ANGLIA RUSKIN UNIVERSITY

FACULTY OF SCIENCE AND TECHNOLOGY

A NEW APPROACH FOR INTERLINKING AND INTEGRATING SEMI-STRUCTURED AND LINKED DATA

MOHAMED SALAH KETTOUCH

A thesis in partial fulfilment of the requirements
of Anglia Ruskin University for the degree of
Doctor of Philosophy

Submitted: July 2017

To my parents.

Acknowledgements

This dissertation was prepared in part fulfilment of the requirements of the degree of Doctor of Philosophy under the supervision of Dr. Cristina Luca and Dr. Mike Hobbs at Anglia Ruskin University. This Ph.D. journey has been a truly life-changing experience for me and it would not have been possible without the support that I have received from many people.

I would like to start by thanking God. Special thanks to my family, my parents Salah and Fatima Zohra Chellouk, my wife Ouissem, my sisters Soumia and Safia, their husbands Hichem and Said and their beautiful daughters Alaa, Layane, Nada, and Loujayne, my father and mother in law Mohamed Khorief and Nassira Boubrik. Words cannot express how grateful I am to them for all the sacrifices that they have made on my behalf. Your prayer for me was what sustained me thus far.

I would like to express my special gratitude and thanks to my first supervisor Dr. Cristina Luca, she has been a fabulous mentor for me from the first email contact. I would like to thank her for all the encouragement and the trust she showed in me. I would like to show my sincere appreciation to my second supervisor Dr. Mike Hobbs for his motivation and knowledge.

I would also like to thank all my friends Ouassim Lezghad, Abderrezzaq Ghafsi, Khaled (a.k.a Soufiane), Bill Blake and Mohamed Amine Laouissi who believed in me and have been a big moral support to me. I highly acknowledge the help that I received from my friends and colleagues Mark Graham, Dr. Arooj Fatima and William Sawyerr who guided me in many ways throughout these three years of my Ph.D. I am also very grateful to my other colleagues in BRY115 or in COM317, especially Ronak Haddad, Andy Moore and Jonathan Kimmitt. Thank you to Robert Waixel, for helping me with the final proof-reading for this thesis.

Finally, I would like to extend my thanks to anyone who has helped me, in one way or another, during my Ph.D. journey. They include many other staff members at the Department of Computing and Technology at Anglia Ruskin University.

ANGLIA RUSKIN UNIVERSITY

ABSTRACT

FACULTY OF SCIENCE AND TECHNOLOGY

DOCTOR OF PHILOSOPHY

A NEW APPROACH FOR INTERLINKING AND INTEGRATING SEMI-STRUCTURED AND LINKED DATA

MOHAMED SALAH KETTOUCH

JULY 2017

This work focuses on improving data integration and interlinking systems targeting semi-structured and Linked Data. It aims at facilitating the exploitation of semi-structured and Linked Data by addressing the problems of heterogeneity, complexity, scalability and the degree of automation.

Technologies, such as the Resource Description Framework (RDF), enabled new data spaces and concept descriptors to define an increasing complex and heterogeneous web of data. Many data providers, however, continue to publish their data using classic models and formats. In addition, a significant amount of the data released before the existence of the Linked Data movement have not emigrated and still have a high value. Hence, as a long term solution, an interlinking system has been designed to contribute to the publishing of semi-structured data as Linked Data. Simultaneously, to utilise these growing data resource spaces, a data integration middleware has been proposed as an immediate solution.

The proposed interlinking system verifies in the first place the existence of the Uniform Resource Identifier (URI) of the resource being published in the cloud in order to establish links with it. It uses the domain information in defining and matching the datasets. Its main aim is facilitating following best practice recommendations in publishing data into the Linked Data cloud. The results of this interlinking approach show that it can target large amounts of data whilst preserving good precision and recall.

The new approach for integrating semi-structured and Linked Data is a mediator-based architecture. It enables the integration, on-the-fly, of semi-structured heterogeneous data sources with large-scale Linked Data sources. Complexity is tackled through a usable and expressive interface. The evaluation of the proposed architecture shows high performance, precision and adaptability.

keywords: Semantic Web, Linked Data, Data Integration, Data Interlinking, Instance Matching, Interoperability.

Contents

List of Figures

List of Tables

List of Algorithms

List of Definitions

List of Examples

Glossary and Abbreviations

API Application Program Interface

DS Data Source

GS Global Schema

HDT Header Dictionary Triples

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

IRI Internationalised Resource Identifier

JSON JavaScript Object Notation

LOD Linked Open Data

OWL Web Ontology Language

RDF Resource Description Framework

RDFS Resource Description Framework Schema

REST REpresentational State Transfer

SPARQL Protocol and RDF Query Language

URI Uniform Resource Identifier

W3C World Wide Web Consortium

XML Extensible Markup Language

XSLT Extensible Stylesheet Language Transformations

List of Related Publications

Fatima, A., Luca, C., Wilson, G. and Kettouch, M. 2015. Result optimisation for federated sparql queries. In: *Modelling and Simulation (UKSim), 2015 17th UKSim-AMSS International Conference on*. IEEE, pp. 491–496.

Kettouch, M., Luca, C. and Hobbs, M. submitteda. Linkd: Element-based data interlinking of rdf datasets in linked data. *Web Semantics: Science, Services and Agents on the World Wide Web* .

Kettouch, M., Luca, C., Hobbs, M. and Dascalu, S. in press. Using semantic similarity for schema matching of semi-structured and linked data. In: *Internet Technologies and Applications (ITA), 2017*. IEEE. 79

Kettouch, M., Luca, C., Khorief, O., Wu, R. and Dascalu, S. 2017a. Semantic data management in smart cities. In: *Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), 2017 International Conference on*. IEEE, pp. 1126–1131. xii, 8, 76, 150, 151, 152

Kettouch, M. S., Luca, C. and Hobbs, M. 2015a. An interlinking approach based on domain recognition for linked data. In: *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. IEEE, pp. 488–491. 79, 107

Kettouch, M. S., Luca, C. and Hobbs, M. 2017b. Schema matching for semi-structured and linked data. In: *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*. IEEE, pp. 270–271. 79, 83

Kettouch, M. S., Luca, C. and Hobbs, M. submittedb. Semild: Mediator-based framework for keyword search over semi-structured and linked data. *Journal of Intelligent Information Systems* .

Kettouch, M. S., Luca, C., Hobbs, M. and Fatima, A. 2015b. Data integration approach for semi-structured and structured data (linked data). In: *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. IEEE, pp. 820–825. 116

Kettouch, M. S., Luca, C. and Khorief, O. 2016. A framework for integrating and publishing linked data in smart cities. In: *Proceedings of The International Conference on Communications, Computer Science and Information Technology*. 44, 76

Wu, R., Hossain, M., Painumkal, J., Kettouch, M. S., Luca, C., Dascalu, S. and Harris, F. in press. Web-service framework for environmental models. In: *Internet Technologies and Applications (ITA), 2017*. IEEE. 34

A NEW APPROACH FOR INTERLINKING AND INTEGRATING SEMI-STRUCTURED AND LINKED DATA

MOHAMED SALAH KETTOUCH

COPYRIGHT

Chapter 1

Introduction

Research Is to See What Everybody Else Has Seen

and Think What Nobody Has Thought

Albert Szent-Györgyi

1.1 Setting the Scene

The ambition of the W3C[1] and the Web experts to develop new functionalities, and the expectations of Web users for better usability are growing. Despite recent progress there continues to be a need for further automation of certain Web tasks (as will be shown in Section 7.3.4).

The World Wide Web has been a major technological achievement that improved the way people publish and access information. It was the result of the ”marriage” of many technological breakthroughs in the late 1980s, such as HTTP[2], HTML[3] and the URI[4]. These technologies were able to connect users with the one data source that contains the information they need. Search Engines were then developed in order to find this one data source in the Internet using keywords. Although they were arguably an essential factor in the success of the Web, currently they are not sufficient to respond to all users queries.

[1] World Wide Web Consortium

[2] HyperText Transfer Protocol

[3] HyperText Markup Language

[4] Uniform Resource Identifier

The Web today is the world's largest data store. It is known as a set of interlinking documents of uncontrolled content and heterogeneous syntax and semantics. A space where everyone can contribute, at any time, which inherently leads to different types of heterogeneity. For instance, the content of documents in the Web is expressed in different languages and uses different units, terminologies, etc.

The Web, however, is more than a set of static documents and Web pages, described as the Surface Web [Alba *et al.*, 2008]. The heterogeneity confronting current information systems and Web application is far from being just syntactical. Indeed, there is another Web called the Deep Web which is many times larger than the Surface Web [Szeredi *et al.*, 2014]. It is also known as the Invisible Web since it cannot be crawled and indexed by any of the current search engines, which are external programs but integral to the use of the Internet. The latter need to send tailored parameters via a Web form submission in order to access the Deep Web, which is something beyond their present capabilities.

Web APIs[5] are an important part of the Deep Web and are an access method to local data in the Web. They allow third party access to functions and data, which generate new knowledge resources and open up new opportunities for applications to utilise, combine and re-propose information. Along with offering the possibility of mashing up content from different Web data sources [Bizer *et al.*, 2009], most of the results are expressed in a semi-structured format that can be automatically exchanged and computed. This led to a growth in their volume and in the availability of semi-structured data on the Web.

In 2006, a new Web started to emerge: the transformation from a global information space of linked documents to a Web of Linked Data. The concept of Linked Data conforms with long standing aims of the Semantic Web community, which is to associate meaning and semantics to data that is both machine and human readable, as pointed out by Berners-Lee and Fischetti [1999, p 177]:

> ”The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web - a Web of data that can be processed directly or indirectly by machines.”

[5] Application Program Interface

Linked Data is a materialisation of this idea. It is a paradigm that lowers the barriers and facilitates the publishing of interlinked structured and machine-readable data based on a set of recommendations and principales [Berners-Lee, 2006b]. Linked Data has completely changed the procedure of sharing knowledge over the World Wide Web [Bizer *et al.*, 2009]. The reduction of restrictions in publishing Linked Data led to a dramatic growth in the Web of Data and an extension to many areas and domains [Bizer *et al.*, 2009].

Taking the semantics of data on the Internet into consideration is another limitation of search engines that is beyond their current capabilities. Their search method is mainly textual [Khodaei and Shahabi, 2012; Mukherjea *et al.*, 1997], based on matching the terms of the request (query) with the terms of the indexed documents. This (full-text) method is compensated for by sophisticated results ranking stage(s) [Szeredi *et al.*, 2014].

The result of the growth of the Web, over time, with different development stages, is that there are large islands of information stored in distributed, independent and autonomous data sources. The latter are heterogeneous in terms of their structure, syntax, semantics, the access method, language, or protocol, etc. These data sources are described in different data models. In many cases, these data sources can be used complementarily to exploit their full potential and to respond to the user or systems' needs.

Currently, it is required, in many Web search scenarios, to manually gather and connect the results in order to find the information the user needs, which can be difficult or even infeasible given the scale of data today. This also involves access difficulties to the user, such as having to learn a new query language (for example SPARQL[6] for Linked Data), and for systems, to rewrite the query or to adapt the request according to each data source. Other challenges for systems include changes to these data sources and their content, which will affect the mechanism they designed in the first place to adapt or to rewrite the query or the request.

The content of previous data paradigms and spaces that has not been migrated to a newer data space is not necessarily of no value. For instance, there is a large amount semi-structured data, provided by the Web APIs, which are not part of the Linked Data cloud but still have a high value. Converting from XML (a semi-structured format) to RDF (the data model of Linked

[6] (S) Protocol and RDF Query Language

Data) is not sufficient as this only deals with the syntactic rather than the semantic aspects of the data. It is rather one of many steps. Several barriers exist and specific characteristics to be considered for semi-structured data to be published as Linked Data (see Section 3.7).

This thesis extends the available solutions to address the challenges stated in the two previous paragraphs, which can be summed up as:

- The difficulty of retrieving and connecting information from different sources for users and to sustain the change in the data sources for systems.
- The isolation of previous data spaces from the newer ones that starts to take place.

These challenges are addressed with the focus on semi-structured and Linked Data. The next section explains the choices of inputs and tasks for this thesis, and highlights the significance and impact of the outcome.

1.2 Motivation

The motivation of this thesis is presented in question/answer form, explaining the choices of the subjects and input data models. This section emphasises the gap in knowledge and general significance of the research and the work presented.

MQ1. Why this thesis is specifically focusing on semi-structured and Linked Data?

Documents are generally unreliable and uncontrolled sources of information. Technologies exploiting them are mainly textual (full-text method). Semi-structured and Linked Data, however, can produce data-centric results allowing more flexible search criteria [Dayananda and Shettar, 2011; Popov, 2013].

The latest information available about these two data models shows that that their providers are growing in numbers [Abawajy, 2015; Ahammad *et al.*, 2016]. Figure 1.1 shows the near exponential increase of the number of Web APIs, which are access tools that use semi-structured data in exchanging information and considered one of their primary sources. This is validated

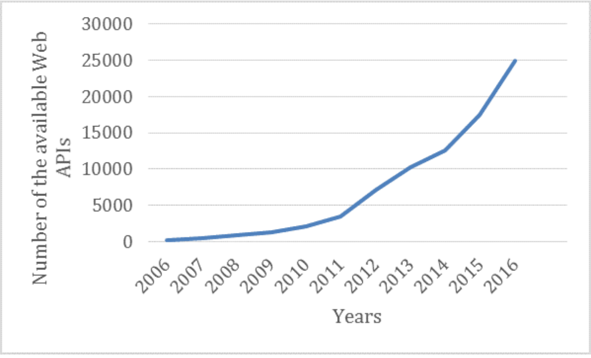


Figure 1.1: The increasing growth of Web API. [ProgrammableWeb, 2013] (until 2013) and [Burton, 2012] (projected from 2013 to 2016).

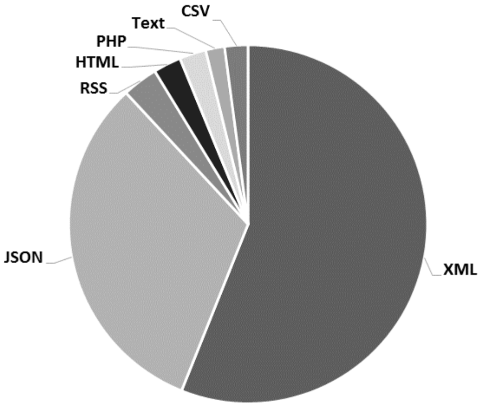


Figure 1.2: The format of the output of Web APIs [ProgrammableWeb, 2013].

in Figure 1.2 that indicates the formats utilised to represent the outputs of Web APIs. There are many other indications that confirms the rapid growth of semi-structured data being their use in remote sensors, social media, smart phones and archives [Ahammad *et al.*, 2016].

Likewise, both the number of datasets and the size of the information published as Linked Data, continue to rise. Table 1.1 illustrates two important aspects. First, the wide range of domains where information is published as Linked Data. Second, the average increase of 80% in the number of contributors into the Linked Data cloud. Because of the increase in the number of datasets is not a clear indication that the overall size of Linked Data follows the same pattern, Table 1.2 shows the increase in terms of triples in DBpedia[7], which is the largest reservoir of Linked Data in the world [Alam *et al.*, 2015]. It also shows that Linked Data is expanding to cover more domains as shown by a growth in the number of entities and ontology classes.

Additionally, the providers of these two data models, the semi-structured and Linked Data, are both part of the so-called the deep Web, its size is considerably greater than the conventional Web pages but still unreachable by current search engines.

To sum up, semi-structured and Linked Data are two data models that are growing and promising in terms of capabilities and size; yet, they are still unexploited by the current [Szeredi *et al.*, 2014] (all but experimental) search engines. They can be easily parsed, comparing documents, even though they are not restricted by any predetermined or fixed vocabulary defin-

[7] http://dbpedia.org/

Category	Datasets 2011	Datasets 2014	Growth	Example
Life Sciences	41	83	102%	taxonconcept.org
Social Networking	-	520	-	quitter.se
Publications	87	96	10%	bibsonomy.org
Governments	49	183	273%	data.gov.uk
Cross-domain	41	41	-	dbpedia.org
User-generated Content	20	48	140%	data.semanticweb.org
Geographic	31	21	-32%	geonames.org
Media	25	22	-12%	linkedmdb.org
			80%	

Table 1.1: Linked Data Datasets Increase between 2011 and 2014 [Schmachtenberg *et al.*, 2014].

Release version	Triples (billions)	Entities (millions)	Ontology (classes)	Triples Growth Percentage
2016	8.8	6	754	27%
2015	6.9	5.9	736	130%
2014	3	4.58	685	22%
2013	2.46	4.26	529	+13% (3.77M in 3.8 release)

Table 1.2: DBpedia growth in the last 4 releases. [Alam *et al.*, 2015] [Freudenberg *et al.*, 2017]

ing their structure.

MQ2. Why integrating Semi-structured and Linked Data?

Data integration has been the focus of database research community, and then the Semantic Web community, for more the 35 years [Doan *et al.*, 2012]. The need for a unified view that reconciles the different types of heterogeneity between different data models and formats has not cease [Cambiaghi *et al.*, 2016]. The new data paradigms and models, such as Linked Data, that have different features and characteristics suggest the need to revisit this task.

Integrating semi-structured and Linked Data sources does not only mean collecting and combining them (i.e. data federation), but also providing a logically unified view. This allows a transparent and expressive access to these data sources that are unreachable by current text based search engines. It is an important service to provide, especially today, as the capabilities of general Web users does not seem to be growing at the same rate as technologies and query languages in the Web. It is not sufficient to provide theoretical or technical achievements without having usable access to these data sources within the reach of non-expert users. Providing a unified and homogeneous view allows users to search using every property that constitutes source datasets without the need to write complex queries.

Non-expert is used as a term that does not only imply individuals without SPARQL background, but it also includes organisations without expertise in this new technology and cannot afford to migrate a large amount of data.

MQ3. Why interlinking Semi-structured with Linked Data?

The main obstacle in publishing Linked Data is to connect the dataset being published externally with related data sources in the cloud [Taheri and Shamsfard, 2012]. This is a popular problem in Web Semantics known as data interlinking. It is, however, frequently addressed on existing data stored in benchmark files. In this work the inputs are user datasets against the Linked Data cloud (see Section 6.3). The approach aims at firstly verifying the existence of data being published to provide links with it. It is based on instance matching and similarity measurement algorithms that allow processing a large number of Linked Data datasets.

Linked Data is a (relatively) new paradigm that enables meaning that is both machine and human readable. This conforms with the main aim of Semantic Web. There are many other data sources that are still growing and providing semi-structured data on the Web. The overall objective is to provide an automatic tool that converts and publishes semi-structured data model and links it with Linked Data. Addressing this entire issue is beyond the scope of this thesis. The approach presented here extend the findings of other approaches in this area, and focuses on providing external identity links between the output of the other approaches, such as unpublished and unlinked RDF[8] datasets, with dataset in Linked Data space. This enriches the information available about the resource described in the Linked Data space, providing semantics to semi-structured data.

MQ4. What do this research outcomes signify to ordinary users?

The direct service that this research can offer to the user is clearly demonstrated in its prototypes (see Sections 7.3 and 7.4). Having a transparent and data centric access to multiple dispersed sources will expand the content queried, thus increase the number of results, espe-

[8] Resource Description Framework

cially if the sources follow different data models and paradigms from what was initiated in different years. Furthermore, the prototypes and implementations the proposed data integration architecture, which are a keyword search systems, allow the users to break free from the learning curve of new query languages, such as SPARQL, or the structure of the HTTP request in Web APIs. They also offer other mechanisms to enrich the keyword with other information to achieve a better expressivity. The final result is the automation of many tasks and functions that the user would normally have to complete such as adapting to different structures, distributing the request or the query, and combining and presents the results.

Data interlinking has clear and direct benefits to the ordinary user, which facilitates following Linked Data principles in publishing data. This improves the usability and accessibility of the data published and increases its chances for reusability [Hietanen *et al.*, 2016].

There are various research areas where integrating semi-structured and Linked Data can contribute indirectly, including life science, enterprise information systems stream data and Smart Cities [Abelló *et al.*, 2013; Kettouch *et al.*, 2017b; Kienast and Baumgartner, 2011; Le-Phuoc *et al.*, 2012].

MQ5. What are the general barriers and challenges related to integrating and interlinking semi-structured and Linked Data?

This answer covers the generic difficulties that can arise in integrating and interlinking semi-structured and Linked Data. For more detailed and specific challenges, please see Section 3.5 and 6.3. Three key barriers are:

- The structure and content of Linked Data is expected to rapidly change; therefore, many tasks have to be done on-the-fly which imposes time and computations limitations;
- Semi-structured data are not defined by any ontology and do not follow any predetermined structure;
- Many types of heterogeneity might be present including, the access protocol, the structure, the query language, etc.

1.3 Research Questions and Objectives

This thesis seeks to solve these challenges by responding to these research questions:

RQ1. Can (converted) semi-structured datasets be interlinked with the Web of Linked Data automatically via a process that shows good quality measures?

Answering this question contributes to the long-term solution of bridging between semi-structured and Linked Data. It facilitates complying with Linked Data principles by providing an identity link between the datasets being published and their counterparts in the Web of Linked Data. The focus of this thesis is not the conversion from a semi-structured data format or model to the Linked Data's model, which can be straightforward depending on the selected method (see Section 2.5.4), but it is rather on using the extracted characteristics of the output of this conversion as requirements in designing the interlinking approach. Addressing this question will lead to achieving the following objectives:

- To highlight the obstacles in publishing into Linked Data generally, and to publish semi-structured data particularly.
- To critically investigate how related interlinking systems address the problem and highlight the difference between their scope and this thesis's focus.
- To design and implement an approach to link externally (converted) semi-structured to its counterpart in Linked Data cloud.
- To test and evaluate the designed data interlinking approach against similar approaches.

RQ2. Is it feasible to search and access semi-structured and Linked Data through a transparent, usable and expressive interface?

This question represents the short-term solution to bridge between semi-structured and Linked Data. Addressing this question involves reconciling the different types of heterogeneity between the sources belonging to the two data models. This thesis aims at offering data-centric access that is both usable and expressive. This allows users to access information without having to learn a technical query language.

RQ3. Can an integration system sustain the dynamism of the continuous expansion and changes in the Web of Data?

The problem of giving an integration system the ability to accommodate future changes is not new. It is a common problem for legacy systems, but not at the current scale. The frequency that Linked Data changes is relatively very high [An *et al.*, 2013; Svoboda and Mlỳnková, 2011] due to the lowering of barriers in publishing it on the Web and the possibility to create new vocabularies and ontologies to define it, which themselves can expend or change. Moreover, many Linked Data sources extract their content from a publicly-editable and living dataset. Therefore results of querying these sources will change over time. For example, DBpedia, which is the largest and most popularly used dataset in the Web of Linked Data extracts its content from Wikipedia[9] via an evolving code-base.

The research questions RQ2 and RQ3 will guide to attain the following research result objectives:

- Identify the challenges in accessing and integrating semi-structured and Linked Data.
- Explore and examine the scope, limitations and strengths of some of the popular integration systems and other related research areas dealing with similar problems according to the identified challenges.
- Conceptually design a highly automated and adaptable data integration system that takes as an input semi-structured and Linked Data, and that has the ability to:
  - Reconcile the different types of heterogeneity between the two inputs; and
  - Accommodate and sustain the continuous changes and expansion of Linked Data sources.
- Implement the data integration approach into a tool with a usable and expressive interface.
- Test and evaluate and the implementation of the data integration approach.

[9] https://www.wikipedia.org/

1.4 Original Contributions to Knowledge

The thesis makes practical as well as theoretical contributions in bridging between semi-structured and Linked Data. The main contributions are:

- Designing a novel schema matching approach for semi-structured and Linked Data that has the ability to automatically accommodate the continuous changes of Linked Data sources (see Chapter 5).
- Designing a new data integration approach for semi-structured and Linked Data with a high degree of automation (see Chapter 6).
- Designing a new data interlinking approach that takes only a source dataset as input and provides identity external links with many Linked Data cloud's sources (see Chapter 7).
- Using the domain and UMBC semantic similarity tool to allocate variable weights in measuring the similarity of the instances, according to the significance of their properties in defining the identity of the dataset (see Chapter 7).

1.5 Research Methodology

The research presented in this thesis is firmly grounded on current requirements and needs. The results not only have a theoretical value, but also the nature of the problem addressed and the contributed approaches are highly applied as well. The research methodology can therefore properly be described as constructive research, as presented by Kasanen *et al.* [1993]; Lukka [2003]. Crnkovic [2010] revealed that the constructive research method is very common in computer science, even though it is not very frequent to find it in their methodological discussion. Crnkovic [2010, p. 4] defined constructive research as:

> ”Constructive research method implies building of an artifact (practical, theoretical or both) that solves a domain specific problem in order to create knowledge about how the problem can be solved (or understood, explained or modeled) in

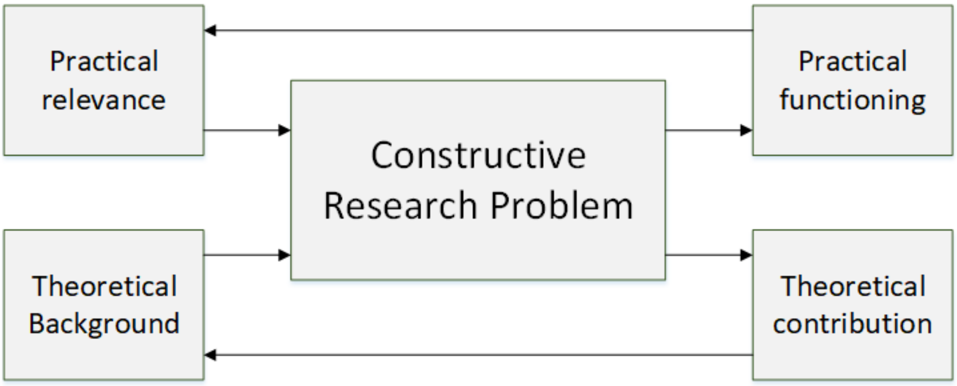


Figure 1.3: The research methodology used in this thesis [Author, 2017].

