données des Linked Open Data est un processus complexe car elle se compose d'information structurée soutenue par des modèles, des ontologies et des vocabulaires et contient des liens vers des système de requêtes. Dans cette thèse, nous avons réussi à affiner l'ensemble des problèmes de qualité autour des Linked Data à ceux qui peuvent être objectivement mesurés et évalués par des outils automatiques. Notre outil proposé couvre 85% des indicateurs de qualité proposés. Une extension possible serait d'intégrer des outils d'évaluation de la qualité des modèles, en plus de vérifications syntaxiques au sein de Roomba. Ceci fournirait une couverture complète des indicateurs de qualité proposés. En outre, il n'y a actuellement aucun poids attribués aux indicateurs de qualité. Une contribution légitime serait de proposer des poids à ces indicateurs qui se traduirait par un processus de calcul de la qualité plus objective.

10.8.4 Intégration des Données d'entreprises

Un élément essentiel de l'intégration de données est l'existence de bases de connaissances propres au système d'information de l'entreprise. L'intégration de sources de données supplémentaires de types sémantiques tels que YAGO et comparer nos résultats face à des ontologies telles que OAEI[30] ou islab[31] sont des orientations futures possibles. En outre, notre travail peut être généralisé à la classification des données. De la même façon que l'AMC aide à identifier les meilleures correspondances pour deux ensembles de données, nous avons l'intention de l'utiliser pour identifier les classifications pour un ensemble de données unique base sur des scores normalisés.

[30] http://oaei.ontologymatching.org/2011/instance/index.html
[31] ttp://islab.dico.unimi.it/iimb/