> principle. Constructive research gives results which can have both practical and theoretical relevance. The research should solve several related knowledge problems, concerning feasibility, improvement and novelty. The emphasis should be on the theoretical relevance of the construct. What are the elements of the solution central to the benefits? How could they be presented in the most condensed form?"

As Figure 1.3 illustrates, for a problem to pass as a constructive research, it ought to be linked with an accumulated theoretical knowledge and be demonstrated through novel working solutions. The research presented in this thesis is explicitly directed at building theoretical models that can be implemented and adapted to various context and use cases. The approaches proposed in Chapter 5, 6, and 7 are the main novelty and achievement of the research, and are of both practical and theoretical value.

Kasanen *et al.* [1993] presents the constructive research method as six phases that can vary in terms of order:

1. **Obtaining a comprehensive and theoretical background of the topic.** In this thesis, the background is formed by current knowledge about the data input, such as semi-structured and Linked Data, and the topics researched, such as schema matching, data interlinking and integration.

2. **Constructing innovative solutions.** This is the main content of Chapter 5, 6, and 7 of this thesis.

3. **Showing the theoretical contributions of the research.** In this thesis, the theoretical contributions are stated in this chapter, validated in the innovative solutions and elaborated and expanded in the conclusion.

4. **Finding a practical relevance of the research.** The problems addressed in this thesis are by their nature practical. The practical relevance is demonstrated and made clearer through the implementation of the solutions proposed.

5. **Demonstration that the solution works.** All the implementations of the solutions presented in this thesis were evaluated. Several technologies and libraries were used in the implementation of the solutions proposed, including: Java, Jena library, Rest APIs, HDT, RDF, SPARQL, UMBC EBIQUITY-CORE. All the the solutions were implemented in the same setting environment that can be found in Appendix III.

6. **Examining a scope of applicability of the solution.** The scope of the applicability of the solution is indicated in Chapter 8 where all solutions are combined in one applied framework.

1.6 Approach

The aim of this section is to demonstrate how this research is conducted and look at the general method that guided the work in the thesis to design and address the research questions. As Figure 1.4 shows, the approach followed to conduct this research consists essentially of three phases:

1. **Background Examination:** this stage defines the input data and their technologies in order to extracts their characteristics and the challenges associated with working with them. It investigates the techniques and methods for publishing Linked Data content on the Web as well as highlighting the importance of semi-structured data.

2. **Exploratory Research:** in this stage, the operations researched in this work, their components and evaluation method are studied. This phase also explores the general categories of the systems performing these tasks. It benefits from the previous stage by taking into

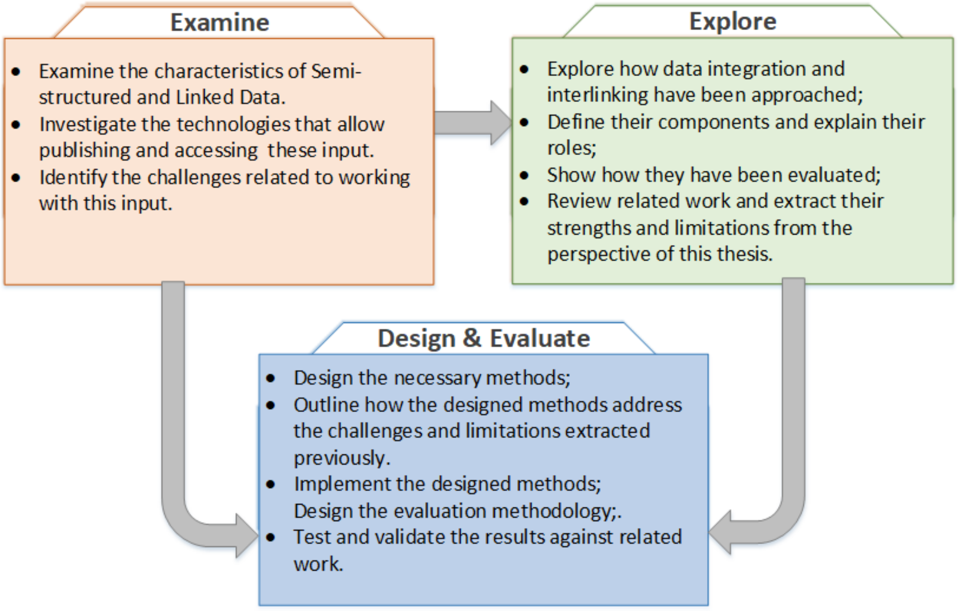

Examine
Examine the characteristics of Semi-structured and Linked Data.
Investigate the technologies that allow publishing and accessing these input.
Identify the challenges related to working with this input.
Explore
Explore how data integration and interlinking have been approached;
Define their components and explain their roles;
Show how they have been evaluated;
Review related work and extract their strengths and limitations from the perspective of this thesis.
Design & Evaluate
Design the necessary methods;
Outline how the designed methods address the challenges and limitations extracted previously.
Implement the designed methods;
Design the evaluation methodology;.
Test and validate the results against related work.

Figure 1.4: Research Approach [Author, 2017].

consideration the characteristics and challenges of the input data and concentrating on particular aspects. The related systems are reviewed in this stage with more focus on those that are more specific and close to the proposed solutions.

3. **Designing, Implementation and Evaluating Solutions:** The recommendations resulted in the background examination and the gaps discovered from the exploratory research are the input for this phase. Both are used to identify the functional requirements for designing and developing the new approaches. Three approaches are proposed as part of this work that are complementary. Hence, the design phase is combined with the implementation and the evaluation in order to give a clear idea about every solution before using it as a component in another approach.

1.7 Overview of Thesis Structure

Figure 1.5 illustrates the structure of the thesis with reference to the research approach followed in this thesis. The remainder of the thesis has been arranged as follow:

Chapter 2 - Linked Data: Technologies, Implications and Challenges: Drawing extensively from the literature, this chapter begins with an introduction to the main concepts and Se-

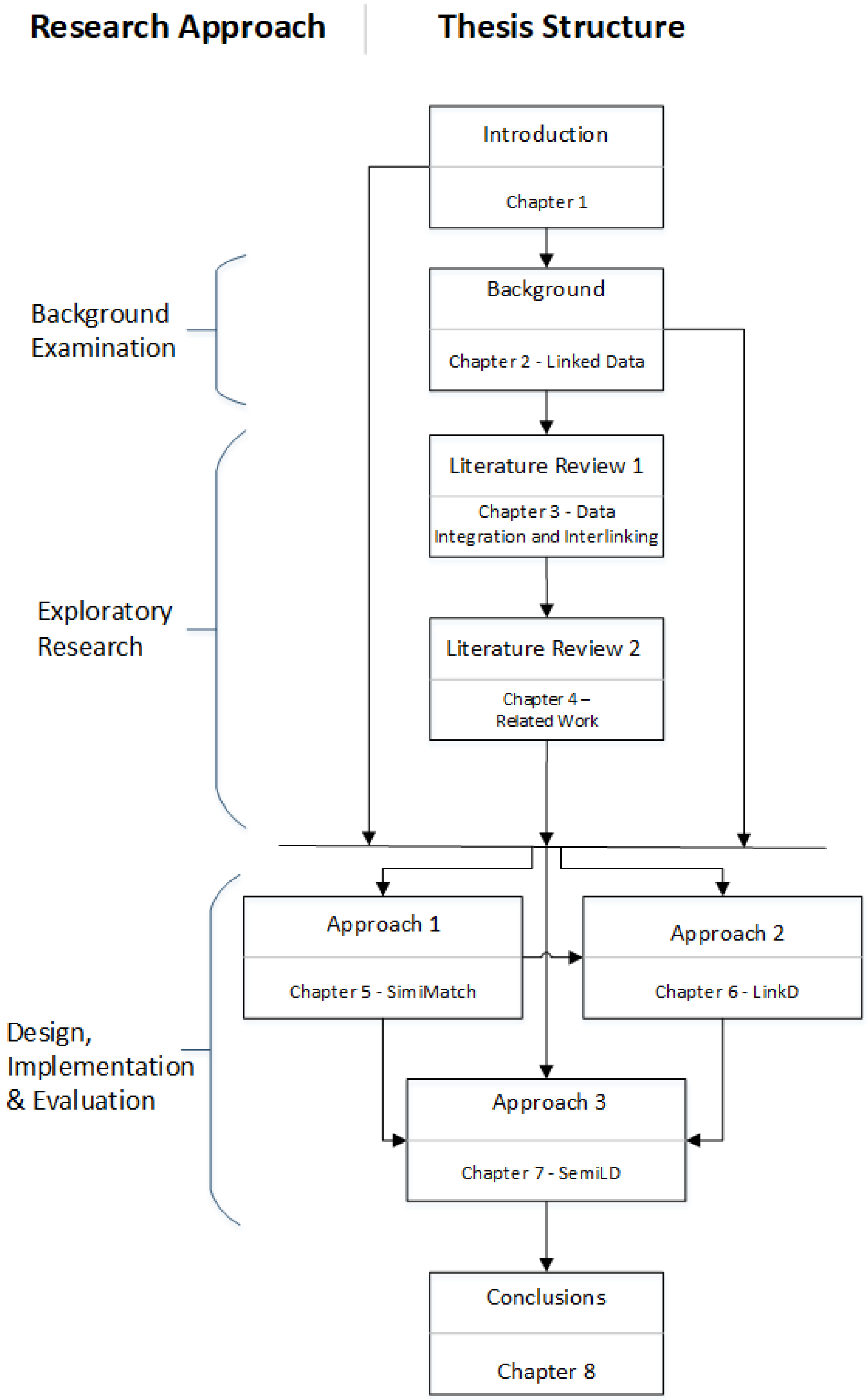


Figure 1.5: Thesis structure : Research approach [Author, 2017].

mantic Web technologies that support the Linked Data paradigm. It covers all the preliminaries to be used in upcoming chapters i.e. RDF, ontologies, SPARQL, URIs, XML[10] and JSON[11]. The chapter also explores several challenges that arise in consuming or publishing data in a Linked Data setting. Finally, it presents semi-structured data from a Linked Data perspective.

Chapter 3 - Data integration and interlinking: Having the introduced the inputs, their paradigms and challenges in Chapter 2, this chapter introduces the operation that this thesis focus on being data integration and interlinking as well as schema matching. It gives an overview of the different techniques, approaches and components they use and their evaluation methods.

Chapter 4 - Related Works: Following the introductory chapters, this gives an analysis of and gathers the related work of the proposed approaches in Chapters 5, 6 and 7, and highlights the gaps that need to be filled. It also identifies the sources of some of the features that the author incorporated when designing the contributed solutions.

Chapter 5 - SimiMatch: Schema Matching for Semi-structured and Linked Data: This chapter introduces SimiMatch, namely Schema Matching for Semi-structured and Linked Data, a common and an important component of the proposed data integration and interlinking approaches. The chapter describes the individual modules in the approach and explains the stages it goes through to reconcile the structural heterogeneity between semi-structured and Linked Data.

Chapter 6 - LinkD: Element-based Data Interlinking of RDF datasets in Linked Data: This chapter tackles the problem of data interlinking of semi-structured data with Linked Data space. It proposes LinkD and its weight allocation module along with showing how SimiMatch is adapted for this context. Finally, a comprehensive evaluation is presented at the end.

Chapter 7- SemiLD: Keyword Search over Semi-Structured and Linked Data: The focus of this chapter is on the data integration task. It introduces SemiLD and its two prototypes, which are keywords access systems that retrieve their data from semi-structured (Web APIs) and Linked Data (SPARQL endpoint) sources. The chapter goes through the components of the modular architecture proposed. An evaluation is carried out at the end.

[10] Extensible Markup Language

[11] JavaScript Object Notation

Chapter 8 - Conclusions and Future Work: This highlights the contributions and limitation of the work presented in this thesis and articulates a vision for the future.

Chapter 2

Linked Data: Technologies, Implications and Challenges

I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important... Good programmers worry about data structures and their relationships

Linus Torvalds

2.1 Introduction

Before proceeding to identifying and to highlighting the challenges that a category of systems exploiting semi-structured and Linked Data are confronted by, it is helpful to study the data themselves, their origin, different representations, and the available ways of publishing and consuming them. Hence, this background chapter starts with introducing the vision of Web Semantics to pave the way for understanding the composition and the origin of Linked Data. The technicalities associated with publishing into the Web of Linked Data are defined in Section 2.3. Then, in Section 2.4, the different ways of exploiting Linked Data are explored with more attention given to the relevant and popular methods. This chapter then changes its focus, in

Section 2.5, to semi-structured data, viewed from Linked Data point of view, their main source and formats. Finally, in Section 2.6, the challenges of Linked Data are stated and explained.

2.2 The Origin: Web Semantics

Before the Semantic Web vision started to be implemented, it was entirely up to humans to interpret the static Web content and derive conclusions. The Web in that ”read-only” era [Berners-Lee *et al.*, 2001] can be described as a display case where there was primarily only one way of exchanging information, which is from the publishers to consumers [Singh *et al.*, 2011]. More technically, it was more like a Web of hyperlinked documents, also known as Web 1.0, in which every page of these documents is identified using a unique global address, a URL[12], and remotely accessed through HTTP links using a Web browser.

The development of new technologies and standards, including XML and Web APIs, revolutionised the Web. They opened up new opportunities for users to interact or even to add or modify content to the Web (through wikis for example). But more importantly, they allowed interoperability, mashup applications and other data exchange capabilities. Broadly speaking, this marked the beginning of the boom of Semantic Web research activities and for their effects on the Web to become apparent, in the late 1990s [Songtao and Junliang, 2005].

In 2006, Shadbolt *et al.* [2006, p. 66] highlighted that Tim Berners-Lee:

> ”described the evolution of a Web that consisted largely of documents for humans to read to one that included data and information for computers to manipulate... This simple idea, however, remains largely unrealized. A Web of data and information would look very different from the Web we experience today.”

The idea of a uniform data structure and paradigm in order to achieve this transformation from a Web of documents to a Web of Data was introduced in the same year, and was called Linked Data. As the title of this section suggests, Linked Data principles are built upon Semantic Web technologies and standards.

[12] Uniform Resource Locator

The Semantic Web aims at allowing a better sharing and re-use of data on the Web ”by giving information a well-defined meaning, better enabling computers and people to work in cooperation” [Berners-Lee *et al.*, 2001, p. 4]. This meaning needs to be machine readable in order to minimise human interference [Wang *et al.*, 2006] and to enhance the interoperability of Web content. The Semantic Web also advocates for a Web where intelligent programs or agents can operate over distributed data sources to either automate certain tasks or work together with end-users to accomplish tasks on demand [Popov, 2013].

The Semantic Web is traditionally represented by its framework: ”Semantic Web Layer Cake” [Berners-Lee, 2006a] shown in Figure 2.1. The framework consists of a stack of technologies and standards that enable the creation of data stores on the Web, building vocabularies, write rules for handling data, and developing applications and services to exploit the output. These technologies will be explained (in the next section) in the context of Linked Data publishing.

2.2.1 Ontologies

Before introducing the core concepts of Semantic Web, regarded from a Linked Data point of view, it is first necessary to explain the concept of ontology. Ontologies are the pillars and the key backbone of the Semantic Web as well as of Linked Data. An ontology is defined as a formal specification of a shared conceptualisation of some domain knowledge [Gruber, 1993]. They represent a shared and common understanding of a domain that can be communicated across people and application systems [Fensel, 2003]. They are flexible, extensible, and scalable mechanisms to describe and structure a stored information. The information is encoded in ontologies in the form of concepts and properties linked via semantic relations.

Ontologies can be seen, as any model in automated reasoning [Krachina and Raskin, 2006], as an attempt to simulate human thinking [Kroeze, 2010] and his/her representation of real world things and their properties. For example, if the question ”What is the last document you have read?” is asked to a human being, his/her thinking process, will not normally consider cars or houses. Only objects of the type document will be taken into account e.g. books, thesis, newspaper etc. This task is very easy for a human brain, but challenging for machines.

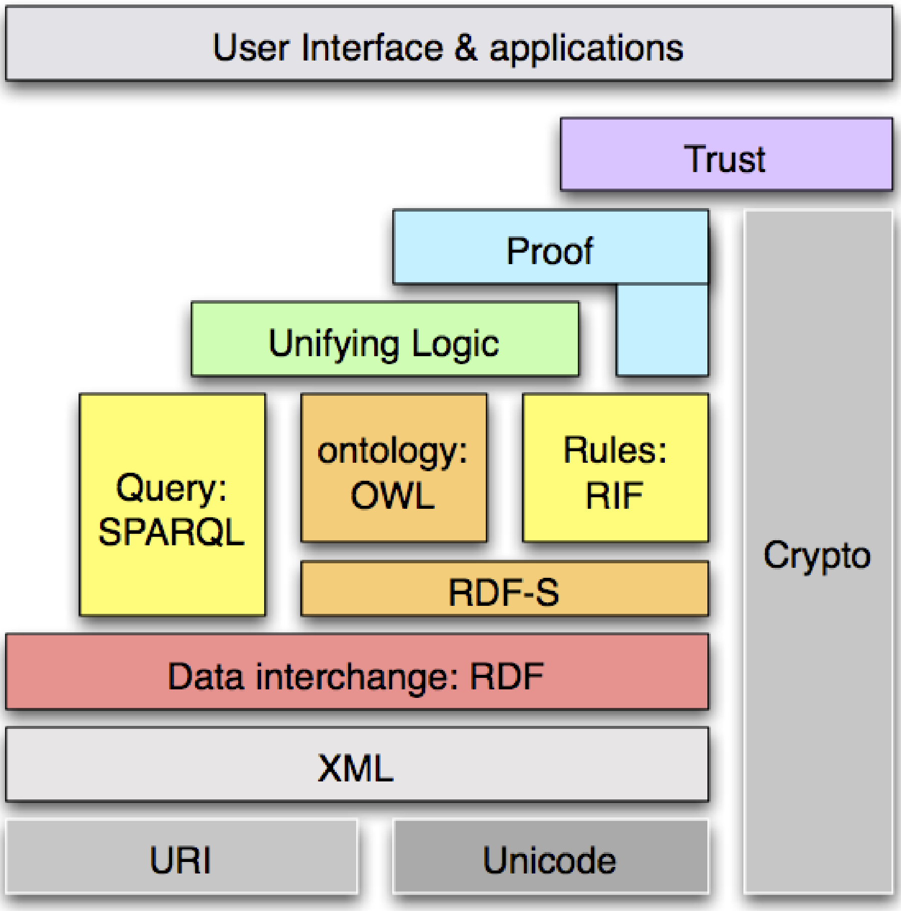


Figure 2.1: Semantic Web general framework (layer cake) [Berners-Lee, 2006a]

Ontologies simulate such complex processes by categorising real world objects according to a common vision and understanding, and allowing reasoning and inference upon the derived class (i.e. category or concept) instances and their relationships [Franco-Bedoya, 2015].

There have been many methods and languages proposed in order to represent ontologies. One of the earliest formalisms was Frame Language [Minsky, 1975] which allows descriptions of subsets and hierarchy of the ontology classes via frames. Then Description Logics offered new possibilities in modelling the relationships between concepts, roles and individuals [Nardi *et al.*, 2003]. One example of description logics is the popular OWL[13], which is an XML-based knowledge representation language for authoring ontologies. It is a W3C recommendation [W3C, 2004] that aims at facilitating machine interpretability of Web content.

2.2.2 RDF Data Model

A data model is an abstraction used to represent real world entities, the relationship between these entities and the operations that can be performed on the data [Ullman, 1990]. Various data models can be distinguished on the Web, including: hierarchical, relational, graph etc.

RDF structure information as a set of statements where each statement comprises a subject, a predicate (property), and an object [Brickley and Guha, 2004]. The subject-predicate-object relationship is called a triple. The subject represents what is being described by the RDF triple and it can be either a URI or black node. The resource represented by a blank node is called an anonymous resource. The predicate is the relationship between the subject and the object, and it ought to be a URI. The object can be a URI, blank node, or a literal and it retains the value of the subject in relation to the predicate. Formally, an RDF triple can be defined as follows:

Definition 2.1 (RDF triple). *An RDF triple is represented as a tuple* $\{S,P,O\} \in (I \cup B \cup L) \times I \times (I \cup B \cup L)$*, where S is called the subject, P the predicate, and O the object and I, B and L, are used to represent IRIs*[a]*, blank nodes and literals respectively.*

[a] Internationalised Resource Identifier

Example 2.1 puts the definition and the previous explanation in an RDF/XML document style:

[13] Web Ontology Language

Example 2.1: RDF/XML example of a triple

```
<rdf:Description rdf:about="subject">
       <predicate rdf:resource="object" />
       <predicate> literal value </predicate>
<rdf:Description>
```

Example 2.2 is an RDF file describing the Cambridgeshire county:

Example 2.2: An RDF description of Cambridgeshire County

```
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <rdf:RDF xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc= "
      http://purl.org/dc/elements/1.1/" xmlns:region= "http://Example.com/ ">
 3        <rdf:Description rdf:about=" http://Example.com/Cambridgeshire ">
 4               <dc:title> Cambridgeshire </dc:title>
 5               <dc:publisher> Mohamed </dc:publisher>
 6               <region:population> 806,700 </region:population>
 7               <region:principaltown
 8                      rdf:resource="http://example.com/Cambridge"/>
 9        </rdf:Description>
10 </rdf:RDF>
```

RDF uses CURIE (or Compact URI) abbreviation style to define reusable prefixes that can be used to write URIs in a shorter form. Three prefixes were declared in the previous example (line 2), one of them is: `xmlns:dc= ”http://purl.org/dc/elements/1.1/”`. The prefix is then used in two predicates: `dc:title` and `dc:published`. The syntax of prefix declaration varies depending on the serialisation.

RDF data model also allows specification of the language or the data type of the literal values, as the following examples show: `''Pyramid''\@en` (`''Pyramide''\@fr`) and `''2016-11-20''^^xsd:date`.

Similarly to XML, RDF is not constrained by any mechanisms for declaring the properties names [Decker *et al.*, 2000]. It is the role of the RDF descriptive vocabulary that is called the RDF Schema (RDFS[14]) [Rula and Palmonari, 2013]. RDFS, positioned just above RDF in the Semantic Web Layer Cake (see Figure 2.1), consists of a set of classes and properties to describe RDF resources [Brickley and Guha, 2004]. It distinguishes resources into classes and predicates in an ontological structure, where their data types and the relationships are defined.

[14] Resource Description Framework Schema

Despite the fact that RDF vocabulary itself possesses predicates such as `rdf:type` to define the domain, it still needs additional schema properties to compensate its lack of (or to enrich its) expressivity, such as [Christodoulou, 2015]:

- rdfs:Class and rdf:Property to define new class and property instances;
- rdfs:subClassOf and rdfs:subPropertyOf to define class and property hierarchies respectively;
- rdfs:domain and rdfs:range to associate a class to the subject and object of a property.

Example 2.3 shows a definition of a property `hasMother` using RDFS:

Example 2.3: Definition of a property hasMother using RDFS

```
<http://www.example.com/humans.rdfs#hasMother> a rdf:Property;
   rdfs:label "has for mother"en,"a pour mere"fr ;
   rdfs:comment "to have for parent a female."en,"avoir pour parent une
      femelle."fr ;
   rdfs:range <http://www.example.com/humans.rdfs#Female>;
   rdfs:subPropertyOf
                <http://www.example.com/humans.rdfs#hasParent>.
```

2.2.3 SPARQL

SPARQL is a semantic query language to retrieve and update RDF data. It utilises triple pattern matching in order to retrieve the results of the queries. The results can be result sets or RDF graphs. In 2008, SPARQL 1.0 [W3C SPARQL Working Group, 2008] became an official W3C recommendation, so did its successor, SPARQL 1.1 [W3C SPARQL Working Group, 2013], five years later. SPARQL 1.1 extended the previous version by providing more support for complex and federated queries.

SPARQL endpoints[15] can also run queries remotely using Jena[16] framework, for instance. Jena framework is an open source and flexible set of Java libraries ”implementing basic functionalities for semantic data storage and querying, following the W3C standards” [Efthymiou

[15] An endpoint is one end of a communication channel

[16] http://jena.apache.org/

et al., 2015, p. 4]. It also provides a SPARQL engine that allows running queries on local RDF files and also to send requests to SPARQL endpoints and retrieve the results.

The SPARQL query described in the rest of this section is a generic SELECT query that extracts raw values from a SPARQL endpoint. Other forms of queries, that are not covered or used in this thesis, exist to retrieve results in other formats or to load and update RDF store using SPARQL. Five main parts, in which two are required and three optional, can be distinguished in a SELECT SPARQL query:

- **Definition of prefixes (optional):** Although their use is optional, they are recommended for better readability of the rest of the query. Similarly to RDF (as described in Section 2.2.2) they are located at the very beginning to define abbreviation for namespaces that will be used in the triple pattern (subject, predicate or object), for example:

  ```
  PREFIX exp: http://example.com/resource/
  ```

  In the above example, the namespace is ”http://example.com/resource/” and `''exp''` is the prefix.

- **Dataset Clause (optional):** It indicates the URI of the RDF dataset to be used during pattern matching. If it is not declared, the query processor determines the dataset to use.

  ```
  FROM <uri> | FROM NAMED <uri>
  ```

- **Results clause (required):** This is the part where the requested variable or set of variables is/are indicated.

  ```
  SELECT [Aggregate_function] { * , var_1, var_2, ..., var_n}
  ```

  The users can also ask for all variables using * as the above example shows. Similarly to SQL, SPARQL also offers the possibility of adding aggregate functions to the query, as the example highlights. The common aggregate functions include `COUNT`, `SUM`, `MIN`, and `MAX`.

- **Query pattern (required):** The following represents the part that specifies the triple pattern that needs to be matched with the underlying dataset.

```
WHERE {
...
Triple Patterns to be matched
...
FILTER(... patterns to be filtered ...)
}
```

- **Query modifiers (optional):** These are applied to create a user desired solution sequence out of the retrieved results. The modifiers supported by SPARQL are `Distinct`, `Order By`, `Reduced`, `Limit` and `Offset`.

```
ORDER BY DESC(m) OR ASC(m)
LIMIT L
OFFSET F
```

2.2.4 URI

A Uniform Resource Identifier (URI) is a compact string of characters for identifying an abstract or physical resource [Berners-Lee *et al.*, 1998]. A URI can be further classified as a locator, a name, or both. Figure 2.2 illustrates the difference between the URL, URI and IRI. URLs identify names of resources on the Web via a representation of their primary access mechanism and network location. URIs are URLs that are also used to identify things that exist on the Web. The IRI extends upon the URI by using Unicode characters to enable its representation in different languages.

As stated in Section 2.2, information on the Web needs to be accessible by both actors: humans and machines. Making data and its links machine-readable does not indicate humans would be excluded from being able to request and receive information in a format that is understandable to them. Dereferenceable URI is the technology that serves that purpose. It is a resource retrieval mechanism that uses any of the Internet protocols (for example HTTP) to obtain a copy or representation of the resource it identifies [Yaghouti *et al.*, 2015].

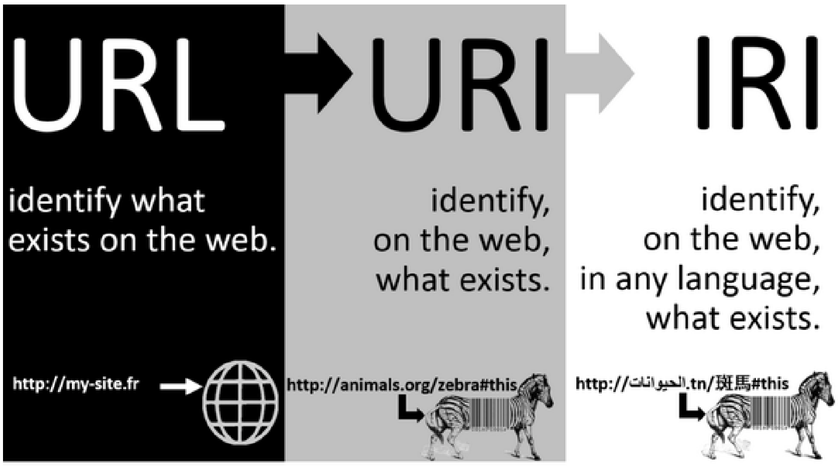

URL
URI
IRI
identify what exists on the web.
identify, on the web, what exists.
identify, on the web, in any language, what exists.
http://my-site.fr
http://animals.org/zebra#this
http://الحيوانات.tn/斑馬#this

Figure 2.2: Difference between URL, URI and IRI [Gandon, 2014]

2.3 Publishing Linked Data

This section explains how the technologies in the lower stack of the Semantic Web architecture are used together to publish Linked Data. First, the Linked Data initiative and its relation with the Semantic Web are explained. This section then goes on to introduce the Semantic Web technologies needed to publish in a Linked Data way.

2.3.1 The Linked Data Paradigm

Before going into the theoretical and technical details of Linked Data, it is essential to make the distinction between data and a document containing data. Data is a machine readable set of discrete and objective facts about events [Tuomi, 1999] or values of qualitative or quantitative variables [Roy and Zeng, 2015]. Document is a broad word, but can be regarded in this context as a human readable medium of data transmission, for example: mail messages, HTML pages, reports, etc.

As outlined in previous sections (see Section 1.1 and 2.1), Linked Data is a pragmatic approach for the transformation from a document-based Web to a Web of interlinked structured data. The idea was to create a Web where anything can be linked to anything. Linked Data aims to provide links between different data sources in order to create a single global data space, ”the Web of Data” [Hausenblas, 2011]. These links ought to be machine-readable and connect to related data whether from the same or from other external sources. This objective can be achieved by utilising RDF, URIs and HTTP to publish and interlink structured data on the Web.

Linked Data and Semantic Web are two terms that are often used interchangeably [Lehmann and Völker, 2014] to refer to an initiative that promotes interoperability and the degree of automation over distributed and heterogeneous sources while preserving their autonomy. Even though this is not completely inaccurate, Linked Data, strictly speaking, was introduced as a common way of publishing data that complement the general architecture and functioning of the Semantic Web.

More formally, Linked Data refers to a set of best practices for publishing and interlinking structured data on the Web, described by Tim Berners-Lee in his Web architecture note Linked Data [Berners-Lee, 2006b]:

P1. Using URIs as names for things.

P2. Using HTTP URIs so that people can look up those names.

P3. When someone looks up a URI, provide useful information.

P4. Include links to other URIs. So that they can discover more things.

Linked Open Data (LOD) is Linked Data released under open license [Méndez and Greenberg, 2012]. ”Open Data” indicates the right to link data and reuse it freely, without any copyright restrictions, something which is indispensable if data is to be linked [Mitchell, 2013]. LOD diagram in Figure 2.3 [Abele and McCrae, 2017] illustrates the wide use of Linked Data in different domains and areas e.g. Sciences, governmental and statistical data, geography, etc. It also highlights the major LOD contributors such as DBpedia, LinkedMDB[17] and Geonames[18].

2.3.2 Identifying Resources using URIs

URIs are one of the technologies that allowed the shift from the Web of documents to the Web of Linked Data. Their usage for identifying things and to allow discoverability is the first and

[17] http://www.linkedmdb.org/

[18] http://www.geonames.org/

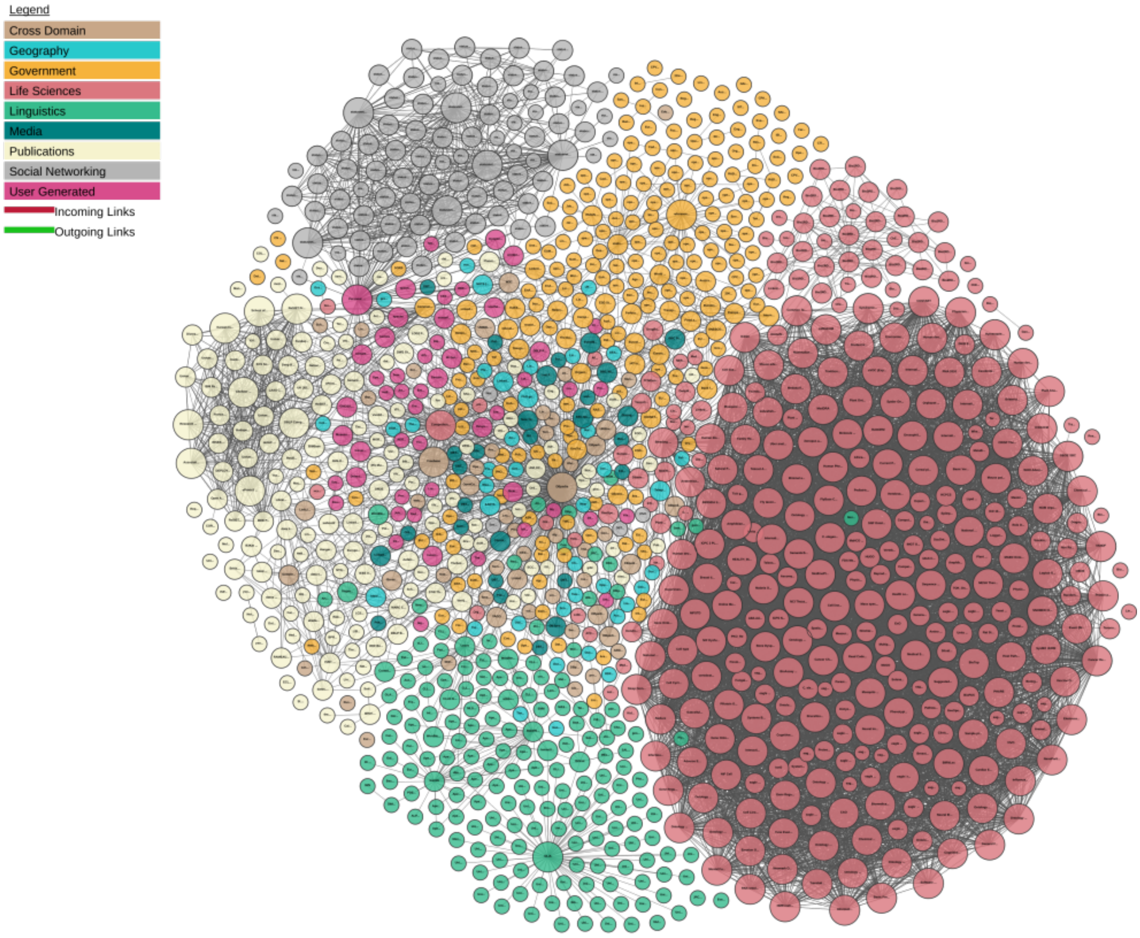


Figure 2.3: The Linked Open Data cloud shows the wide use of the paradigm in many domains and areas [Abele and McCrae, 2017]

the last principle and best practice **(P1)** and **(P4)** of the Linked Data paradigm respectively. ”Things” is the key word in the previous sentence. The traditional utilisation of URIs is to identify Web documents [Berners-Lee *et al.*, 1998]. In Linked Data, their function is extended to serve as a unique ”ID” for resources and real world objects on the Web. Having more than one resource with the same URI leads to ambiguity and assigning more than one URI to the same resource leads to redundancy, which decrease significantly the chances of reusability.

Using dereferenceable URIs is important for the second Linked Data principle **(P2)**, ”in fact, without them, it is not possible to check what is attached to the URI” [Albertoni *et al.*, 2014, p. 8]. Therefore, they are needed for accessibility to the description of resources; and hence, for the reusability of URIs. Content negotiation mechanism [Fielding *et al.*, 1999] is the method to disambiguate the format of the Web document retrieved e.g. XML or RDF for consumption and HTML for humans [Christodoulou, 2015].

2.3.3 Describing Linked Data using RDF

RDF can be associated with the third principle **(P3)** of the Linked Data paradigm. Note that the third principle is also about looking up and accessing information, something that will be discussed in Section 2.4.3.

Linked Data uses RDF data model, which is the standard recommended by W3C. RDF is the uniform model utilised in Linked Data to structure and represent information about resources. RDF in Linked Data can be seen as a directed and labelled graph, composed of a finite set of RDF statements or triples. The nodes of the graph represent the resources (classes) or the subjects depending on its position in the RDF statement, and the typed arcs play the role of predicates (or properties). Various syntax formats can be utilised to represent the RDF data model, such as: RDF/XML[19], Turtle[20], N-Quads[21], N-Triples[22] , JSON-LD[23] and Notation3[24].

[19] https://www.w3.org/TR/REC-rdf-syntax/

[20] https://www.w3.org/TR/turtle/

[21] https://www.w3.org/TR/n-quads/

[22] https://www.w3.org/TR/n-triples/

[23] https://www.w3.org/TR/json-ld/

[24] https://www.w3.org/TeamSubmission/n3/

2.4 Consuming Linked Data

Consumption in this context is the ability to access and retrieve data. The consumption methods presented in this section can be found elsewhere classified as publishing methods such as in [Rietveld, 2016]. This is similar to the ”chicken or the egg” problem. In order to be able to apply certain consumption methods, Linked Data providers usually have to follow certain procedures on top of the four principles (and they follow these procedures in order to be ”consumed” in these ways). In this thesis, they are considered more as consumption mechanisms because their variations are more noticeable at this stage of the process.

Before introducing data integration in the next chapter, it is necessary to understand how to access each of the data sources separately, which are in the context of this thesis semi-structured and Linked Data sources. It also helps explaining the choices made in designing the approaches used in this thesis and their advantages and limitations. Therefore, this section shows some of the popular ways and technologies available to retrieve or query Linked Data sources.

2.4.1 Crawling Pattern

It is the most straightforward method whereby users download, process and parse the RDF files in order to get the results. From a data provider point of view, it is about hosting serialised RDF files of Linked Dataset on the Web. Even though it seems a simple way of publishing Linked Data, Rietveld [2016] stated that the majority of RDF files published through this mechanism fail to follow the standards and best practices of the Linked Data paradigm. He also stated some of the common errors that can be found in files which are: incorrect HTTP headers; published in a corrupt compressed archive and containing duplicate triples or serialisation errors. The crawling pattern also has advantages of being relatively easy to implement on top of the downloaded RDF files and that is not related to the status or the performance of any remote server.

2.4.2 On-The-Fly Dereferencing

On-The-Fly dereferencing pattern is comparable to the functioning of the Web of documents. It conceptualises the Web as graph of documents that consists of dereferenceable URIs [Göçebe *et al.*, 2015]. As introduced as part of the referenceable URIs (see Section 2.2.4), dereferencing means that a description of a resource identified by an URI is recovered using the HTTP GET request in a machine-readable format (as an RDF file for example) and optionally in human-friendly format.

In this pattern, the query is executed by dereferencing the URI address in order to access the RDF file, then follows the URI links by parsing the received file on-the-fly [Hartig *et al.*, 2009]. It can be relatively easy and fast for the server to process if it is used as subject pages access (a simple index lookup) [Verborgh *et al.*, 2014b]. It can also be, however, complex and slow if dereferencing thousands of URIs in the background [Göçebe *et al.*, 2015; Heath and Bizer, 2011]. This pattern is implemented by Linked Data browsers such as Marbles[25].

2.4.3 Using SPARQL to query Linked Data

SPARQL endpoint is a protocol service and one popular method for querying Linked Data sources. It enables users to query a knowledge base via the SPARQL language. SPARQL endpoint is viewed as a machine-friendly interface, as frequently one or many machine-processable formats are offered in expressing the results. Many triple stores offer a SPARQL interface, such as Jena TDB [Grobe, 2009] and Virtuoso [Erling and Mikhailov, 2010]. A human-readable presentation can also be implemented.

Although SPARQL endpoints have shown many capabilities in terms of the ability of expressing and running complex and federated queries, they are often criticised about their performance and availability. At least two scenarios are able to reveal SPARQL endpoints limitation. The first is in case where the query is asking for a considerable amount of result sets, or multiple queries are sent to the same data source. The second is when running federated queries on multiple sources. Because SPARQL endpoints concentrate all their query processing tasks on

[25] https://sourceforge.net/projects/marbles/

the server side only [Beek *et al.*, 2016], the execution of queries can be slow [Heath and Bizer, 2011] or can trigger an interruption as a result of restrictions set by data source to prevent the service from overloading or collapsing. Consequently, as confirmed by a study carried out by Verborgh *et al.* [2014a] who estimated that one and a half (1.5) days each month is the average downtime of SPARQL endpoints servers. This is one impetus for systems like LDF.

2.4.4 Querying through Linked Data Fragments (LDF)

Although it can be argued that LDF approach is more related to the publishing stage, its benefits can be seen in the consuming part; hence, it is classified in this section. LDF is a publishing method ”that allows efficient offloading of query execution from servers to clients through a lightweight partitioning strategy” [Verborgh *et al.*, 2014b, p. 1]. It can be described as a compromise between the limited subject-based Linked Data dereferencing and the difficultly of the scalable server-side SPARQL execution [Verborgh *et al.*, 2014b].

2.4.5 Linked Search Engines

Linked Data search engines are not a method of consuming Linked Data, but rather a category of applications, generally built upon the crawling pattern, facilitating to some extent the exploitation of data in this paradigm. They crawl RDF data on the Web and aggregate it. The retrieved data can be queried by following the links or by keyword search. The results can be presented in various forms depending on the application. Many examples can be listed in this section, for example: Swoogle [Ding *et al.*, 2004] and Falcons [Cheng *et al.*, 2008].

2.5 Un-migrated Sources: Semi-Structured Data

Semi-structured data are ”schema-less” data [Buneman *et al.*, 2001], meaning it does not have any rigid and predetermined schema upfront, which is one of the main advantages that make them very popular. They are ”self-describing” [Buneman *et al.*, 2001], which means the structure and the values are embedded in the same file. These characteristics made semi-structured data the most suitable and natural data model to accommodate heterogeneity [Chung and Jesurajaiah, 2005] and an important feature of the Web [Ya-qin and Wen-yong, 2010]. XML and JSON[26] are the main data models representing semi-structured data. Both of them are hierarchical and can be easily parsed [McMullen and Hawick, 2013; Ray, 2003] due to the availability of tools and libraries.

This subchapter begins by introducing the RESTful API. Then, the two most utilised semi-structured technologies on the Web, XML and JSON, are presented.

2.5.1 RESTful Web APIs

A Web API is a broad class of Web services and interfaces. In the context of the thesis, a Web API can be defined as an interface of a service that consists of a set of HTTP request messages along with a definition of the structure of response messages [Cao *et al.*, 2013]. The most used technologies in representing the outputted messages JSON or XML (see **MQ2** in Section 1.2). This interface is standards-based application-to-application programming interface, meaning it can be called from other programs [Burghardt *et al.*, 2005].

A Web API is considered a RESTful service [Richardson and Ruby, 2008] when conforming to the REST[27] architecture principles [Fielding and Taylor, 2000], being client-server based communication, statelessness of the request and the use of a uniform interface. The common technology used to implement RESTful Web services is HTTP [Maleshkova *et al.*, 2010].

RESTful[28] Web APIs are one major technology that makes use of semi-structured data in their data exchange. Many formats are utilised for this. XML and JSON are, however, the most frequently used mechanisms. They are the preferred representation for machine-readable data [Trifa *et al.*, 2010].

The steady increase of the number of Web APIs, and RESTful web services in general [Wu *et al.*, in press], suggests that the amount of semi-structured data is constantly growing on the Web. More precisely, the number of Web APIs continued to increase even in post-Linked Data

[26] JavaScript Object Notation

[27] Representational state transfer

[28] A service based on REST is called RESTful

era (after 2006) (see Figure 1.1 in Section 1.2). Many explanations can be put forward, for example: some use cases are more suitable to be implemented in a Web API architecture, the implementation of a Linked Data source is less accessible, or the Linked Data paradigm is not a success and does not respond to the developer needs etc. One conclusion that can be drawn is that relatively to Linked Data, older data sources being semi-structured data sources are still growing.

2.5.2 XML

XML is a mark-up language that allows users to define a set of tags which describe arbitrary document structure [Bray *et al.*, 1997]. It is designed to be ”eXtensible” by allowing to create user-defined forms by defining various entities, tags or elements [Van der Aalst and Kumar, 2003]. XML is a labelled tree, where each tag corresponds to a labelled node in the data-model, and each nested sub-tag is a child in the tree [Decker *et al.*, 2000].

The flexibility and the simplicity of defining an XML structure along with the availability of tools for manipulating it, made of XML an effective and a popular mechanism in cross application communication and information exchange.

2.5.3 JSON

JSON is a popular format for data serialisation and a lightweight, text-based, language-independent data interchange format. It is widely used as an alternative to XML [Guinard *et al.*, 2010], not as a mark-up language, but as a data exchange format particularly when dealing with existing Web services [Sumaray and Makki, 2012]. Soon after its creation, JSON was adopted by many well-known companies, such as: Google[29] and Yahoo[30] [Robal and Kalja, 2009], due to its efficiency yet simplicity in representing semi-structured data.

[29] https://www.google.com/

[30] https://www.yahoo.com

2.5.4 Structuring Semi-structured Data

The focus of this section is to give a brief overview about the tools and methods that allow the transition from semi-structured data model, particularly JSON and XML, to RDF data model.

The problem of converting hierarchical, or tree-based, data models to graph-based data models has existed for more than a decade. Various solutions have been proposed [Bohring *et al.*, 2005; Cruz *et al.*, 2004; Johnson, 2013; Van Deursen *et al.*, 2008] that can be classified into two categories: Fixed RDF transformation and ontology-dependent RDF transformation.

The systems of the first category perform syntactical and generic conversions from one data model and format to another. The transition, in this category of approaches, consists of mainly restructuring and reorganising different components of semi-structured data (namespace, root, tags, attributes and values) into a subject, predicate and object RDF structure. This operation is not considered challenging as an XSLT[31] script or the combination of JSON/XML parser with Jena framework can achieve an acceptable result. The disadvantage of this operation is the fact that no meaning will be associated with the resultant RDF file. Many examples of tools appertain to this class of systems can be stated including [Breitling, 2009] or the java library XmlToRdf[32].

The second class of systems are based on ontologies when converting semi-structured data schema, frequently XML, to an RDF schema. It is a challenging task to project the representation of concepts and the relationships between them of a given ontology while converting from one data model to another. This is what Van Deursen *et al.* [2008] attempted to achieve, for instance. The system they proposed takes as inputs an XML file, an OWL ontology, and the mapping document describing the link between the XML file and the ontology. RDF instances conforming to the OWL ontology are the outcome of this tool.

[31] Extensible Stylesheet Language Transformations

[32] https://github.com/AcandoNorway/XmlToRdf

2.6 Linked Data Challenges

It is not the role of the approaches proposed in this thesis to address most of the challenges described below. It is essential, however, that they take them into consideration when Linked Data datasets are queried or retrieved. For example, the schema matching approach (see Chapter 5) does not remove the semantically repeated properties from the data sources, they are only discarded when they become part of the system.

These challenges are not associated with the publishing or the consumption parts as most of them are common and their effects can occur in both operations.

- **Incompleteness:** It is the logical result of the dispersed and distributed nature of Linked Data. Real world entities can be seen from different points of view which can decide the focus of the elaboration of details. Thus, they are frequently only partially described in one data source.
- **Freshness of the data:** Some authors, such as Liu [2015], view it as part of a broader challenge called inconsistency. Inconsistency can be related to various forms of inter and intra source data conflicts, including out-of-date predicates and objects. The freshness of the data, in this thesis, does not only indicate the outdated predicates and objects (for example: `dbo:populationTotal`), it can be a challenge that is associated to an entire Linked Data source. DBpedia, for instance, is a publicly-editable, living data set, being extracted from Wikipedia by an evolving codebase, so results may (and will, and have) change over time.
- **Data and Properties Redundancy:** It generally signifies that the same real world entities are represented in multiple data sources. But in this thesis, it is not just limited to this definition. The redundancy can also be semantic. Two predicates appertain to two different Linked Data vocabularies can express the same meaning.
- **Incorrectness:** This refers to pure errors. It can be argued that these errors can occur in any data model or paradigm and that is not only a concern for Linked Data. On the other hand, this challenge is noteworthy in Linked Data because these errors can be propagated

from one source to the other due to copying and to the linking aspect of this paradigm.

- **Linking resources:** As put by Zhao [2010], connecting URIs leads to bridging the gap between a local namespace and the Linked Data. Although many tools have emphasised this need and addressed this problem over many years, it is still challenging to establish links between resources in the Web of Linked Data. This is due to many reasons, including the four previously stated challenges plus others, such as the scalability of data published as Linked Data. This challenge is addressed more specifically in Chapter 6.

2.7 Summary

This chapter introduced two important data models on the Web: Linked and semi-structured data. It also went in depth into their technologies and components, their different features and challenges. The author also discussed how to publish data as semi-structured or Linked Data, and the different ways of consuming them. From this background review, the author identified the criteria that need to be taken into consideration in designing the modules of the approaches proposed as part of this thesis.

The next chapter explains how to facilitate following the principles when contributing to Linked Data. Moreover, it discusses the process of how the two presented data models (semi-structured and Linked Data) can be used together to respond to the user's queries.

Chapter 3

Data Integration and Interlinking

Of course, shopbots and auction bots abound on the Web, but these are essentially handcrafted for particular tasks; they have little ability to interact with heterogeneous data and information types. Because we haven't yet delivered large-scale, agent-based mediation, some commentators argue that the Semantic Web has failed to deliver

Nigel Shadbolt

3.1 Introduction

This literature review chapter turns its attention to two of the most researched subjects or tasks in the Semantic Web generally and in Linked Data in particular: data integration and interlinking. In the work presented in this thesis (in particular, see Chapters 6 and 7), both these two tasks utilise schema matching or at least one of its subcategories, as Figure 3.1 shows. Hence, this present chapter starts by defining and investigating the common problem of schema matching and the different methods utilised to address it. Then, from Section 3.3 to Section 3.6, the chapter defines data integration and its concepts, and presents a classification of data integration approaches. Those sections also show how the current challenges of data integration go beyond

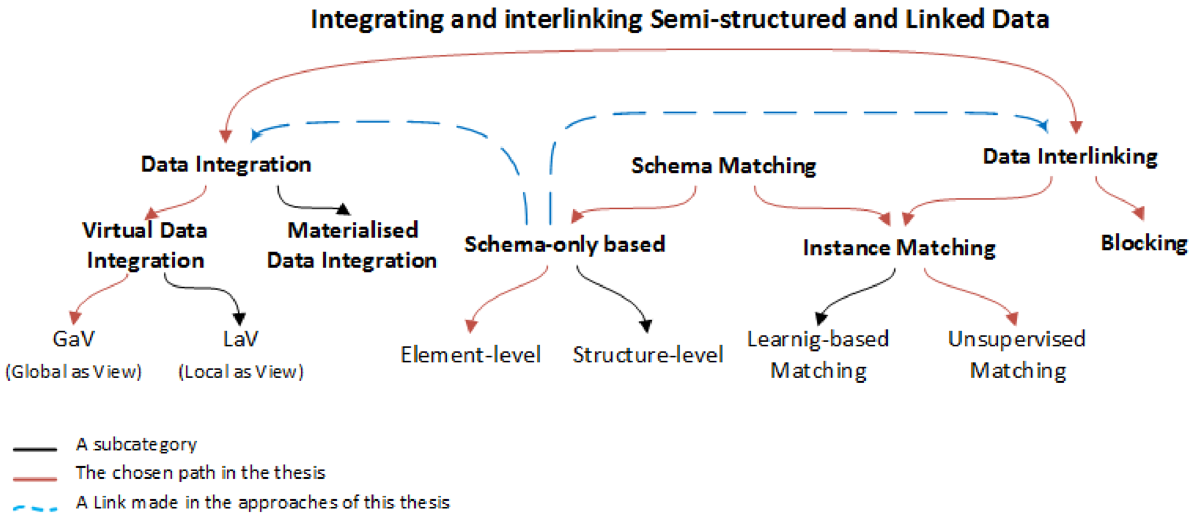


Figure 3.1: A diagram the shows the relationship between different terms used in this chapter [Author, 2017].

the issue of heterogeneity. The remaining sections of this chapter cover data interlinking, taking an in-depth look at its stages and reviewing its subtopics.

3.2 Schema Matching

Schema matching is the process of finding semantic correspondences [Do and Rahm, 2002] (frequently equality) between the elements of two or more schemas which describe datasets originating from the same or different dispersed data sources. The model and format of the datasets may differ as a result.

3.2.1 Difference between Schema Matching, Mapping and Integration

This section resolves the ambiguities which exist across the terms schema matching, mapping and integration. The common term in all these subareas is schema; this term can refer to a database schema, a generic model or an ontology [Atzeni and Torlone, 1997; Giunchiglia *et al.*, 2009; Madhavan *et al.*, 2001]. Schema mapping is the process of finding relationships between the instances of the elements of two schemas. Matching the elements of the schemas is a fundamental requirement for schema mapping [Bellahsene *et al.*, 2011]. For example: schema matching finds that the academic grades of a French university student (20 points grading scale) in the source schema corresponds to the divisions of the grades of a UK university student (100% grading scale) in the target schema. Schema mapping utilises this match in order to specify the division rate that relates the instances of the source with the target properties. In this example (academic grading in France compared to the UK) such a division rate would be five (5). Schema integration consists of the merging of a number of sources schemas into one integrated schema, called the global schema (see Section 3.4.3). The latter is an important subarea, a module and/or step of data integration. In some scenarios, including the approach presented in this thesis (see Chapter 5), schema integration is essentially a schema matching process that produces an integrated schema via the generation of the constraints and rules that specify its creation.

3.2.2 Schema Matching Qualitative Evaluation Measures

The measures described in this section are utilised in this thesis to evaluate the approaches proposed in Chapters 5 and 6. Figure 3.2 summarises the evaluation methodology for schema matching techniques and illustrates the components of the evaluation equations. Three measures [Do *et al.*, 2002] are utilised to verify the effectiveness of a schema matching approach:

- **Recall**

The recall measure represents the ability to retain the true matches, or true `owl:sameAs` links in the Linked Data terms. It is calculated using the equation below:

$$Recall = \frac{The\, number\, of\, true\, sameAs\, discovered\, links}{The\, number\, of\, actual\, links}$$

$$= \frac{|\,true - positive\,|}{|\,true - positive\,| \cup |\,false - negative\,|}$$

- **Precision**

The precision measure represents the percentage of true matches that lie within the discovered links. The equation to calculate the precision is similar to that used to calculate recall, but

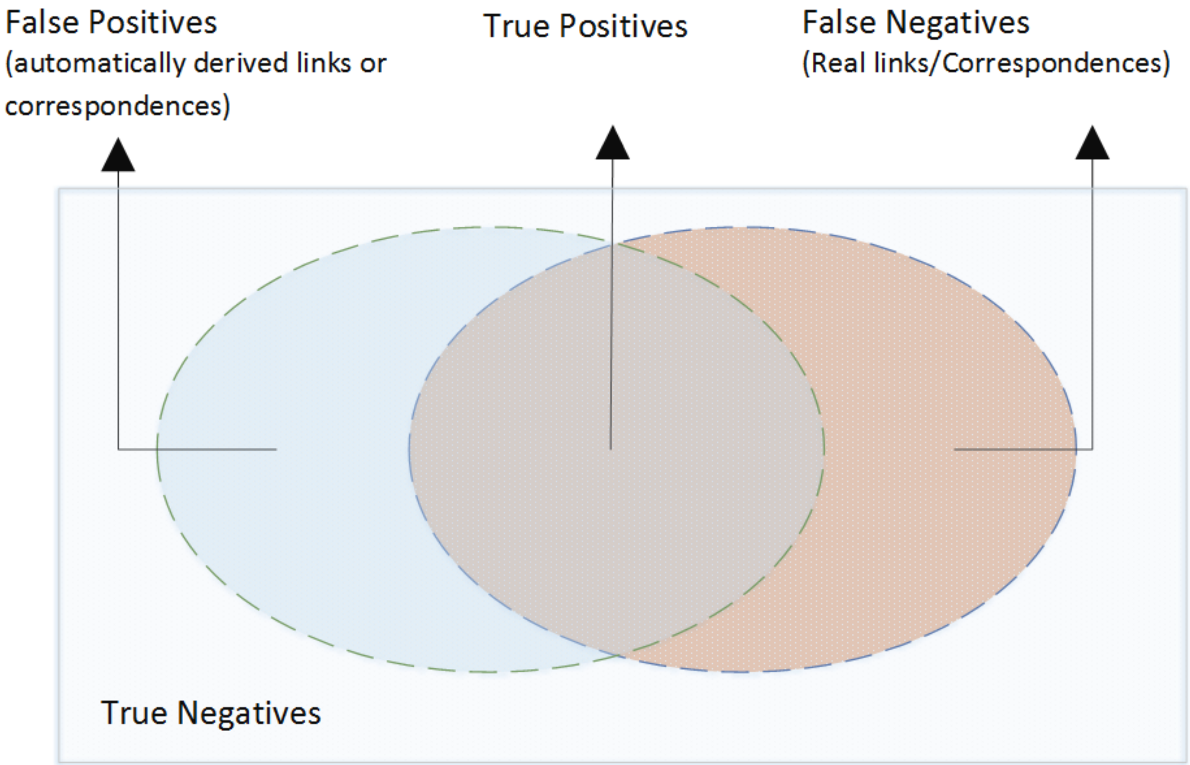

False Positives
(automatically derived links or correspondences)
True Positives
False Negatives
(Real links/Correspondences)
True Negatives

Figure 3.2: A comparison between real and automatically derived correspondences [Do *et al.*, 2002].

instead of dividing the number of true matches by the number of actual links, for precision they are divided by all the discovered links.

$$Precision = \frac{The\, number\, of\, true\, sameAs\, discovered\, links}{The\, number\, of\, all\, discovered\, links}$$

$$= \frac{|\, true - positive\, |}{|\, true - positive\, | \cup |\, false - positive\, |}$$

- **F1 score**

Neither precision nor recall separately will accurately reflect the match quality since their values can be maximised at the expense of each other (high recall can be easily achieved at the cost of poor precision by returning as many candidates as possible, and to maximise the precision at the expense of poor recall the matcher may return only a few correct correspondences) [Do *et al.*, 2002]. Therefore, it is necessary to take into account both measures or a combined measure. F1 is the combined measure and the harmonic mean of the recall and the precision.

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

3.3 An overview of Schema Matching Techniques

Schema matching has been approached in various ways, many of which have prompted the creation of important research subareas. This section provides an overview of the most popular categories of such approaches and gives examples of tools designed to target semi-structured and Linked Data in particular.

One of the most popular classifications is the one which distinguishes between schema-level and instance-level matching. Within the schema-level classification, two sub-classes of approaches can be identified: structure matching and element (or property) matching. Structure matching generally uses a knowledge base or ontologies to generate mapping rules. This approach has problems with consistency [Dong and Hussain, 2014] as it cannot sustain the dynamism, freshness, and large scale of data.

There is another approach which does not use ontologies to support the mapping but rather employs other means of finding correspondences between the properties of the schemas, including: similarity measurements (syntactic or semantic), linguistic matching, constraint-based matching [Bernstein *et al.*, 2011], etc. In this thesis, the term property matching (or alignment) refers to this type of approach.

Other approaches are hybrid matching, referring to methods which utilise various disparate factors in the matching process. Discriminations between the systems of this kind can be established according to the degree of the automation of the process – manual, semi-automatic, automatic – or according to the kind of links they create – relationship or identity. Approaches can also be classified as either pairwise/2-way schema matching (where the maximum number of inputted schemas is two) or holistic/n-way (in which more than two schemas can be considered).

Ontology matching is an important subarea of schema matching, and has been used in systems such as: AgreementMaker [Cruz *et al.*, 2010], RiMOM [Wang *et al.*, 2010], SERIMI [Araujo *et al.*, 2011], etc. These have all been presented as part of the annual Ontology Alignment Evaluation Initiative (OAEI) event. These approaches address the problem of finding correspondence between ontologies and discovering `owl:sameAs` relationships between Linked Data resources.

There have been some approaches, addressing the problem of data integration generally, which incorporate schema matching: e.g., in Smart Cities [Bischof *et al.*, 2014; Kettouch *et al.*, 2016; Nemirovski *et al.*, 2013]. But such approaches frequently tend to consider ontology matching as a secondary issue. Hence, some of this work lacks a detailed description of the matching and of the reconciliation process.

3.4 Data Integration

Data integration is the process of providing homogeneous access to a set of heterogeneous and autonomous sources [Calı *et al.*, 2004]. Data integration is more than schema integration, as pointed out in Section 3.2.2. It is the combination of many operations, including the latter, but with the addition of others such as: formulating and distributing the queries, reconciling the output and finally collecting and displaying the results.

Data integration is sometimes referred to by the term data interoperability since such topics share approximately the same issues and objectives. Data integration is still considered to be a very challenging task, despite it having been a major research subject (as a database problem) in Web Semantics for a number of years [Kalja *et al.*, 2014]. The rise of new data models and representations, as well as new challenges in terms of precision and performance, has kept this research area continuously active.

As Figure 3.3 shows, data integration should be viewed as a black box from a user's perception. It needs to hide the complexity and provide a single point of access to many data sources. Note that in Figure 3.3 the word ”integrate” is underlined. The reason for this is that by changing that word to ”collect” the black box becomes a data federation system rather than a data

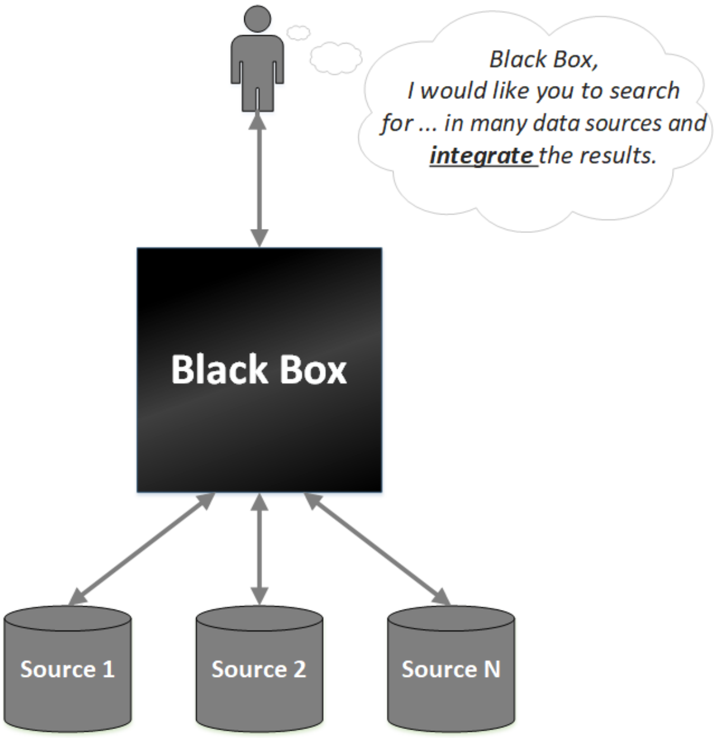


Figure 3.3: How data integration should be seen from the user's point of view [Author, 2017].

integration system. Data integration not only gathers data, but also offers a logically unified view to the query process.

Data integration is a process that can be divided into two phases. The first phase is top-down and aims at preparing, distributing and running the queries and requests. The second phase is bottom-up and aims at reconciling the heterogeneity of the results, running the queries in relation to the unified view and preparing the outputs to be displayed. This component in data integration can perform different tasks depending on the phase in which the system is running.

Two main methods have been proposed (and researched) for solving the problem of data integration and reconciling heterogeneity: the materialised architecture and the virtual architecture.

The materialised architecture, also known as data warehousing, is a consistent, but non-volatile, solution which integrates data and stores it in a single information repository [Poe *et al.*, 1997]. The latter serves as a materialised view to one or more sources [Gupta and Mumick, 2005]. It is frequently implemented in decision-support systems and for OLAP[33] queries.

[33] Online Analytical Processing

Queries, in the materialised integration architecture, are executed on, and loaded in, a single central database which stores, in advance, all the data which can be retrieved. Materialised views are physical structures that store data retrieved at a specific time. Hence, the major issue with using this approach is the freshness of the information and the possible resource implications for very large datasets.

Virtual (mediator-based) data integration is referred to using a number of other terms, such as centralised architecture or distributed query processing. In this architecture, the integration system provides a ”unified and transparent view to a collection of data stored in multiple, autonomous and heterogeneous data sources” [Lenzerini, 2003, p. 14]. This unified view, known as the global schema, decomposes the inputted query into a number of sub-queries, which will all then be distributed, and then executed, on the different sources considered – before the eventual combining of the outputs. This gives the user transparency and the impression of querying a single data space. One of the features of virtual integration is that the system has no control over the participant sources. This can be seen as a drawback in some use cases but an advantage in others (which require the preservation of the autonomy of the data sources). Virtual data integration's most well-known positive, and crucial for the present contributed research, aspect is the flexibility to add or accommodate new sources.

The characteristics of the data considered in this thesis, particularly Linked Data, suggest that virtual integration is the more suitable approach. Linked Data's flexibility in publishing new information and creating new vocabularies necessitate a view which is repairable and able to be changed over time.

The focus of the remainder of this thesis, addressing the problem of data integration, is on virtual integration. Figure 3.4 suggests the position of these components in the data integration process. The next sections describe four main concepts that are associated with, or define to some extent, virtual data integration:

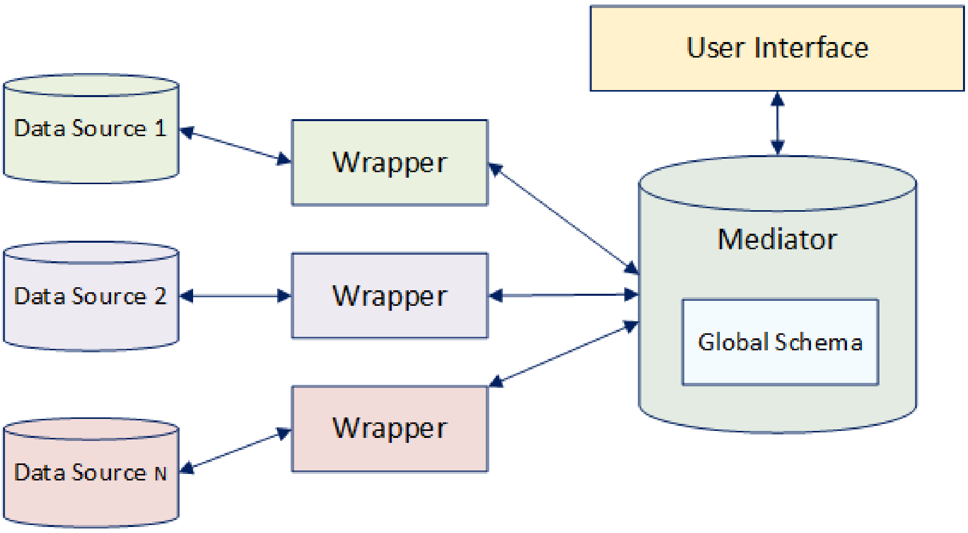


Figure 3.4: General architecture of virtual data integration [Katsis and Papakonstantinou, 2009].

3.4.1 Usable Front End

The user interface, as observed, is generally not considered as a component or as a major part of a data integration system. It is mentioned in this section as a separate concept because of the new challenges generated by the appearance of new data paradigms – which suggest more work should be invested in this component of systems. Data integration systems, taking into account Linked Data sources, are different from the conventional environments for which developers write queries, knowing the search requirements and the data schema and structure. This means that they also have different challenges to overcome in terms of offering a usable interface that does not significantly limit expressivity (see Section 3.5). Having both a usable and an expressive interface that offers different features to assist users in their access sessions, and allows more structured queries and thus less confusion in the integration stage is important. Considering its order in the running of the system, the user interface can deeply influence the results of the following steps. In the second phase of data integration, the front end focuses on result visualisation. Presenting information that can be clearly interpreted by the user is a major consideration in relation to usability.

3.4.2 Mediator

The mediator is a core component of data integration systems. Positioned as an intermediate and a broker between the users and the sources, the mediator has the function of abstracting the user from the fact that information is coming from various different sources. Conventionally, the word mediator was used interchangeably with the term global schema. In the context of this thesis, however, the mediator is the middleware that comprises the global schema and other modules that together act as a transparent interface to a set of data sources. The mediator has many roles. These can be summarised using two headings:

- accepting and processing the user query before distributing it to the adapters (or wrappers); and
- collecting and reconciling the outputs in order to present them as results in the form requested by the user.

The mediator can involve a pre-processing step. The pre-processing role can vary from one approach to another. In most cases, however, it is needed to either formulate and clean the queries that will be sent to the sources, or/and the reverse, to prepare the sources to be processed. Either way, the execution of effective pre-processing is generally important before any further operations are carried out.

3.4.3 Global Schema

As introduced in section 3.2.1, the global schema is the virtual view that temporarily stores and represents the information presented from the sources. The global schema is the component responsible for representing the outputs of the different sources into a single and uniform temporary storage. Various methods of expressing the relationships between the global schema and the sources schemas have been proposed – to respond to different conditions and use cases. These methods can be classified into two categories:

- **Local as View (LaV):**– This is the paradigm whereby the local schemas are described as views of the global schema (as illustrated in Figure 3.5). Thus, new sources can easily be integrated and added.

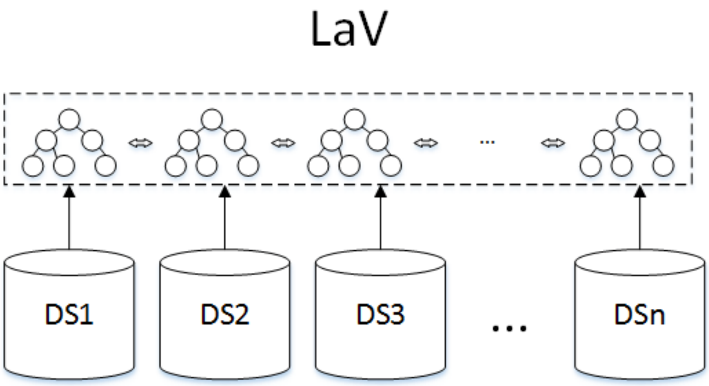


Figure 3.5: Local as View global schema design [Author, 2017].

- **Global as View (GaV):** – Here, the global schema is described according to the local schemas (as shown in Figure 3.6). The advantage of GaV is simplicity of query rewriting [Wu *et al.*, 2012] and decomposing [Chou, 2005].

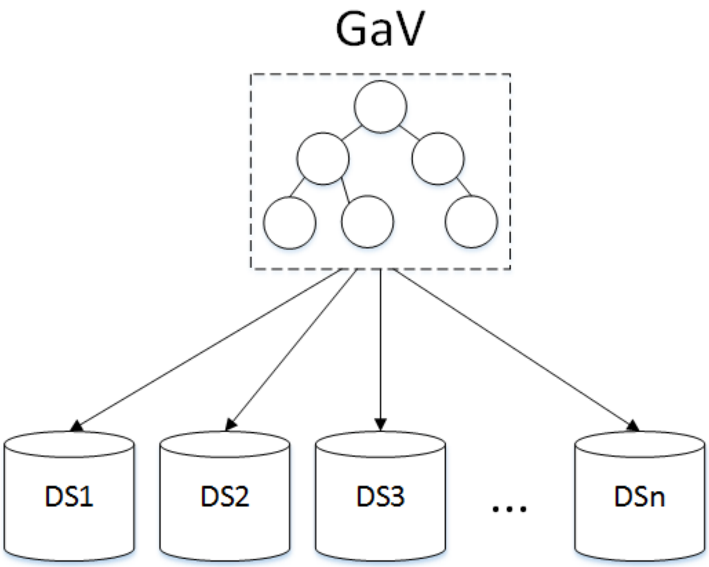


Figure 3.6: Global as View global schema design [Author, 2017].

The global schema is situated at the centre of the virtual data integration architecture and it represents the transition from the heterogeneity of the data sources to the uniformity (of the global schema). For this reason, it is crucial to choose an appropriate structure and format for representing the global schema that suits the characteristics of the data models taken into account. Other requirements ought to be considered, such as the flexibility to manipulate the chosen data structure and format (for example: the availability of tools to parse it).

3.4.4 Wrapper/Adapter

An adapter is a program, specific to one data source, which conventionally has two roles depending on the stage of the process. The first role is to connect the mediator with the source so the query can be executed. This includes the adapter's function of being responsible for translating between the mediator query language and the query language native to the data source [Ashish and Mehrotra, 2010]. An example of an adapter would be that of a Web API source in a locally based application which must establish an HTTP connection with the Web server so that requests can be sent. The second role, which adapter components are most known for, is to extract and parse the results of a query so that they are formed and structured in a suitable form for the user. The position of the adapter is that it is a process situated between the mediator and its data source.

3.5 Current Challenges in Linked Data Integration

One of the primary reasons for the introduction of the Linked Data space was in order to promote interoperability and uniform access to many RDF sources using one query language. Yet, a new type of heterogeneity has arisen due to the distributed nature of the publishing of Linked Data and the use of different vocabularies and structures to represent it. This area of research still has an important emphasis on design approaches to reconcile the heterogeneity within Linked Datasets. The following are the eight major challenges that this thesis identifies in regard to most of the current data integration systems:

- **Decentralisation and Autonomy of the Sources:** It is paramount for data integration systems, in or out of the Web of Linked Data, to deliver fresh and up-to-date information. This cannot be achieved without addressing the dynamism of the relationships between the mediated schemas, or those between a central repository and its sources. The distributed nature and the autonomy of Linked Data sources make it unlikely that the use of one model to represent all the data across a particular domain can be sustained. Each source has its specificities, conditions, and a different vision in regard to the way to expand. The internal links within one Linked Data namespace can be consistent and easily maintained. The external links, however, represent a challenging task, given that they connect two vocabularies, models or views that are situated in separate locations and are regularly changing.

- **Heterogeneity:** Many different and incompatible data and knowledge description formats exist due to both legacy systems, on the one hand, and the increasing variety of new approaches on the other. Various types of heterogeneity may occur at many levels, including structural, syntactical and/or semantic mismatches; the access method; the language; and/or the protocol [Macura, 2014].

- **Usability for end users:** This does not merely indicate that the system ought it to be easy to operate, but also that it should provide sufficient information for the users to appropriately interpret the outputs. The most usable method for accessing data sources, arguably, is keyword search [Freitas *et al.*, 2012], as no pre-knowledge is required to use this [Macura, 2014].

- **Expressivity:** This is defined as the ability to ”query datasets by referencing elements in the data model structure” [Freitas *et al.*, 2012, p. 26]. A system can be considered expressive if it helps the users to make their queries more specific and structured and helps them to provide more details when querying data sources. In addition, defining and limiting the domain can reduce the semantic conflicts; therefore, this increases expressivity. Targeting domain-specific knowledge is one of the characteristics of systems with the ability to perform complex semantic interpretation and inference [Kaufmann and Bernstein, 2010].

- **Adaptivity and the degree of Automation:** Here, this can be defined as the flexibility to accommodate continuous changes in data structures and models. This is an essential criterion, particularly when Linked Data namespaces are amongst the sources, due to the increasing expansion of the Web of Lined Data. It is also crucial for the potential to add further sources in the future. Offering tailored integration views [Ziegler and Dittrich, 2007] for specific sources or cases can be seen as a limitation.

- **Privacy:** Privacy of personal data access is a recent major issue in terms of this area. Since most data integration systems are designed to be used by the public, the solutions are required to not include, mine or index sources which contain personal data and other sensitive information. Privacy is a big issue that can be met by including only publicly accessible sources – by, for instance, using Web APIs and SPARQL endpoints.
- **Implementability:** This is a common data integration issue and is of particular prominence in the generic and highly automated integration approaches, especially when addressing different data structures alongside Linked Data. It is not a problem in the cases of study or task-specific systems.

3.6 An overview of Data Integration Approaches

Many solutions, intended to solve the problems of data integration and reconciling heterogeneity, have been proposed and researched (see Section 4.6). Since data integration is a fundamental issue for any deployed information system and a long-studied topic [Brennan *et al.*, 2011], various ways of classifying the available systems and approaches proposed in this area have emerged. For instance, data integration systems can be categorised according to:

- The data models and the formats considered, or;
- The mediation method, or;
- The autonomy of the sources and the degree of automation of the process.

The various data integration approaches are reviewed in this section in the context of search systems, with the exception of the last class – commercial data integration – where the contrast between the focus of the semantic research community and the commercial tools is looked at instead. Although not common to categorise according to the search mechanisms used, this classification scheme is best suited to the collection of the maximum number of works that are related to the research approaches proposed in this thesis. This classification accords with the type of application of the contributed approach (in Chapter 7) and allows for the covering of a wide variety of solutions. Four classes are identified in this section:

3.6.1 Document-Centric Search

This is also referred to as universal search. This category consists mainly of the popular search engines, such as Google, Yahoo and Bing. At the time that these were introduced, many other technological breakthroughs were able to connect users with that one data source which contained the information that they needed. Search Engines had the role of finding this one data source by searching for documents across the Internet using keywords. Although they were arguably an essential factor in the success of the Web, currently they are not competent to respond to all user queries. Data, originating from various sources, can be used complementarily to respond to the users' needs, and this feature is not supported by current search engines. Whilst search engines are able to perform federated searches across autonomous repositories, and some of them extend their search to cover ”non-document” repositories, they still lack the ability to interpret and join the information retrieved, semantically.

3.6.2 Semantic Search

Semantic Search is also referred to as Question Answering (QA) [Collarana *et al.*, 2016]. The collection of systems involved with semantic searching includes most of the Natural Language Processing (NLP) systems. A knowledge base [Collarana *et al.*, 2016] or an unambiguous ontology [Lopez *et al.*, 2013] is commonly used in this type of system to process the query and to integrate the outputs. The main limitation of these systems is revealed when addressing large open-domain sources where the system ontology or the knowledge base is unable to disambiguate the query. Frequently, this class of systems is used to tackle heterogeneity within Linked Data, whereas semi-structured data sources, accessed through Web APIs, are not considered. Various solutions can be mentioned which fall within this category [Lopez *et al.*, 2012; Schlaefer *et al.*, 2007; Shekarpour *et al.*, 2015]. The term heterogeneity is used, in regard to most of the approaches in this category, to refer to the ambiguities and discrepancies in describing Linked Data resources. These can lead to multiple, yet not similar, results for one query. The semantic search approaches propose tools for combining and ranking the results that give priority to the most accurate information.

Heterogeneity in the present thesis has a wider meaning, as discussed in Section 3.5, being the differences in the structures, syntaxes, access methods, languages, and protocols present not only internally between Linked Datasets, but externally in relation to semi-structured sources.

3.6.3 Hybrid Search

The term hybrid search indicates the ability to take in various data structures to respond to the user's query [Morbidoni *et al.*, 2008; Usbeck *et al.*, 2015]. Frequently, the various data structures referred to include both structured and unstructured data [Usbeck *et al.*, 2015]. In this thesis, the data structures in the definition of hybrid search are limited to semi-structured Linked Data (which is structured data). Bhagdev *et al.* [2008, p. 567] argued that the interesting aspect of hybrid search is its ability to overcome "an implicit limitation of most of the current literature, that is that semantic search must rely on metadata only".

This category of systems can be studied from many perspectives. The main aspect that will be discussed in relation to the existing tools that belong to this category, in Section 4.6, is the methods used to integrate different data structures.

3.6.4 Commercial Data Integration Solutions

Many products, some of them offered by well-known IT companies, can be listed in this section. DB2 information integrator (DB2II) and Oracle Integration are just two instances; these are provided by IBM and Oracle respectively. They generally support various data types and formats and they are fairly efficient in their query optimisations. On the other hand, commercial data integration products generally, including these two examples, offer solutions which instead of integrating data, combine and fuse data originated from heterogeneous and dispersed sources

[Friedrich and Wingerath, 2010]. They ”are essentially data federation tools that are still far from [being] data integration systems” [Poggi, 2006, p. 24]. The user of these products is still not provided with a logically unified view in relation to their queries (as discussed at the beginning of this chapter), but rather a tailored interface which accommodates different result types and formats.

More importantly, Linked Data sources, as far as can be seen from the documentation of these products, are not listed as part of the considered sources. The RDF data model, as highlighted in Sections 2.2.2 and 2.3.3, has distinct structural features that distinguish RDF sources from other data sources. Therefore, it requires different procedures to integrate its schemas and to sustain these changes. These procedures are not part of the scope of the commercial products as they are still experimental and under research and development.

3.7 Data interlinking

The interlinking of Linked Data is a subject which has been extensively researched by the Semantic Web community over the last few years [Lesnikova, 2016]. It is a fundamental concept of Linked Data, and a key factor for the success of the Semantic Web, to create typed links between the different data sources in the extension of the global data space [Bizer *et al.*, 2009].

The ideal scenario in publishing Linked Data is allocating a unique URI to every real-world entity. Having multiple identities for the same resource reduces its discoverability and therefore reduces significantly its value and the chances of it being re-used. This ideal scenario, however, is practically unachievable, considering the distributed nature of the Linked Data paradigm [Hu *et al.*, 2014] and the massive number of real word things that exist. Hence, alternative solutions, such as data interlinking, take place in order to provide `owl:sameAs` links, as illustrated in Figure 3.7, between items representing the same resources that may be situated in the same or in different data sources. `owl:sameAs` links, as provided by OWL semantics, allow the discoverability of references to identical resources residing in different machine readable data repositories. They are also used to materialise inferable knowledge and to potentially generate additional results [Umbrich *et al.*, 2012].

Data interlinking can be seen as a reverse of data integration. The term indicates the operation of discovering the machine counterparts of the same real-world object [Nguyen *et al.*, 2012b]. The term "data interlinking" is used by some researchers interchangeably with the term "instance matching" [Euzenat, 2015]; for others, instance matching is one stage of the interlinking process [Nguyen *et al.*, 2012b]. The interlinking referred to in this thesis represents the entire process and steps needed to establish similarity links between two resources. Instance matching and ontology alignment are techniques that can be used as steps to solve the problem of interlinking.

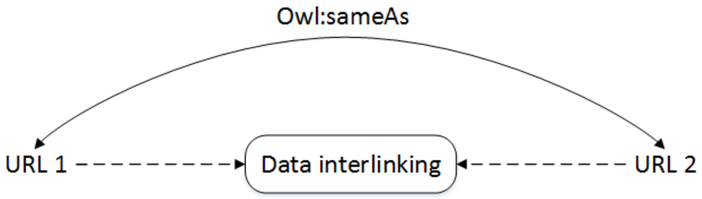


Figure 3.7: The Data Interlinking Process [Scharffe and Euzenat, 2011].

There is an ambiguity between the terms "link discovery" and "data interlinking". Demidova *et al.* [2015] indicated, basing on some examples, that entity interlinking is broader than link discovery since the latter focuses only on sameAs links. Ngomo and Auer [2011], however, stated that that link discovery approaches aim at finding typed links, including sameAs. One of the first and most popular approaches which labelled itself as link discovery, SILK (see Section 4.4.4), aligns with this statement. This thesis agrees with Ngomo and Auer [2011], and other authors, and considers link discovery a broader topic than data interlinking since it is not limited to identity links.

The rest of this section identifies and explains four of the main phases and concepts related to the interlinking task; two of these are fairly indispensable, being blocking and instance matching. Figure 3.8 shows how these stages are positioned in the data interlinking process. The use of ontologies and similarity measures are two popular methods employed to determine whether two descriptions or labels, respectively, refer to the same real-world entity.

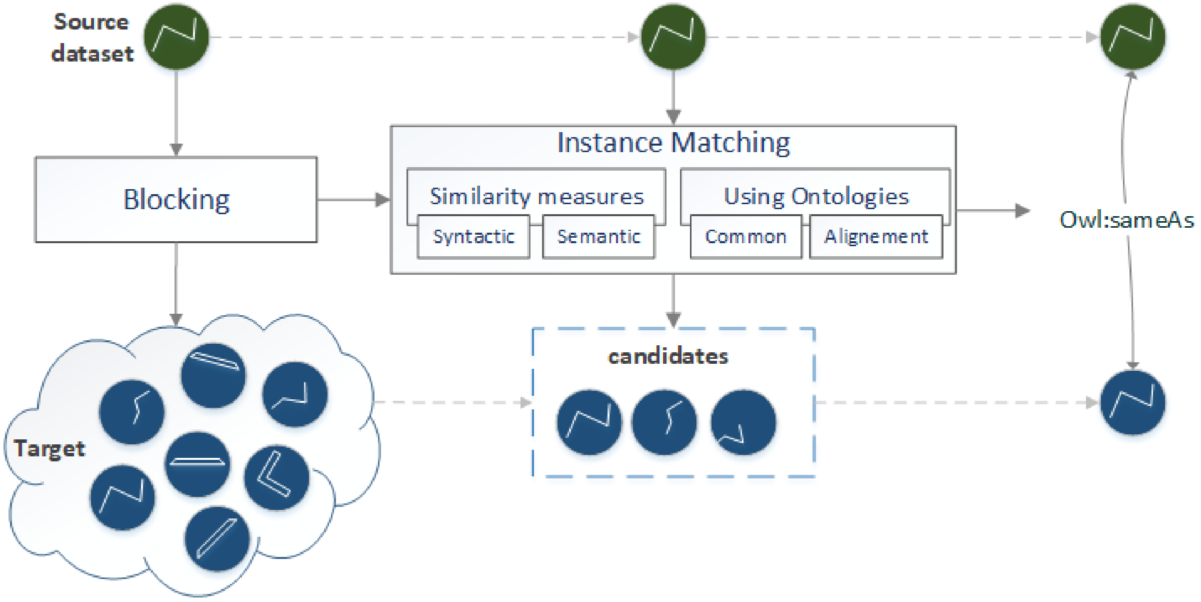


Figure 3.8: General architecture showing the different ways in which data interlinking was approached [Author, 2017].

3.7.1 Blocking

Blocking, in this context, means grouping similar objects, as Figure 3.9 illustrates, using a blocking key. It is the initial stage in the interlinking process whereby the number of candidates is reduced. As a result, a block that consists of a set of potential identity pairs of instances is generated. This is an important step as it affects the performance of the system, considering that the inputs of the heavy processing operations in the instance matching stage will have resulted from the blocking. The blocking stage aims to achieve two goals:

- **Reduction Ratio**

This measure represents the efficiency of the blocking. It quantifies the ability of a blocking algorithm to minimise the number of comparisons (in further stages) by removing obvious non-matches. More formally:

$$RR\,(Reduction\,Ratio) = 1 - \frac{The\,number\,of\,candidates}{The\,number\,of\,All\,pairs}$$

The number of all pairs = |S| x |T| (S and T are the number of inputs of the source and target datasets respectively).

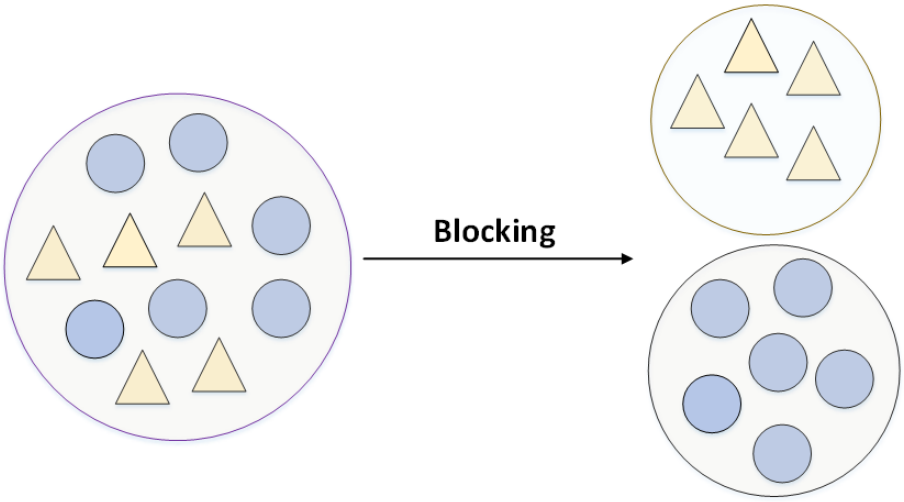


Figure 3.9: A digram illustrating and explaining the blocking step [Author, 2017].

The number of candidates indicates the number of pairs produced by the blocking. Generally, the blocking algorithms search for obvious non-matches and exclude these from the target set:

$$\mid T_b \mid \leq \mid T \mid \; (T_b \, target\, set\, produced\, by\, the\, blocking)$$

$$The\, number\, of\, candidates = \mid S \mid \times \mid T_b \mid \; (\leq \mid S \mid \times \mid T \mid)$$

- **Pair Completeness**

This value measures the number of true matches identified by the blocking algorithm versus the number of these that exist in the entire dataset, as described in the equation below:

$$PC\,(Pair\, Completeness) = \frac{C_m}{M}$$

C_m indicates the number of true matches candidates found by the blocking algorithm.

M refers the number of the true matches in the entire dataset. Therefore, theoretically: $C_m \leq M$

3.7.2 Instance Matching

Instance matching goes by a number of different names, these being: record linkage, data matching, the merge-purge problem and entity resolution [Christen, 2012; Elmagarmid *et al.*, 2007]. Instance matching is the problem of matching pairs of instances that refer to the same underlying entity [Scharffe *et al.*, 2013]. Instance matching is a technique originating from knowledge discovery and data mining algorithms [Elmagarmid *et al.*, 2007]. But recently, it has seen numerous applications in Web Semantics.

In data interlinking, this is the stage that immediately succeeds the blocking step. The matching status of the outputted pairs is verified in order to discover identity pairs [Nguyen *et al.*, 2012b].

3.7.3 Using Ontologies in Data Interlinking Alignment

Ontology alignment is a subarea of schema matching (see Section 3.2) and is the process of finding correspondences [Euzenat *et al.*, 2007] between concepts, properties, or instances in two or more ontologies, based on their similarities [Gunaratna *et al.*, 2014]. Ontologies in data interlinking are generally used to identify and compare instances that are part of the same classes, based on them having the same properties.

Using ontologies does not exclude the possibility of using other similarity techniques. Their utilisation can serve as a hint that materialises as a coefficient or as an element of a similarity algorithm, for example. Experiments have revealed also ”that the use of ontology features increases accuracy of instance matching for data integration” [Wang *et al.*, 2006, p. 1].

There are many methods by which ontologies can take part in an interlinking process. They can be summarised, however, under two broad headings. The first approach is to describe the two resources using a common ontology before the interlinking and matching process takes place, as Figure 3.10 illustrates. The second approach is to align the independent ontologies of the two resources to draw correspondence that will then be used in the interlinking, as Figure 3.11 shows.

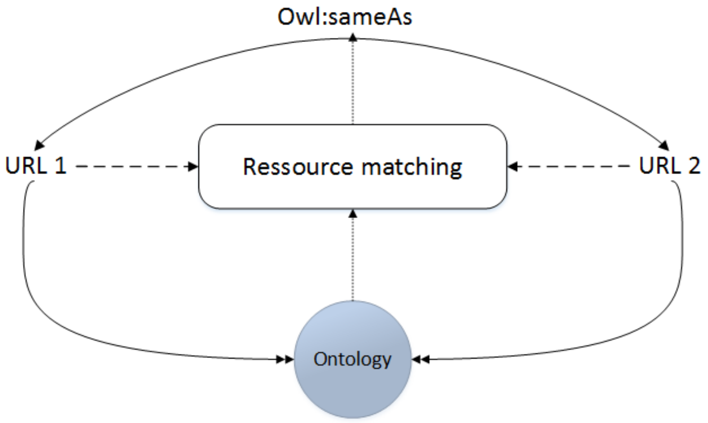


Figure 3.10: Data interlinking via a Common Ontology [Author, 2017].

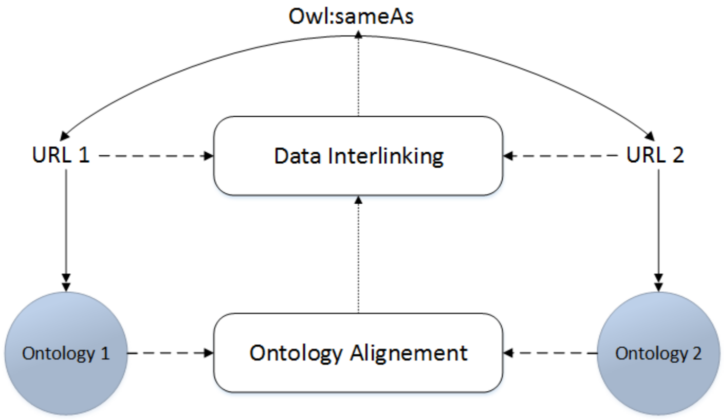


Figure 3.11: Ontology Alignment in Data Interlinking [Author, 2017].

3.7.4 Similarity Measures

Similarity algorithms are used to measure the distances between the properties of the elements of the source and target datasets. They can be sorted into one of two categories:

3.7.4.1 Syntactic Similarity

Syntactic similarity refers to the set and string similarity algorithms that are used in some instance matching approaches to calculate the syntactic distance between two predicates or entity labels. Jaro-Winkler is a popular example of a string similarity algorithm. This algorithm uses a mixture of string and set similarity [Nikolov *et al.*, 2008a], meaning that the compared values may be tokenised before the standard Jaro-Winkler algorithm is applied and the maximal total score is selected [Nikolov *et al.*, 2008b].

3.7.4.2 Semantic Similarity

In this group of algorithms and tools, the distance used is based on the meaning of the word rather than on its label or lexical form. Semantic similarity is an essential element of many Semantic Web topics, including Natural Language Processing (NLP) [Ma *et al.*, 2015] and information retrieval [Batet *et al.*, 2013]. The UMBC tool is an example of one kind of tool which has been proposed for semantic similarity measurement. It is constructed by combining the use of LSA word similarity and WorldNet knowledge. UMBC focuses on the semantics of the word but not on its lexical category. This makes it a typical similarity measurement mean for data interlinking and integration approaches which take Linked Data as at least one of their inputs, since the available vocabularies for describing resources in this paradigm vary between nouns and verbs. UMBC also provides a Web API whereby external systems can retrieve the similarity between two texts without the necessity of going through a re-implementation of the approach. The following URL provides a prototype of the UMBC API:

```
http://swoogle.umbc.edu/SimService/GetSimilarity?operation=api&phrase1=
SourcePropertyLabel&phrase2=TargetPropertyLabel
```

3.8 An Overview of Data Interlinking and Link Discovery Approaches

In the last few years, the problem of interlinking has been approached in a number of different ways, and this has led to the appearance of many sub problems and classifications. The most popular classification is the one which distinguishes between ontology matching and instance matching. An ontology may not be consistent and the solutions using it as a reference cannot precisely represent real-world knowledge [Dong and Hussain, 2014] nor can they sustain the dynamism and freshness of data.

Discriminations between the systems proposed can also be established according to the degree of automation of the process (manual, semi-automatic, automatic), and the kinds of the links they create (relationship, identity or vocabulary links). The existing data interlinking systems also use many techniques to optimise interlinking efficiency, these being: contextual matching, probabilistic matching and logic-based matching [Homoceanu *et al.*, 2014] etc.

SLINT [Nguyen *et al.*, 2012b], SLINT+ [Nguyen and Ichise, 2013], [Böhm *et al.*, 2012], SERIMI [Araujo *et al.*, 2011] and RiMOM [Zheng *et al.*, 2013] are some examples of relatively successful approaches for interlinking large-scale data. However, Homoceanu *et al.* [2014] and many other authors believe that their results are either still unsatisfactory or are based on data sets which are biased in order to achieve the highest possible precision and recall. These were all presented as part of the yearly event: Ontology Alignment Evaluation Initiative (OAEI).

EventMedia [Khrouf and Troncy, 2016] and the system proposed by Zhang *et al.* [2013] are projects aiming at interlinking data within a specific domain. As part of their projects, they both tried to find the most accurate weights to give to the properties in their particular domains.

There are a large number of approaches and tools designed to address the problem of matching and discovering links in the Linked Data space. What has been presented in this section are examples of arguably the most popular approaches that have addressed the problem differently or shown a promising result at some point.

3.9 Summary

This chapter explained two major Semantic Web research topics: data integration and data interlinking. It also described these approaches' different components and has highlighted their current challenges. Schema matching is another topic that was discussed – at the beginning of this chapter – because both data integration and data interlinking approaches use it. This latter is the module responsible for reconciling the structural heterogeneity in both the data integration and the interlinking approaches presented in this work. This chapter also presented a general overview of the different methods by which these topics have been approached along with a discussion of some of their advantages and limitations.

The author through this chapter showed the different paths, as well as their limitations and advantages, that can be taken in addressing data integration and interlinking of semi-structured and Linked Data. Along with Chapter 2, This allows the explication of some of the choices made in designing the contributed approaches in the present thesis.

The next chapter expands the literature review started in this chapter by analysing existing schema matching, data integration and data interlinking approaches.

Chapter 4

Related Works

> *You can have data without information, but you cannot have information without data.*
>
> Daniel Keys Moran

4.1 Introduction

This chapter looks into existing systems addressing schema matching, data interlinking and integration in Section 4.2, 4.4, 4.6 respectively. Each of these sections are followed by an analysis and discussion section where limitations of the related works studied are discussed and constructive observations are made. It is not the author's intention to cover all existing approaches, but to review and investigate the capabilities of a selection of the most popular and related systems with respect to their approach, evaluation method, scope, etc. Different metrics are utilised to select the reviewed approaches in each of the topics covered in this chapter.

4.2 Schema Matching

Many schema matching approaches were proposed throughout the last two decades. This review presents a selection of the popular approaches that significantly differ in their matching process or the evaluation method they used.

4.2.1 Cupid

Cupid [Madhavan *et al.*, 2001] is a generic and hybrid schema matcher that utilises a name matcher and a structure-based match algorithm. This tool uses a generic internal representation, called the schema tree. The schema tree is used to find the element matchings of a schema, using the similarity of their names and types at the leaf level [Manakanatas and Plexousakis, 2006]. However, since the structural matching phase is primarily based on the similarity of the leaves, Cupid cannot find accurate mappings if there are considerable variations in the structure of the given graphs [Le *et al.*, 2004].

Evaluation: Cubid was evaluated against two other schema matching, DIKE and MOMIS. The first part of the evaluation the systems were tested with some canonical match tasks considering very small schema fragments [Do *et al.*, 2002]. In the second part, the systems were tested using 2 real-world semi-structured (XML) schemas for purchase orders. A comparison was then drawn by exploring for the correspondences which could (or could not) be identified by a particular system [Do *et al.*, 2002]. Although Cubid showed a better efficiency against other systems by identifying all necessary correspondences for these match tasks, no quality measures were calculated.

Some pre-match effort was needed in Cupid to specify domain synonyms and abbreviations.

4.2.2 COMA

COMA [Do and Rahm, 2002] is a generic match system that supports semi-structured (XML) and relational schemas. It provides a library of match algorithms and allows various ways for combining match results. This library can be extended with new match algorithms to be used in combination with other matchers. The combination of strategies addresses different aspects of match processing, such as, aggregation of matcher-specific results and match candidate selection. In COMA, schemas are converted into rooted directed acyclic graphs [Do *et al.*, 2002] and used as the input of the match algorithms. The complete path from the root of the schema graph to the corresponding node is the unique ID of each schema element. COMA++ [Aumueller *et al.*, 2005] extends COMA by supporting both schemas and ontologies (written in OWL). Other improvements were also made such as adding a graphical user interface and new matchers ontology matching and reusing existing match results.

Some pre-match effort was needed in COMA to specify domain synonyms and abbreviations.

Evaluation: 5 XML schemas (for purchase orders taken from `biztalk.org`) were used to evaluate COMA [Do and Rahm, 2002]. Their size ranged from 40 to 145 unique elements (paths). Ten match tasks were defined, each matching two different schemas. To provide a basis for evaluating the quality of different automatic match strategies, the authors first manually performed the match tasks.

The evaluation of COMA went through 12,000 test series in order to evaluate the system using different choices of matchers and strategies. Each test series constituted of 10 experiments (predefined match tasks). The best combinations of parameters were identified based on their precision and recall across the series. The best F1 score achieved by COMA was 0.91 (average precision 0.93, average recall 0.89). Some pre-match effort was needed in COMA to specify domain synonyms and abbreviations.

4.2.3 Harmony

Harmony is a match module proposed as part of the Open Information Integration project (OpenII) [Seligman *et al.*, 2010]. It takes as an input both structured and semi-structured data sources including XML, relational, and ontology-based data sources. As described in [Bellahsene *et al.*, 2011], this approach utilises external dictionaries in the matching process which indicates that the update of Harmony is related to the freshness and ability of these dictionaries to be extended automatically or semi automatically.

Evaluation: a clear evaluation of the effectiveness of this approach has not been yet presented [Rahm, 2011].

4.2.4 MapForce

MapForce [Force, 2014] is a graphical schema matching and mapping tool which supports XML, database, flat file, EDI[34], and Microsoft Excel. MapForce has a data mapping environment which can load source and target schemas so that the user can easily map the schemas using functions and features provided by the tool. It tries to achieve higher efficiency through a graphical user interface [Rathinasamy, 2011]. This enables mapping functionalities like child elements mapping, functional libraries, and filters. A drag and drop mapping system makes the mapping easy from source to target schemas. This mapping process can be exported to XSLT transformations. The user in MapForce, for instance, has to draw correspondences between two schemas in order for the data integration queries to be generated automatically.

Evaluation: MapForce is a commercial tool so it has not been qualitatively evaluated as part of a research project.

4.3 Analysis and Discussion of Schema Matching Approaches

A common aspect of the systems and approaches discussed in the previous section is that they need a human action in order to function or to apply settings to accommodate changes in circumstances or in the considered data sources. This is infeasible in practice when paradigms like Linked Data are amongst the sources, as its main characteristic is it can change and extend very rapidly. The problem of human interference, or the degree of automation, in adapting the schema matching systems to current or likely future changes has not been the main priority in the approaches explored.

Unlike the OAEI instance and ontology matching techniques [Jimenez-Ruiz, 2017] (see Sections 4.4.1,4.4.2 and 4.4.3), where reference alignments are provided to allow a clearer comparison, evaluating schema matching and mapping techniques targeting different data models against existing tools and approaches has always been a difficult task [Bellahsene *et al.*, 2011]. It is challenging ”to make data of different types of benchmarks comparable with each

[34] Electronic data interchange

other" due to the lack of a common description or a parameter that can be measured upon [Pfaff and Krcmar, 2014, p. 1]. The comparison is frequently carried out theoretically taking into account approaches that have not necessarily presented explicit results. Well-established measures for qualitative evaluation are needed to evaluate the precision, recall and scalability, as well as identifying the best suited threshold value for semantic distance comparisons. As far as explored, there are no implemented schema matching approaches targeting semi-structured and Linked Data which also address the problem of sustaining and accommodating the inevitable continuous changes in Linked Data sources.

4.4 Data Interlinking

Many data interlinking approaches have been proposed throughout the years either independently or as part of the yearly OAEI event. This section presents a review of the most popular and more related solutions.

4.4.1 SERIMI

SERIMI [Araujo *et al.*, 2011] was the second best system at the OAEI Instance Matching 2011 [Nguyen *et al.*, 2012a]. It does not require any ontology alignment upfront or prior knowledge of the data or the schema. It is the tool that the interlinking approach proposed in this thesis based on (see Section 6.3). It consists of two phases: the selection phase and the disambiguation phase.

The selection phase is based on what their authors Araujo *et al.* [2011] described as existing traditional information retrieval and string matching algorithms. More specifically, it begins by extracting the entity label properties of the source dataset, which are the properties describing the labels that most represent the resource being interlinked (all RDF predicates that have a literal value with less than 200 characters). Only discriminative predicates with a higher entropy than a certain threshold are considered. Then, the labels of these properties are utilised to search for resource candidates with the same or similar labels. The results are resource candidates called a pseudo-homonym set. The entity label properties of each of these resource candidates are extracted in the same way as was done to the source dataset. The entity labels of the common property between the source and target entity label properties is then normalised, tokenised and compared using RWSA [Branting, 2003] algorithm. The resources with a similarity score below 70% are discarded from the pseudo-homonym set.

Having a set of pseudo-homonym for each source resource, the disambiguation phase then takes place to filter out false positive matches from true positive matches. They define false positive matches as resources in which their instances share the same label but belong to different classes, for example: Algiers can be a street, hotel, or a city. This problem is addressed in SERIMI via a model called Resource Description Similarity (RDS). RDS identifies the class of interest by finding the set of resources that are the most similar among pseudo-homonym sets.

The limitation of SERIMI is that it is restricted to only a single [few] properties for the matching [Nentwig *et al.*, 2017]. Additionally, the similarity threshold and other parameters have to be specified manually.

4.4.2 SLINT

SLINT is a domain independent Linked Data interlinking system. It uses coverage and discriminability to select the important predicates. Then, these predicates are aligned based on their confidence. The confidence in this context is high when corresponding predicates describe instances sharing the same type and characteristics. Using a three steps process (indexing, accumulating and candidate selection) SLINT generates the pair of instances with a high possibility to be homogeneous. The score of the instances of the generated candidates is calculated taking into account the confidence of their predicates and their similarities. The similarity is calculated differently according to the data type. For objects of type date, for instance, the similarity is 1 if the two values are equal and 0 otherwise. For strings type objects and URIs, they utilise TF-IDF[35] which gives advantage to instances sharing more common tokens.

SLINT was published as part of the yearly ontology matching event in 2012 (OM-2012).

[35] Term Frequency-Inverse Document Frequency

Other versions and extensions were published since then, including SLINT+ [Nguyen and Ichise, 2013], ScSLINT [Nguyen and Ichise, 2015]. SLINT+ presented the same principles of SLINT applied on OAEI 2013 benchmarks. ScSLINT identifies the lack of scalability of its predecessors and tried to address the performance. ScSLINT, however, does not consider balancing performance with precision or recall. Nguyen and Ichise [2015] (the authors who proposed ScSLINT) also described the use of weighted matrix structure in computing the similarity in candidate generation stage of SLINT+ (and SLINT) as not scalable and inaccurate on ambiguous data. The main features presented in ScSLINT and later versions have been: i) to normalise the data format in calculating the confidence, ii) to consider only target properties' objects that overlap the objects of its source counterpart property, and iii) to enable the user to install new similarity measures. These modifications naturally enhance the performance relative to previous releases, but they are also expected to impact other non-performance measures (such as precision and recall), something that is not elaborated in ScSLINT.

4.4.3 RiMOM

Risk Minimization based Ontology Mapping (RiMOM) [Zhang *et al.*, 2016] was first developed in 2006 [Li *et al.*, 2006a] and was originally a multi-strategy ontology matching and property matching approach. It is based on the combination of three lexical strategies being EditDistance, Vector-Distance and WordNet [Niu *et al.*, 2011]. An adaptive variation of similarity flooding is also used with the structural matching.

In 2010, RiMOM focus shifted, to some extent, to instance matching. As described in Shvaiko *et al.* [2010], their approach consists of four stages being: Preprocessing, Information Complementation, Matching and Spread Similarity, which respectively aim for:

- classifying individuals by their classes;
- completing information of each individual;
- running the matching algorithm for each class respectively;
- computing the similarity of two candidates based on weight-mean of properties assigned with specified weights.

RiMOM2013 [Zheng *et al.*, 2013] is an extension of RiMOM. It was presented as part of the ontology matching annual event OM-2013. Generally, the new characteristics that have been contributed to this new version, in contrast to the 2010 version, were a new interface and control layers that allow the user to customise the matching procedure. This included selecting preferred components, setting the parameters for the system, choosing to use translator tool or not. For the instance matching track, particularly, a new algorithm inspired by Wang *et al.* [2012], called Link Flooding Algorithm, was used. It is constituted of three modules. The first module performs a simple pre-processing and normalisation of the data such as unifying the language and data format and removing special characters. The second module is described using examples, but it is mainly logical matching whereby the subjects are aligned. The third module is for objects alignment (another term of instance matching described in Section 3.7.2). A weighted average score of the similarity of the instances of specific properties is calculated and compared against a threshold to decide whether two instances are aligned. The similarity measure used is EditDistance [Navarro, 2001].

RiMOM is a popular tool that produced promising results as an ontology matching solution [Li *et al.*, 2006a]. As an instance matching approach, RiMOM similarity metrics has been evaluated in Rong *et al.* [2012] against existing learning approaches. The results suggested that the combination of the three strategies is not accurate enough for instance matching. RiMOM2013 showed good results, but it targeted specific properties (`comments`, `mottos`, `birthDates` and `almaMaters`) that were rather tailored for the addressed benchmark of the OM-2013[36] event.

4.4.4 SILK

The Link Discovery Framework (SILK) [Volz *et al.*, 2009] is a link discovery system that supports a data publisher in setting explicit links between two datasets. It has its own declarative language Silk - Link Specification Language (Silk-LSL) that data publishers can use to choose which types of RDF links ought to be discovered between data sources and which conditions data items must fulfil in order to be interlinked. Various similarity metrics can be applied to these link conditions as well as taking into account the graph around the data item using an

[36] http://om2013.ontologymatching.org/

RDF path language.

Four main advantages the authors of SILK outlined: i) the flexibility that Silk-LSL offer in defining link conditions; ii) generating not only identity links, but other types of RDF links; iii) the ability to be applied in distributed environments without replicating the data locally; iv) the implementation of multiple caching, indexing and pre-selection to improve the performance.

On the other hand, SILK has not been evaluated and tested in the same way as existing data interlinking approaches have been. No benchmark was used and the primary focus was the number of the links that can be discovered. The precision and the recall have not been considered. This is may be due to two reasons. SILK was published in 2009, when there were not many systems to compare against and OAEI had just started publishing benchmarks, for instance matching, in the same year. The second reason is that SILK is a link discovery system; therefore, other RDF links can also be discovered which makes it challenging to evaluate the same way as identity links (interlinking) approaches. It is used to assist with linking data with existing resources in the Web of Data. Although the evaluation is not as revealing as it can be, Nguyen and Ichise [2015] stated that the limitation of SILK can be seen when addressing a large-scale dataset.

4.4.5 LIMES

LIMES [Ngomo and Auer, 2011] is a link discovery tool that, similarly to ScSLINT, focuses on improving the processing time when mapping large knowledge bases. It views the problem of data interlinking from a metric space perspective. It uses mathematical characteristics, such as triangle inequality, to compute pessimistic approximations of distances and to estimate the similarity between instances [Symeonidou, 2014]. Based on these approximations, LIMES find and exclude a large number of computations without losing links.

LIMES showed more efficiency in terms of time-consuming than SILK [Rajabi *et al.*, 2015]. Similarly, to many record linkage and link discovery tools, it concentrates much of its efforts on filtering out non-matches before going through the more time-consuming comparisons.

4.5 Analysis and Discussion of Data Interlinking Approaches

The one noticeable aspect in the data interlinking approaches, including some that were not mentioned in the previous section, is that they are multidisciplinary. Many techniques originated from different computing and mathematics subfields are employed or combined to improve the results, including: Semantic Web, Data Mining, Geometry, Probabilities, etc. In terms of modules constituting the approaches, the most common stage in most of the interlinking systems is the pre-processing. This suggests that the pre-processing is necessary before the heavy processing of data interlinking.

The other point that can be drawn is that most of the solutions explored have not addressed balancing between performance and precision. Balancing between performance and precision indicates that the solution ought to aim at or attempt an acceptable precision and recall of a relatively large-scale dataset in a reasonable time. The reason for this can be linked to the nature of the instance matching organised challenges, such as the OEAI yearly event, where the comparison is centred mainly on precision and recall but not performance. The other argument or observation that can be stated is that the application of data interlinking is not always time-critical as in other fields, such as data retrieval.

Although OAEI provides reference benchmarks from a variety of sources, the available evaluation data is still insufficient to recognise and compare the performance of the approaches [Ferrara *et al.*, 2008]. This statement was made in 2008; yet, it can be still considered as accurate to some extent, as not too many changes have been introduced since then.

All the approaches discussed, except SILK, and seen to date but non-mentioned interlinking approaches, were designed to discover identity and/or other links on existing published data. Going back to the point made in the previous paragraph about performance, it would be theoretically better to find the links in the publishing stage. Meaning that the tool should automatically interlink the data being published with its existing counterpart in the Web of Linked Data.

4.6 Data Integration

The data integration approaches reviewed in this section have been chosen on the basis that: i) they can be compared against the integration approach proposed in this thesis (see Chapter 7); ii) consider semi-structured and Linked Data as input models, or; iii) those that the author utilised some of their components.

4.6.1 FuhSen

FuhSen [Collarana *et al.*, 2016] is a keyword search platform that integrates heterogeneous data sources. It is a usable and an adaptable solution within its scope of interest, which is crime data information. As part of the integration process, the application uses many components including their own vocabulary named ”OntoFuhSen”. The vocabulary is used as an exchange and an intermediate language between the other parts of the system. In its current status, it was tailored to accommodate information about their targeted data, being information about a person, a product or an organisation. In spite of the fact that the vocabulary can be extended, its position suggests that many other changes would also need to be made. Wrappers (adapters) are one of the components connected to the vocabulary. They are designed to extract data from the source outputs. In FuhSen, every data source is associated with a collection of wrappers.

4.6.2 PowerAqua

PowerAqua [Lopez *et al.*, 2012] is one instance of QA[37] solutions, evolved from AquaLog [Lopez *et al.*, 2007], that proposes the use of multiple ontologies that will be selected according the user’s query. It is a concept that will be utilised in the proposed data integration approach because, as stated by the authors, it is not possible to select in advance which of the vocabularies or ontologies will be needed to answer a query.

[37] Question Answering

4.6.3 MOMIS

In Vincini *et al.* [2013], the authors described the Mediator/Wrapper based architecture for integrating semi-structured and structured data. Mediator envirOnment for Multiple Information Sources (MOMIS) uses its own description, definition language and thesaurus for extracting, defining and storing the information inputted and retrieved from the sources. In their semi-automatic methodology, they follow Global as View (GaV) paradigm (see Section 3.4.3) to express the global schema following the view of local schemas.

4.6.4 SWIM

Semantic Web Integration Middleware (SWIM) uses query mediation and provides tools to ”view data as virtual RDF” [Koffina *et al.*, 2006]. The middleware publishes, or re-publish, XML and Relational databases (RDB) as RDF. Resource Query Language (RQL) queries which are then composed and optimised according to the RDF views and the mappings constructed.

4.6.5 LSM

The layer architecture in Le-Phuoc *et al.* [2012] illustrates Linked Stream Middleware (LSM), a middleware system to integrate time-dependent data, or sensor data, with the Linked Data cloud. It unifies and publishes stream raw data, coming from different sources, as Linked stream data before finally executing SPARQL queries over them. The system uses wrappers in the data acquisition layer to collect the data from different sensor devices and publishes them into a unified format. Having the data stored in a Linked Data layer, it will then be accessed via two types of query engines: a standard SPARQL query processor and Continuous Query Evaluation over Linked Streams (CQELS) engine processor.

4.7 Analysis and Discussion of Data Integration Approaches

It can be clearly seen from the schema matching and data integration approaches review (see Section 4.6 and 4.6) that the problem of reconciling heterogeneity affects many research areas and systems. One of the major aspects distinguishing one category from another is their interpretation of heterogeneity or the part thereof they are concentrated on. The other point is the characteristics and the context of the data. For instance, Smart Cities' integration systems or modules receive stream data originated from sensors; thus; they demand more ability to maintain data freshness [Kettouch *et al.*, 2017b, 2016]. Whereas search systems' needs, for example, are different. They are more interested in retrieving accurate results in a reasonable time from the largest, but relevant, number of sources and display them in an interpretable form.

The idea presented in the previous paragraph suggests that there is not one optimal solution for the entire data integration problem. Still, this does not mean that there cannot be an effective solution for particular data models in a particular context. The two previous statements are one of the primary reasons why schema matching is addressed separately from its super class (data integration) in this thesis. Schema matching reconciles specifically structural heterogeneity without paying attention to the other differences, such the access method for example. It is a common and a fundamental part of most of data integration and exchange systems [Bellahsene *et al.*, 2011; He and Chang, 2003]. It can also be argued it is the most context-free module as it only matches between structures regardless of where it is applied. It aims at achieving generic challenges, such as: adaptivity, performance, data freshness and precision, so the module suits nearly all types of applications.

4.8 Summary

The analysis of related works have led the author to conclude that there is a need for new approaches for data integration and interlinking. This chapter also found that the explored schema matching techniques are not adapted for the data input and the requirements, such as: degree of automation and adaptivity to changes, that this thesis focuses on. On the other hand, the author has also identified several features that the proposed system can incorporate or base on in addressing the research questions and designing the contributed solutions, such as: the knowledge base in-dependency in SERIMI [Araujo *et al.*, 2011] and the selection of an ontology on-the-fly depending on the user's query presented in PowerAqua. This chapter has also identified features that can be used as evaluation criteria for the proposed system (see Sections 4.3, 4.5 and 4.7).

The next three chapters present the new approaches proposed to fill the knowledge gap identified in this thesis, along with addressing the limitations discussed in this chapter and the criteria extracted in Chapters 2 and 3. The next chapter introduces the new schema matching approach and explains how it copes with the data freshness requirements and continuous changes in the Linked Data space.

Chapter 5

SimiMatch: Schema Matching for Semi-structured and Linked Data

Intelligence is the ability to adapt to change.

Stephen Hawking

5.1 Introduction

Schema Matching is at the core of many Web semantics, database systems and topics, such as data integration, data warehouse and semantic query processing. It is frequently considered as the most challenging and decisive stage. Today, not only does schema matching have to deal with ”static” heterogeneity, but it also needs to accommodate continually changing data content, structure and organisation. This is something that legacy systems have also experienced, but not at the current scale.

The Linked Data paradigm is a common standard, implemented through a set of recommendations, that complement the general architecture of the Semantic Web. The aim was to create a single space containing data, that is machine-readable and connected to related data whether from the same or other external sources. However, figures show (see **MQ1** in Section 1.2) that one of the main sources of semi-structured data providers, Web APIs, continued to grow even after the creation of the Linked Data concept [Abawajy, 2015]. Given that data sources with significant value are still in a semi-structured format, it is essential to bridge between the two data models, so that the full potential of the Semantic Web can be realised.

The reduction of barriers in sharing knowledge over the Web via Linked Data gave more flexibility in contributing to this data space. Consequently, data repositories appertaining to this paradigm are changing and expanding very rapidly. Linked Data aims to provide links between different data sources in order to create a single global data space, ”the Web of Data”.

The appearance of new data spaces along with the growing amount of semi-structured data suggests that more research on schema matching is needed. In addition, since Linked Data is expected to change over time using different vocabularies, the schema matching as part of it ought to be able to accommodate these changes.

This chapter presents an approach for schema matching that takes as input both semi-structured and Linked Data. The new approach proposed in this thesis is called SimiMatch Kettouch *et al.* [2017a, in press] and it is based on the observation made by the author in [Kettouch *et al.*, 2015a] that datasets in the same domain may not be syntactically identical but the semantics of their properties may overlap. The contribution lies in exploiting automatically this overlap by mapping between the elements of the sources and global schema. The process involves extracting the semantically distinct properties from the sources, regardless of their model or namespace, and transferring the output into the virtual view. This process is repeated, incrementally building up a virtual view, ignoring duplicate information, to create a unique set of semantically distinct properties from all the sources. As shown in Figure 5.1, SimiMatch is a module in both the data interlinking and integration approaches proposed in this thesis.

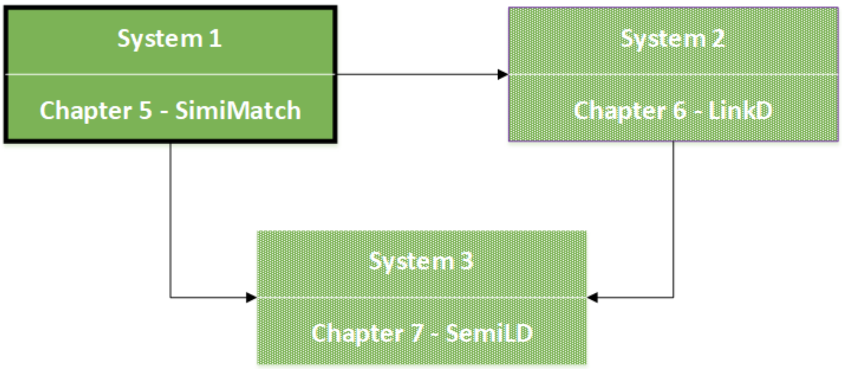


Figure 5.1: Relation of SimiMatch approach with reference to other systems of this thesis [Author, 2017].

5.2 The New SimiMatch Approach

SimiMatch is an element-based schema matching approach that targets two data models, the semi-structured (hierarchical) model and Linked Data (graph) model. It is designed to be adapted and employed as a module, that reconciles the structural heterogeneity in the proposed integration approach SemiLD (see Chapter 7) and the interlinking approach LinkD (see Chapter 6).

SimiMatch preserves the autonomy of the participant source. The matching is performed without converting the data model or migrating it to another data space. SimiMatch is a domain-dependent solution. Thus all the sources need to belong to the same domain for its full potential to be achieved. Its focus is on the matching operation. No automatic recognition of the domain is involved. In cross-domain Linked Data sources (DBpedia for example), the domain is extracted from the value of the property rdf:type. For semi-structured data sources, the Web APIs are initially categorised manually to their domain.

SimiMatch does not utilise any reference, such as a knowledge base or an ontology, in generating the mapping rules. As a result, it has the ability to process large-scale sources. The result is a global schema that is a virtual view: a single and uniform temporary storage able to accommodate the data coming from two, or more, data sources. Its creation and update is automatic and unsupervised; hence, it does not require any manual interference. Consequently, the approach can adapts any future changes from these dynamic data sources.

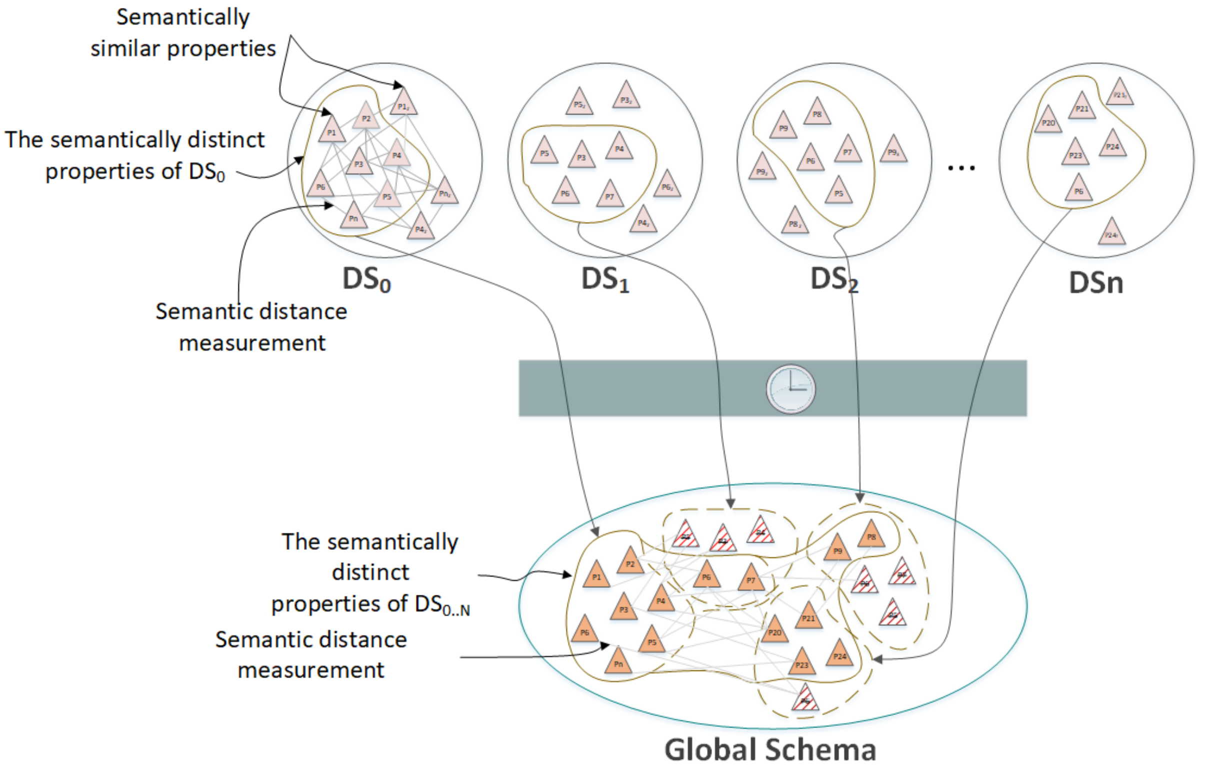

Semantically similar properties
The semantically distinct properties of DS$_0$
Semantic distance measurement
DS$_0$
DS$_1$
DS$_2$
...
DSn
The semantically distinct properties of DS$_{0..N}$
Semantic distance measurement
Global Schema

Figure 5.2: Description of the creation and update of the global schema in SimiMatch [Author, 2017].

Although it is technically more precise to refer to SimiMatch as a schema integration approach since the output is a global schema (see Section 3.2.1), it is presented and investigated in this thesis as schema matching problem as the main task and the focus is in matching the elements (within the sources and between the sources and the global schema).

The overall functionality of SimiMatch is described in Figure 5.2. It can be seen that it does not differentiate between semi-structured and Linked Data sources. Both the elements of semi-structured data and the properties of Linked Data are represented as triangles. This shows that once the properties labels are extracted and pre-processed, they will be treated in the same way regardless of their origin or type of data model

The clock sign in the middle shows that the same process is repeated on a time-lapse basis to ensure the global schema is kept up to date. This is achievable due to the high degree of automation of the process. The approach is based on properties (elements) matching, which are the tags or attributes for semi-structured data and the predicates for Linked Data. The global schema is created when the system is first run. To keep the global schema updated, the system verifies periodically whether a new source has been added or the structures of existing sources have been modified or extended.

Before the process described in the diagram takes place, the properties go through a pre-processing stage whereby the labels are cleaned and prepared to be compared semantically. To achieve this, the last part of the URI of the predicate for Linked Data is extracted, and numbers and some special characters, including commas and underlines, are replaced with null.

For example, if the property retrieved is:

`http://schema.org/releaseDate`

The result after the pre-processing is: `release data`

5.2.1 The Extraction of the Semantically Distinct Properties of the Sources Schema

First, the set of the semantically distinct properties of each of the sources is extracted locally. This is achieved by calculating the semantic difference between the properties in each source. As Figure 5.3 shows, where there are two properties ($P1$ and $P1_2$), or more, sharing the same meaning, only the first one ($P1$) is transferred to the result set and the rest are discarded.

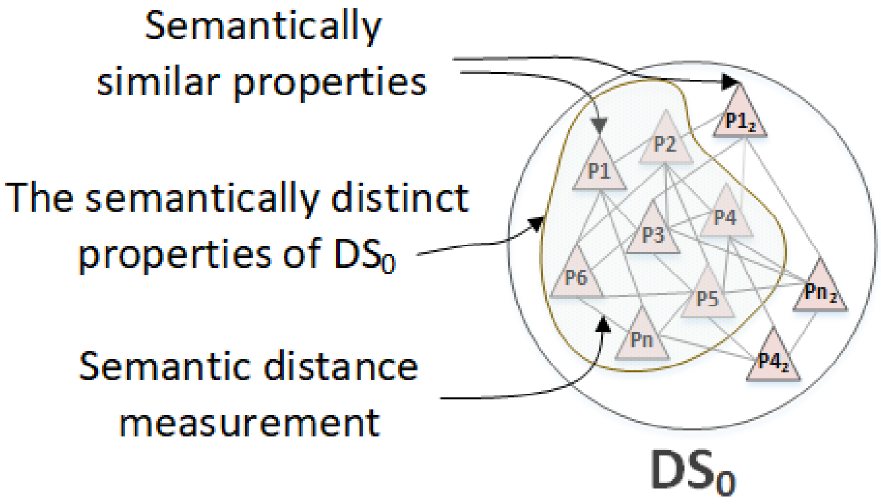

Semantically similar properties
The semantically distinct properties of DS_0
Semantic distance measurement
P1
P2
$P1_2$
P3
P4
P6
P5
Pn_2
Pn
$P4_2$
DS_0

Figure 5.3: The extraction of the semantically distinct properties (extracted from Figure 5.2).

The sets of the semantically distinct properties are illustrated in Figure 5.3 by the bubbles inside the circles of the data sources DS_0, DS_1 ...DS_n.

The retrieval of the properties differs between the semi-structured and Linked Data. For semi-structured data, originated from Web APIs, the properties are extracted by processing one result, since all datasets share the same properties. On the other hand, not all the datasets in a particular Linked Data namespace share the same structure. Therefore, the properties are extracted through a separate process that takes into consideration many Linked Data datasets. Various vocabularies are utilised to describe Linked Data. Yet, many of these vocabularies, particularly when they are related to the same domain, have overlapping semantics [Kettouch *et al.*, 2017a]. Hence, a representative sample is sufficient to retrieve all semantics of the properties.

More formally, a data source (DS) consists of a set of n properties (P), as follows:

Definition 5.1 (Data source)**.**

$$DS = \{P_x \mid x = \overline{1,n}\}$$

For every data source, the semantically distinct properties are extracted as follows:

Definition 5.2 (Semantic Distinction)**.**

$$SemD(DS) = \{P\ in\ DS | sDist^n_{x=1}(P,P_x) < Th\ \ where\ P_x \in DS\}$$

$$P = \begin{cases} pr\ in\ (s,pr,o) \in LD\ where \begin{cases} s = subject \\ pr = predicate \\ o = object \end{cases} \\ \\ t\ in\ (r,t,v) \in SsD\ where \begin{cases} r = root \\ t = tag/attribute \\ v = value \end{cases} \end{cases}$$

LD and *SsD* refer to Linked Data and Semi-structured Data respectively. $sDist(P,P_x)$ denotes the semantic distance between P and P_x.

The set of the semantically distinct properties of the local schema DS consists of all those properties with a unique meaning. In other words, each property is compared with all the other properties from the same DS. If the semantic distance ($sDist$) between them is less than a threshold (Th) then the property goes into the set of the semantically distinct properties.

The properties of DS can be originated either from a Linked Data model, where the property is the predicate of the triple, or a semi-structured model, where the property is a tag or an attribute.

The semantic text similarity system used in this approach is UMBC EBIQUITY-COR [Han *et al.*, 2013]. The UMBC tool is constructed by combining Latent Semantic Analysis (LSA) word similarity and WorldNet knowledge. Other semantic similarity algorithms were explored, such as [Li *et al.*, 2006b], but UMBC is used because it concentrates on the semantics of the word but not its lexical category, which fulfils the requirements of the semantic similarity tool adapted to SimiMatch, since the available vocabularies for describing vary between nouns and verbs (see Section 3.7.4.2).

5.2.2 The Creation of the Global Schema

Once at least two semantically distinct property sets are extracted from two DS, the creation of the global schema starts to take place. The creation of the global schema is incrementally processed each time a new property set is extracted from another DS. In the creation of the global schema all the semantically distinct properties of all the sources that are considered are extracted. This forces a semantic overlap between the global schema and all the sources considered. It is designed to make the properties of every source semantically a subset of the properties of global schema. The following formula describes the creation of the global schema (GS) and is based on the semantic distinction ($semD$) formula previously described. At the time when the first data source DS_0 considered the GS would be empty, therefore, all the semantically distinct properties are transferred directly as there is nothing to compare against. The system then goes on to incrementally extract the semantically distinct properties from the other sources, which are transferred to the global schema.

Definition 5.3 (Global Schema)**.**

$$GS = \begin{cases} semD(DS_0) \\ \\ \bigcup_{i=1}^{n} \{P_j \in semD(DS_i) \mid sDist(P, P_j) < Th, \\ \qquad\qquad \forall P \in GS,\ j = \overline{0, size(DS_i)}\} \end{cases}$$

Algorithm 5.1 SemanticDistinction [Author, 2017]

```
Input:  set1, set2: PropertiesSets
        threshold
Output: P: PropertiesSet

get_index(set1, set2)
sizeSet1 = size (set1)
sizeSet2 = size (set2)

for $i$=0 And $i$ < sizeSet1 do
    for $j$=0 And $j$ < sizeSet2 do
        if check_if_not_indexed(set1[$i$], set2[$j$]) then
            distance = SemanticDistance (set1[$i$], set2[$j$]);
        else if distance > threshold then
            P.addAttribute(set1[$i$])
            Insert_index(set1[$i$], set2[$j$])
        else
            $i$++; $j$++;
        end if
    end for
end for

return P;
```

Algorithm 5.1 shows how the semantic matching of the properties are being indexed. Note that not only the matchings between the data sources and the global schema are indexed, but also internally within each of the Linked Data sources. `Set1` and `set2` in an internal extraction of the semantically distinct properties are, respectively, the new properties retrieved (after a period of time) and the current set of semantically distinct properties. It is not applied to semi-structured sources as they do not share the same characteristic of Linked Data, which is the continuous dynamism and changes in vocabularies and structures describing their datasets. For external semantic distinct between a source and the global schema, `set1` and `set2` represents the set of semantically distinct properties of each of them. The order is irrelevant when inserting or accessing indexes in this approach.

Finally, the same process is repeated but this time for the matching of instances rather than the creation of a global view. The only difference between the creation of the global schema and the matching of the results is a further stage, which is the transfer of the values of the sources properties to the global schema properties. The values of the elements do not affect the operation as it has no role in the matching. Therefore, it is theoretically valid for the outputs of a SPARQL query or an HTTP request to be matched with the created global schema if the semantic distance measurement is used, as every source is a semantic subset.

5.2.3 Indexing the Global Schema

A record of the global schemas previously created is saved containing the time stamp, results were required for its creation, and the data sources considered. This allows the system to verify whether a global schema conforming to the user's queries is available.

Algorithm 5.2 GlobalSchemaIndex [Author, 2017]

Input: set1: SourcesQueried

```
1: for $i$=0 And $i <$ GS_Index.size do

2:     if ( ! GS_Index.get(SourcesList).containsAll(set1)) then
3:         Create(GlobalSchema, SourcesList)
4:         Update(GS_Index, SourcesList)

5:     else if ( Now() $-$ GS_Index.get(set1).CreationDate $>$ UPDATE_RATE) then
6:         Update(GlobalSchema)
7:         Update(GS_Index)
8:     end if

9: end for
```

The input for Algorithm 5.2 is the list of the sources queried by the user. The other input is the current time which is retrieved from the system. In Line 1 the algorithm iterates through the records of the global schemas previously created. Then Line 2 checks whether there is no global schema stored locally that contains all the sources that user queried. If that is the case, SimiMatch goes on creating a global schema for these sources following the same steps described in the previous sections of this chapter. It also updates the records and adds metadata about the newly created global schema.

Lines 5-8 verify whether the global schema that created is still up to date. Finding the number of the days (or months) needed to update the global schema (`UPDATE_RATE`) is not part of the scope of this research. It can be the focus of another study that investigates the average rate of changes in Linked Data sources. The aim of SimiMatch is provide the ability to accommodate these changes. In the experiments carried out as part of this thesis, the update rate is set at one week (based on observations and estimations). The update can be launched in the implementation by either the developer or the user.

If the global schema containing the sources that user queried is not up to date, SimiMatch re-creates the global schema and transfers the new properties, that are not present in the previous version, to the global schema and updates the time stamp.

To help evaluate this process, the semantic similarity threshold and the number of results are saved. This records the effect of different parameters on the efficiency of the approach; and therefore, to identify the most accurate combination.

5.3 Implementation and Evaluation of SimiMatch

This section presents the implementation and proposes an evaluation of SimiMatch. The flowcharts in Figures 5.4 and 5.5 describe the overall functionality of SimiMatch. Figure 5.4 illustrates the different stages the approach goes through to match two schemas and update the global schema. Figure 5.5 highlights the concept of automatically re-running the schema matching process according to a time-lapse to check for any changes in the schemas.

SimiMatch and the semantic similarity tool are implemented using Java. The same programming language is used in the preparation of the inputs along with other tools, such as Jena framework, SPARQL, XML, JSON and various parsers libraries. The XML format has been used for implementation to represent the global schema, as it is an effective and a pop-

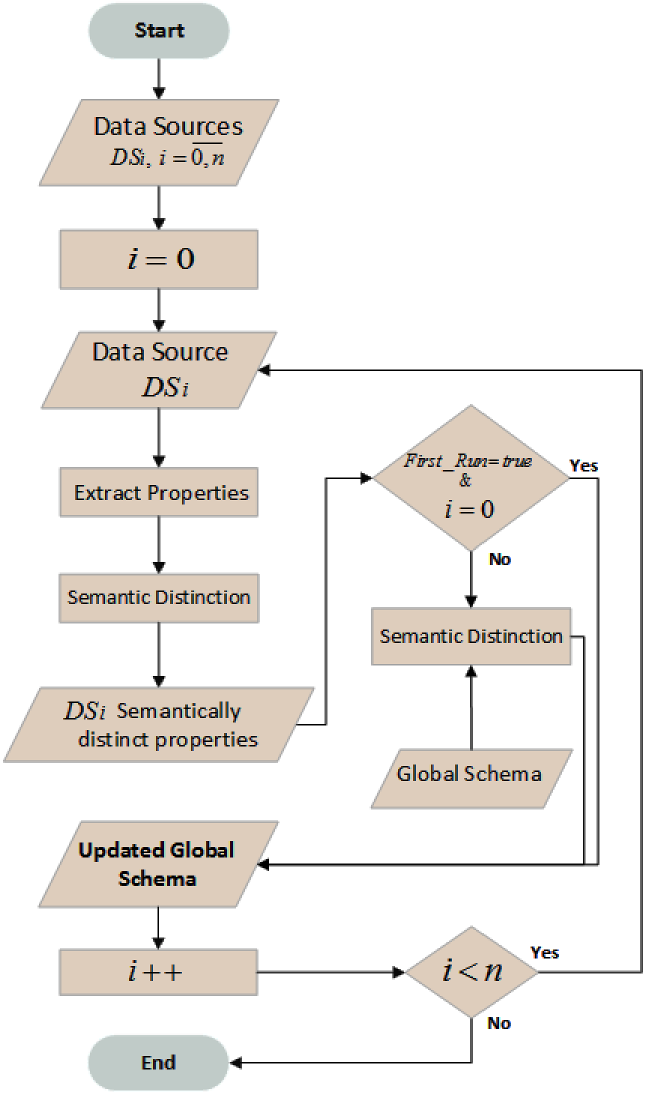


Figure 5.4: Flowchart describing the implementation of SimiMatch [Author, 2017].

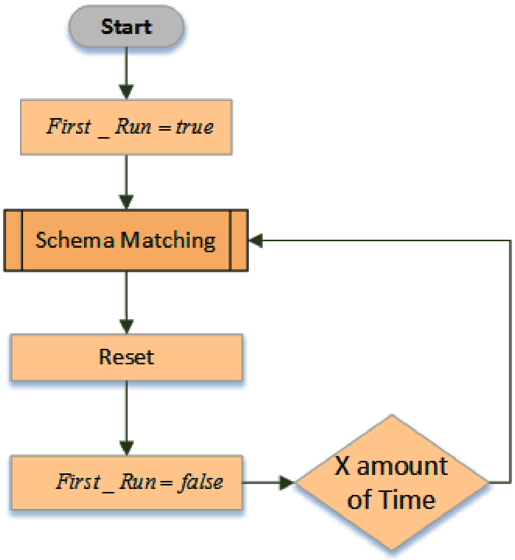


Figure 5.5: Flowchart describes the re-running of SimiMatch according to a timelapse [Author, 2017].

ular mean of information exchange (see Section 2.5.2). The UMBC semantic similarity tool is re-implemented locally to eliminate the time penalty of connecting to the API every time a semantic distance is calculated.

Note that Linked Data input files used for this evaluation were produced in February 2016. As explained previously in this thesis, Linked Data sources are changing quickly over time. For example, in DBpedia version 2016-45, triples are filtered from the Raw Infobox Extractor and some properties will not be loaded on the public endpoint. Thus, running the system at a different time can require a different method to prepare the input and may display different results but the trend will be maintained. The public services (SPARQL endpoints) of Linked Data sources frequently apply resource limits and they are occasionally unavailable (see Section 2.4.3). Therefore, RDF dumps were downloaded locally in HDT[38] format to avoid this limitation. HDT (Header, Dictionary, Triples) is a compact data structure and binary serialisation format for RDF that keeps big datasets compressed to save space while maintaining search and browse operations without prior decompression.

The system is tested in three domains that have available and accessible data sources in both data models (semi-structured and Linked Data):

- **Movies:** data were retrieved from four data sources, of which two are Linked Data: DBpedia and LinkedMDB, and two semi-structured: The Open Movie Database[39] (OMDb), The Movie Database[40] (TMDB).
- **Location data:** three Linked Data sources were considered: DBpedia, LinkedGeoData[41] and Geonames and one semi-structured: GoogleMapsAPI[42].
- **People's data:** this category involves data sources that provides information only about public figures (celebrities, writers, actors, producers, politicians, singers, etc.). Two Linked Data sources were included: DBpedia and LinkedMDB, and two semi-structured:

[38] Header, Dictionary, Triples
[39] http://omdbapi.com
[40] http://themoviedb.org/documentation/api
[41] http://linkedgeodata.org/About
[42] http://developers.google.com/maps

IMDB[43] (the Internet Movie Database) and Last.fm[44]

The choices of data sources in each of domain is a combination between specialised data repositories, such as: LinkedMDB (for movies related data) or Last.fm (for music related data) and general (cross-domains) repositories, such as DBpedia. This selection is to provide a representative testing datasets that cover the different scenarios that may occur.

5.3.1 Retrieving the Syntactically Distinct Properties

Before running the system, the inputs of the schema matching system are prepared by querying the considered data sources. The query retrieves the properties of N results (movies, locations and persons) of each of the data sources of the various domains. A common keyword (that occurs in many resources in the domain), such as ”the”, is used to retrieve a sufficient number of results to carry out the experiment. Only the syntactically distinct properties have been fetched.

Tables 5.1a, 5.1b and 5.1c show the number of the properties retrieved in each domain. The limit in the number of results is set at N=700 because presenting additional results does not have any effect on results in Tables 5.2a, 5.2b and 5.2c.

5.3.2 Internal Semantic Distinction and The Creation of the Global Schema

Comparing Table 5.1a with Table 5.2a, Table 5.1b with Table 5.2b and Table 5.1c with Table 5.2c show that the number of syntactically and semantically distinct properties of the semi-structured data sources is the same. This points out a major difference between Linked and semi-structured data. Although semi-structured data is created to be flexible in terms of changing the structure, the technology exploiting it, particularly in data exchange and RESTful services, frequently uses a fixed structure. Whereas with Linked Data sources, if the same tables are compared, there are different heterogeneous vocabularies that are continuously changing in describing datasets even in the same sources that belong to the same domain.

[43] http://imdb.com/help/show_leaf?about

[44] http://www.last.fm/api

N movies	Syntactically Distinct Properties				Total
	Linked Data Sources		Semi-structured Data sources		
	DBpedia	LinkedMDB	TMDB	OMDb	
100	103	36	20	13	172
200	131	41	20	13	205
300	157	43	20	13	233
400	171	47	20	13	251
500	179	47	20	13	259
600	184	48	20	13	265
700	186	48	20	13	267

(a) Movie data sources [Author, 2017].

N movies	Syntactically Distinct Properties				Total
	Linked Data Sources		Semi-structured Data sources		
	DBpedia	LinkedGeoData	Geonames	GoogleMapsAPI	
100	115	89	15	11	230
200	156	97	15	11	279
300	177	97	15	11	300
400	193	97	15	11	316
500	193	47	15	11	316
600	193	48	15	11	316
700	193	48	15	11	316

(b) Location data sources [Author, 2017].

N movies	Syntactically Distinct Properties				Total
	Linked Data Sources		Semi-structured Data sources		
	DBpedia	LinkedMDB	IMDB	Last.fm	
100	104	10	4	8	126
200	134	10	4	8	146
300	148	10	4	8	160
400	155	10	4	8	167
500	179	10	4	8	191
600	184	10	4	8	196
700	186	10	4	8	198

(c) People data sources [Author, 2017].

Table 5.1: The number of the syntactically distinct properties extracted per source [Author, 2017].

N movies	Semantically Distinct Properties				Total	Global schema
	Linked Data Sources		Semi-structured Data sources			
	DBpedia	LinkedMDB	TMDB	OMDb		
100	74	36	20	13	143	112
200	93	41	20	13	167	136
300	110	43	20	13	186	153
400	119	45	20	13	197	164
500	121	45	20	13	199	164
600	121	45	20	13	199	164
700	121	45	20	13	199	164

(a) Movie data sources [Author, 2017].

N movies	Semantically Distinct Properties				Total	Global schema
	Linked Data Sources		Semi-structured Data sources			
	DBpedia	LinkedGeoData	Geonames	GoogleMapsAPI		
100	83	82	12	11	188	159
200	109	88	12	11	220	190
300	125	88	12	11	236	206
400	136	88	12	11	247	215
500	136	88	12	11	247	215
600	136	88	12	11	247	215
700	136	88	12	11	247	215

(b) Location data sources [Author, 2017].

N movies	Semantically Distinct Properties				Total	Global schema
	Linked Data Sources		Semi-structured Data sources			
	DBpedia	LinkedMDB	IMDB	Last.fm		
100	84	10	4	8	106	97
200	106	10	4	8	128	106
300	110	10	4	8	132	112
400	114	10	4	8	136	116
500	117	10	4	8	139	122
600	120	10	4	8	142	122
700	120	10	4	8	142	122

(c) People data sources [Author, 2017].

Table 5.2: The number of the semantically distinct properties of the data sources and the global schema created by SimiMatch [Author, 2017].

Movies	Locations	People
release_Date, date subject, is_Primary_Topic_Of producer, executive_producer	population_Total, population operated_By, operator region, area	honours, awards name, given_Name influenced_By, influenced

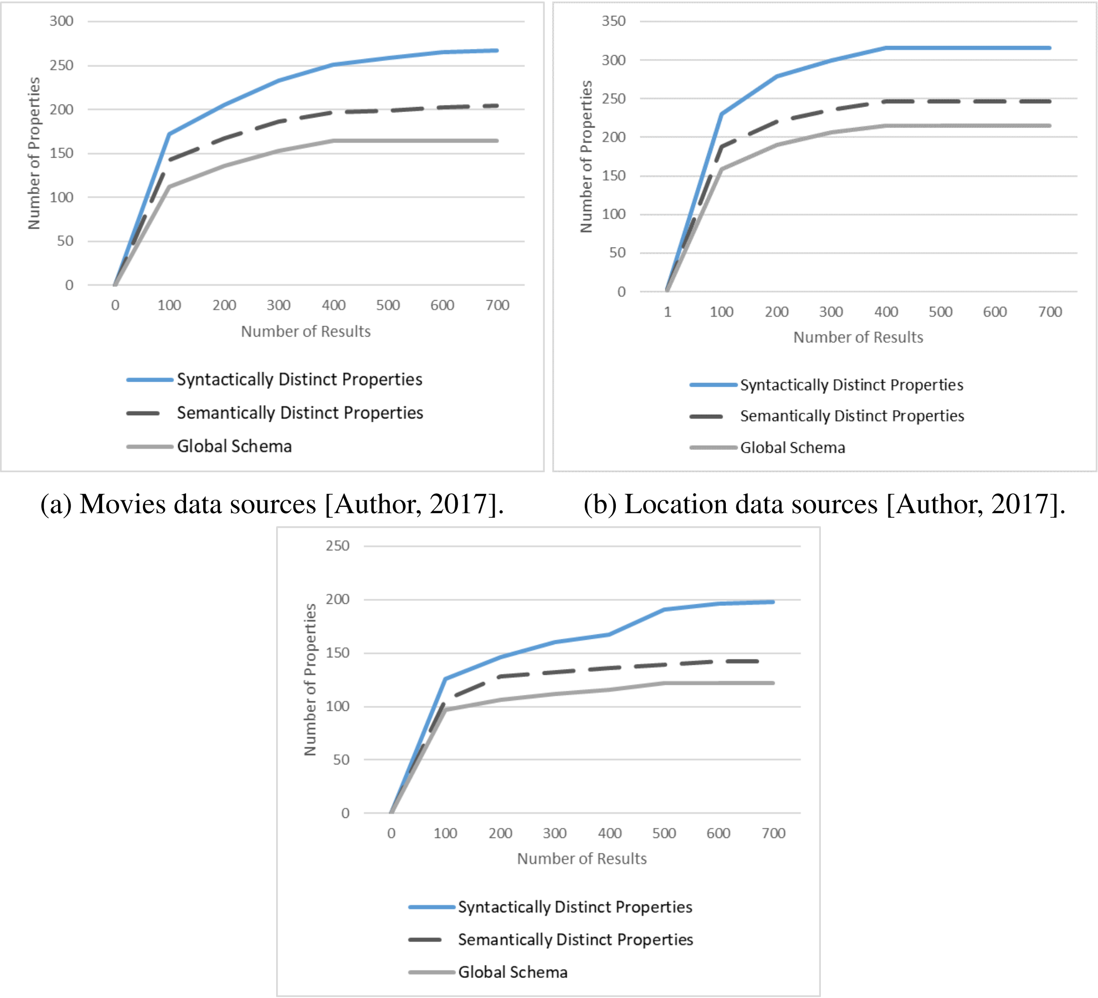
Table 5.3: Examples of semantically similar labels of properties [Author, 2017].

The changes in the column ”Total” in Tables 5.1a, 5.1b, and 5.1c compared to Tables 5.2a, 5.2b, and 5.2c respectively are the result of the first step of the approach presented, which is the extraction of the semantically distinct properties from each of the data sources. The global schema column demonstrates the second step, which is the incremental extraction of the semantically distinct properties between the new output of the first step and the current global schema.

Additionally, Table 5.2a highlights the difference between the degree of semantic heterogeneity within the vocabularies used in LinkedMDB and DBpedia. The increase in the semantically distinct properties which resulted from inputting more datasets to the system is more significant in DBpedia. LinkedMDB, however, showed a slight rise and stabilised at 400 datasets. Likewise, LinkedGeoData, in Table 5.2b, demonstrates roughly the same pattern. This contrast can be related to many reasons, including: the large scale of DBpedia compared to LinkedMDB or LinkedGeoData, or it can be more a cross-domains versus specialised data source issue. It is not, however, the focus of this thesis to confirm it.

Figure 5.6a, 5.6b and 5.6c are based on tables 5.2a, 5.2b and 6.5 respectively. They highlight the considerable gap between the numbers of the syntactically and semantically distinct properties according to the number of results retrieved in the movies, people and location data sources respectively. It shows the difference in the terminology of the vocabulary used to describe resources in the same domain, but it also points out to the redundancy and the overlap in the semantic of these terminologies. Table 5.3 contains examples of some semantically similar Linked Data properties found in the process in each of the domains:

Two additional noticeable aspects are demonstrated in these three figures. First, the number of properties in the global schema is not equal to the total number of semantically distinct properties of the sources. It highlights that some properties are discarded during the second step because their semantic information already exists in the global schema. Second, the number of



(a) Movies data sources [Author, 2017].

(b) Location data sources [Author, 2017].

(c) People's data sources [Author, 2017].

Figure 5.6: Comparison between the number of semantically and the syntactically distinct properties and the number of properties in their global schema [Author, 2017].

semantically distinct properties stabilises at N=500, which confirms, to some extent, that the semantic information of the vocabulary used to describe a resource in a specific domain is limited and considerably lower than its syntax.

5.3.3 Evaluation

Diverse methodologies are utilised to evaluate schema matching techniques and tools [Do *et al.*, 2002] due to the absence of a universal benchmark [Bernstein *et al.*, 2011; Pfaff and Krcmar, 2014]. The methodology used in this thesis is drawn from [Do *et al.*, 2002]. The basis for this qualitative evaluation is created by manually solving the match task. The manual match result is then considered as a ”gold standard” to assess the quality of the result of the proposed tool. This allows the calculation of the precision and recall in the same way as described in Section 3.2.2.

Table 5.4 shows a summary of the results of the evaluation in the three domains (M, L and P stands for movie, location, and people data, respectively) as well as the global precision (Pr_{Glb}), recall (Rec_{Glb}) and their harmonic mean, the F1_{Glb} score. The full results of the evaluation can be found in Appendix I. All the factors affecting the evaluation of the results were taken into account, being: number of the results queried, threshold (of semantic similarity measurement) and the global schema version used to accommodate the results (labelled by the number of datasets utilised to create it).

Threshold	GS	N	Pr$_{\mathrm{M}}$	Rec$_{\mathrm{M}}$	F1$_{\mathrm{M}}$	Pr$_{\mathrm{L}}$	Rec$_{\mathrm{L}}$	F1$_{\mathrm{L}}$	Pr$_{\mathrm{P}}$	Rec$_{\mathrm{P}}$	F1$_{\mathrm{P}}$	Pr$_{\mathrm{Glb}}$	Rec$_{\mathrm{Glb}}$	F1$_{\mathrm{Glb}}$
0.7	10-700	10-700	0.98	0.89	0.94	0.97	0.84	0.9	0.83	0.93	0.87	0.93	0.89	0.91
0.75	10-700	10-700	0.98	0.89	0.94	0.99	0.83	0.9	0.84	0.93	0.88	0.94	0.88	0.91
0.8	10-700	10-700	0.99	0.89	0.94	0.99	0.83	0.9	0.92	0.88	0.9	0.97	0.87	0.91
0.85	10-700	10-700	0.99	0.89	0.94	0.99	0.81	0.89	0.94	0.88	0.91	0.97	0.86	0.91
0.9	10-700	10-700	0.99	0.89	0.94	0.99	0.81	0.89	0.96	0.88	0.92	0.98	0.86	0.92
0.95	10-700	10-700	0.99	0.89	0.94	0.99	0.8	0.89	0.98	0.88	0.93	0.99	0.86	0.92

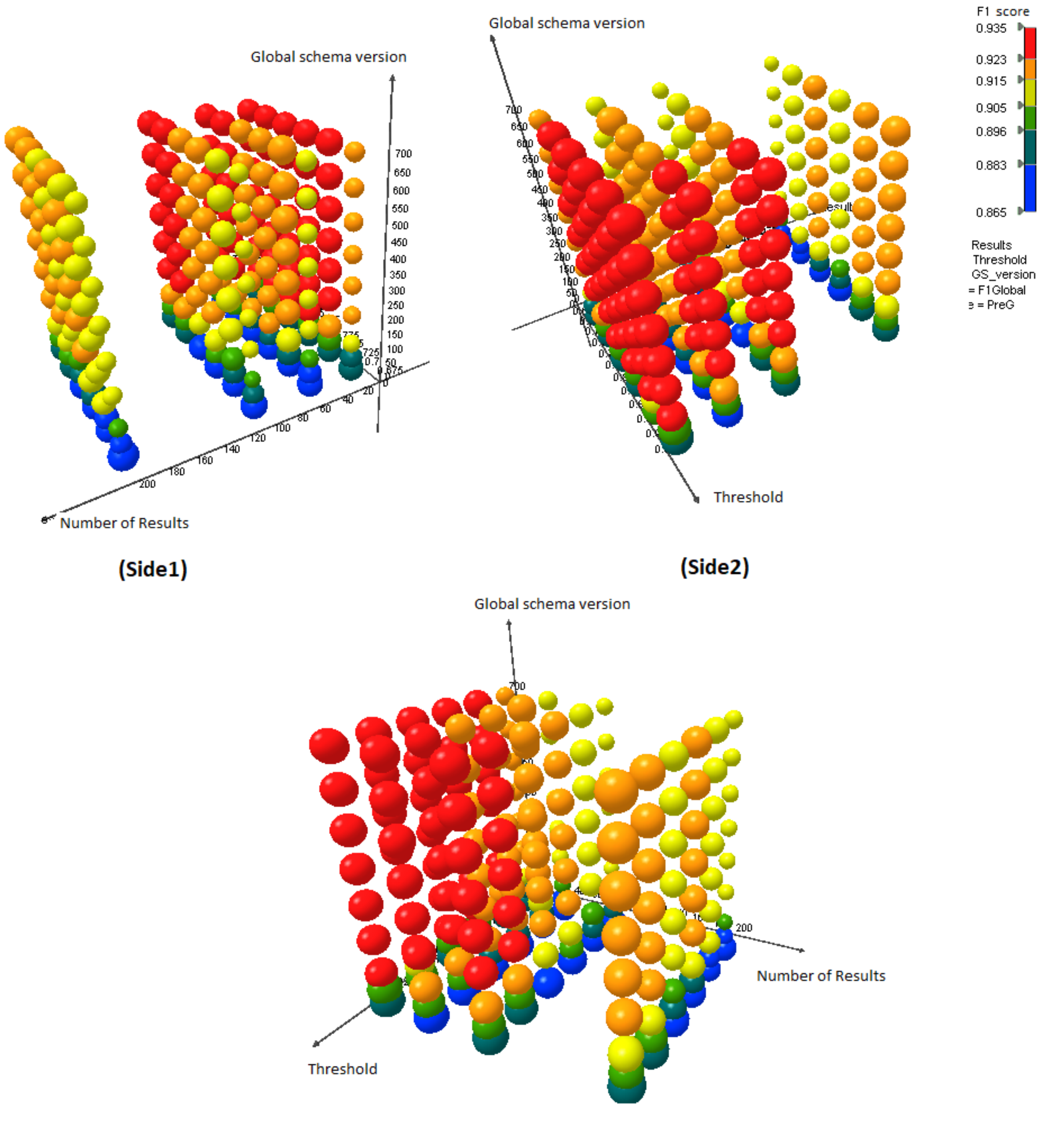
Table 5.4: Evaluation Results of SimiMatch [Author, 2017].

To analyse the results of the table, the 3D scatter diagrams in Figure 5.7 illustrate the correlation between the three parameters affecting the global F1 score of SimiMatch, which are the version of the global schema (or the number of results out of which the global schema was created), the threshold used in the semantic similarity and the number of results queried (N).

The global schema version and the number of results stops at 200 as further data does not significantly change the F1 score or the trend already reported. It can be seen that SimiMatch performs best when the threshold is more than 0.75 and the version of the global schema is greater than 100. The number of the results also has slight negative effect on the F1 score. The average precision and recall when utilising this set of parameters are 0.96 and 0.90 respectively.

In terms of performance, the time to retrieve and process 100 to 700 results grows virtually linearly from 1 to 3 seconds (more information about the implementation environment can be found in Appendix III). Figure 5.8 shows the linear growth and the insignificance of the delay caused by increasing the number of the results in SimiMatch. It conforms to the indexing approach in Algorithm 5.1 where only new properties that have not been previously processed are considered.

This finding is exploited in this chapter in order to create a domain-dependent global schema out of set of heterogeneous data sources in terms of model, structure and terminologies of their properties. The global schema contains the minimum number of properties to accommodate the results, which decreases the number of comparisons to be processed if the inputs filled with the results are to be mapped again. The approach proposed not only improves performance by pruning the number of comparisons in the mapping, but significantly minimises the possible conflicts that may occur due to duplicate meanings in data sources.



(Side1)

(Side2)

(Side3)

Figure 5.7: A 3D scatter diagram showing the correlation between the different parameters affecting the F1 score of SimiMatch [Author, 2017].

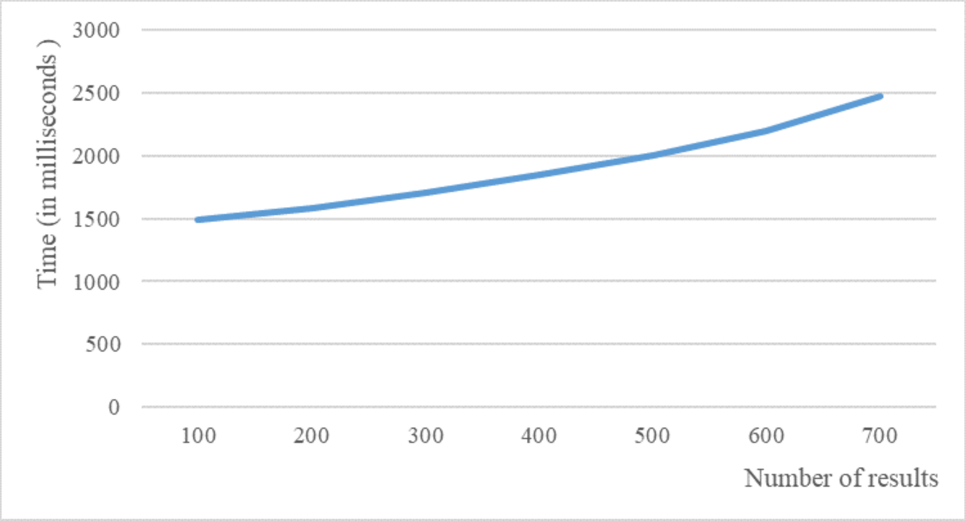


Figure 5.8: Processing time according to the number of results [Author, 2017].

5.4 Limitations

The primary aim of SimiMatch is to improve the degree of automation and to sustain the dynamism of Linked Data space. Addressing such a challenge comes at a cost. The schema integration system that aims at adapting itself to unpredictable and unknown future changes of Linked Data datasets cannot rely on and employ the structure of these datasets in the reconciliation process. Hence the first limitation of SimMatch is that it concentrates on the semantics of the textual form of the properties. The outcome of this approach can be affected negatively if the labels of the properties are encoded in a semantically meaningless syntax. But in the context of this thesis, where the sources are Web services and Linked Data sources, the datasets generally are expressed in a data-mining and parsing friendly form as they have been designed to facilitate data exchange with other services or applications. This does not exclude the need for a properties disambiguation in the pre-processing step, which will be one of the targets of future work.

5.5 Summary

The Semantic Web community has been researching schema matching and heterogeneity reconciliation for decades. Yet, the appearance of new data spaces, along with the continuing growth of existing data models, such as semi-structured data model, keep this area active. Schema matching has to cope with new challenges as the web of Linked Data, the new data space, reflects different properties and features. One of the most important challenges is taking into account its rapid and continual expansion via different vocabularies and visions in schema mapping. This chapter highlighted this challenge and proposed SimiMatch, an approach that is able to match the schemas of numerous semi-structured and Linked Data sources. It explained how it is adaptable to future changes and why it is an effective solution when utilised as part of a data integration context (see Section 5.3.3). An in-depth explanation of each of its stages was provided, which include the external and internal extraction of the semantic distinct properties (see Section 5.2.1), and the creation and the indexing of the global schema (see Sections 5.2.2 and 5.2.3). The implementation and evaluation results (see Section 5.3) in three domains, being movies, locations and people's data, were presented to show the effectiveness of the approach.

SimiMatch contributes towards a virtual integration system called SemiLD (see Chapter 7) that will be able to provide transparent access to heterogeneous and autonomous sources. It addresses the challenge of sustaining the continuous changes of a large-scale Web of Data via similarity measurement and property matching. SimiMatch is also adapted and embedded in the novel data interlinking system that will be presented in the next chapter. The new data interlinking approach is an approach that is able to effectively provide identity links between RDF datasets that are not associate with any ontology with datasets that are part of the Linked Data cloud.

Chapter 6

LinkD: Element-based Data Interlinking of RDF datasets in Linked Data.

Sites need to be able to interact in one single, universal space.

Tim Berners-Lee

6.1 Introduction

The problem of interlinking datasets in and with the Linked Data cloud has been one of major challenges and an important research subject in the Semantic Web. The existence of links enhances the value of the data and facilitate information discovery [Getoor and Diehl, 2005]. Datasets residing on dispersed data sources without links resemble islands of data [Hassanzadeh, 2013], where every island stores part of the data needed by the user. To gather all the necessary pieces of information, the user needs to manually find each island.

Even though these links between things in the world may be implicit in semi-structured data returned from Web APIs, semantic links in Linked Data allows ”Web publishers to make these links explicit, and in such a way that RDF-aware applications can follow them to discover more data” [Heath and Bizer, 2011, p. 8]. This leads to a Web where data is more discoverable for both machine and human users, and therefore more usable.

Creating semantic links between different Linked Datasets creates the Web of Data, a global database where data is connected to relevant data. The value of the Web of Data rises and falls with the amount and the quality of links between data sources [Volz *et al.*, 2009]. Providing a tool to facilitate the migrating of these datasets to the Web of Data will enhance the value of both the published Linked Data and migrated datasets.

Publishing semi-structured data as Linked Data involves many steps, including enhancing the quality of the converted semi-structured data [Yeganeh *et al.*, 2011], such as by merging duplicated resources, identifying resources type, internally linking related entities etc. It is the stage that xCurator [Yeganeh *et al.*, 2011] concentrated on, the only tool found at the time of writing this thesis that specifically aims at publishing and linking semi-structured as Linked Data. xCurator also has a module to provide external links between the transformed semi-structured data and the Web of Linked Data, which is the other stage, but it is considered secondary and has neither been detailed nor evaluated by their authors.

Unlike publishing semi-structured in the Web of Data, there are many tools and approaches proposed to interlink Linked Data. Most of these tools were proposed as part of the yearly event of OAEI [Jimenez-Ruiz, 2017]. Their aims do not seem to differ considerably from this chapter's objective, which is to link transformed semi-structured data (RDF file) with the Web of Data, or the second stage of semi-structured linking, but they are not adapted to be directly utilised in this use case. The proposed interlinking approaches employs various information, such as the structure and resources types, in order to find identical instances between a set of source resources and target resources. These information and options are frequently unavailable or incomplete when interlinking automatically transformed semi-structured data using a fixed semi-structured to RDF transformation. Additionally, it is time consuming and requires a significant amount of input and manual setting to convert a considerable amount of semi-structured data using the ontology-based semi-structured to RDF transformation (see Section 2.5.4) in order to generate some structural information, which can be incomplete, imprecise, or inaccurate.

The interlinking is frequently addressed on existing data. LinkD, the proposed approach, however, is a system that verifies in first place the existence of the URI of the resource being published in the cloud in order to establish links with it. The overall aim of the research is to facilitate the following of best practices and recommendations in publishing data into the Linked Open Data cloud. The input of the system is transformed semi-structured data (XML and JSON) using a fixed semi-structured to RDF transformation. The focus of this chapter is to provide a new design and tool to externally link transformed semi-structured data. The main contribution of the presented chapter is the use of the domain to allocate variable weights in measuring the similarity of the instances, according to the significance of their properties in defining the identity of the dataset. Figure 6.1 shows the relation of LinkD with reference to the other two systems proposed in this thesis.

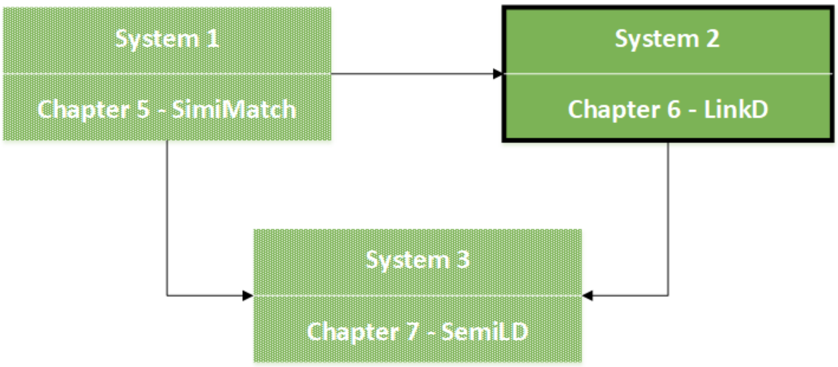


Figure 6.1: Relation of LinkD approach with reference to other systems of this thesis [Author, 2017].

6.2 An Overview of the Contributed LinkD Approach

The scope of this chapter is to design a system that can externally link the output of the fixed Semi-structured to RDF transformation, which is an RDF file with no explicit meaning or structure associated with it. The transformation itself is beyond the objectives of this thesis. The RDF file are a set of triples describing resources according to the hierarchy of the transformed tree-based XML or JSON file. The fixed transformation is selected because it can be automated and requires less pre-transformation effort (see Section 2.5.4), which is necessary for the system to be adapted to interlink large-scale datasets. Interlinking datasets with the Web of Data (a large database of resources) necessitates the utilisation of lightweight processes and avoidance of operations such as type identification or type-related comparisons.

Based on the characteristics of the output of the fixed XML to RDF transformation of semi-structured dataset and the requirements of the approach, the designed system needs to be based on string and set similarity. The labels of the semi-structured data are the only features that can be confirmed to be maintained after the fixed transformation. Additionally, it would be challenging to employ resources expensive operations and apply them to a large amount of candidates retrieved from different name-spaces of Linked Data cloud.

The existing approach that conforms to most of these characteristics and requirements is SERIMI (see Section 4.4.1). Hence the approach proposed is drawn from the SERIMI system [Araujo *et al.*, 2011], and based on the assumptions of a variety of domain-dependent interlinking systems, for instance: EventMedia [Khrouf and Troncy, 2016] and the system proposed by Zhang *et al.* [2013]. SERIMI is one of the advanced approaches allowing the matching of instances of two large-scale datasets without the need of pre-knowledge about their data, schema or domain. However, properties weights of the instances have not been taken into consideration in the SERIMI approach. EventMedia and the approach proposed by Zhang *et al.* [2013] are projects aiming at interlinking data in a specific domain. As part of their project, they tried to find the most accurate weights to be given to the properties in their particular domain.

The idea presented in this work is to add a domain detection phase, before matching the instances, in order to impose variable weights to the properties of the data being interlinked. The weights are extracted from the existing observations of the specialised systems (cited in the previous paragraph) and extended by the author's observation in other domains. The proposed system aims at interlinking an RDF dataset with its counterpart in the Linked Data cloud using different algorithms for similarity measurement, taking into account the domain of the dataset being interlinked. The main aim of this system is to facilitate and to automate following best practices in publishing data into the Linked Data cloud. The contribution of this system is to apply variable weights to the value of the matched resource properties according to the domain extracted e.g. the similarity of the values of the resource properties longitude and latitude will be given more weights in the case where the domain detected is geospatial; whereas, if the domain is an event, the coefficient of the similarity index between the value of resource properties time and place will be higher than other properties.

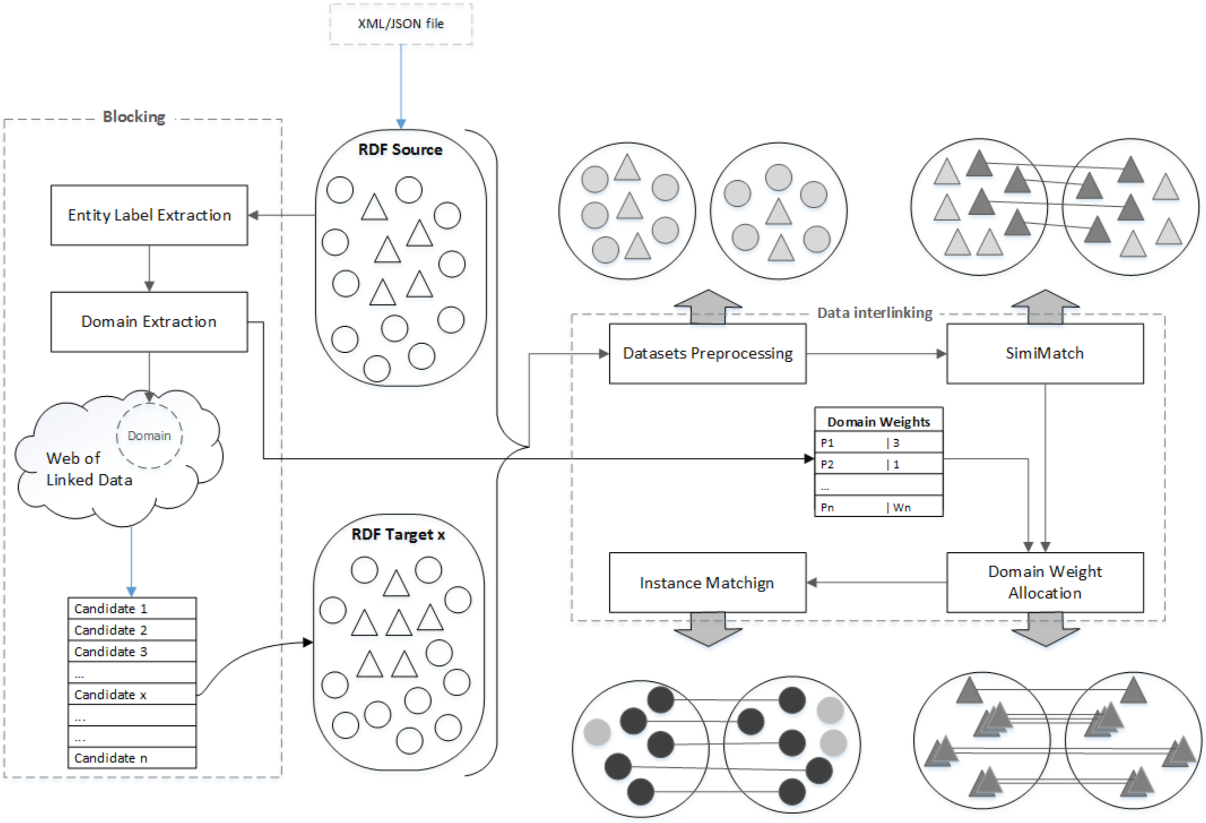


Figure 6.2: General architecture of LinkD [Author, 2017].

An asymmetric and unsupervised approach is used in LinkD to compute the similarities in the system presented. It would be almost impractical to process the integral heterogeneous web of Linked Data using a manual or supervised (based on conceiving training set) approaches due to the prior knowledge or the resource needed to apply such techniques [Araujo *et al.*, 2011]. The common drawback of unsupervised algorithms, however, is the high computational cost required to implement them.

Figure 6.2 describes the different processes of the new LinkD approach proposed by the author. It can be seen that the dataset goes through many stages before matches in the cloud can be found. These stages can be organised and grouped in two main phases as follows:

6.2.1 Candidate Selection (Blocking)

Contrary to the common instance matching approaches, LinkD starts with one dataset, which is the source dataset. The source dataset is an RDF file derived from XML/JSON files that might not be described by an ontology or associate with any meaningful structure. The target dataset is retrieved after running a keyword SPARQL search query on the Linked Data namespaces considered. The SPARQL query is composed by extracting the domain and the entity label (labels that represent the dataset) of the source dataset. The latter is generally the content of the property title, name or label that have a literal value with less than 200 characters [Araujo *et al.*, 2011].

The domain extraction is an important phase in finding the counterparts of a dataset in Linked Data cloud in the proposed system. The domain is determined by extracting the content of the property rdf:type and classify it with one of the pre-defined domain categories available in the system. In the case where rdf:type is not used, the domain is selected manually.

Having the domain extracted, the potential candidate for the interlinking will be significantly reduced and limited as a result of more specified SPARQL keyword search. In the next stage (Date interlinking), different weights will be applied according to the domain detected. More weight will be given to the properties that define more the identity of the dataset (properties with unique values) and create less conflict with the other datasets, example: longitude and latitude for location or ISBN number for books.

Example 6.1 is a template of SPARQL query to search for target datasets from DBpedia.

Example 6.1: A template of a SPARQL query to search for target datasets from DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?resource WHERE {
       ?domain rdf:type <http://schema.org/''source_domain''.
       ?resource foaf:name ?resource_title.
       ?filter contains(?resouce_title, ''source_title'')
}
```

6.2.2 Data Interlinking

After the blocking stage, the system compares between the properties of each of the candidates with the properties of the sources datasets in order to determine which candidate can be linked with it. Three types of links exist in Linked Data: relationship, identity or vocabulary links.

Identity links or `owl:samAs` are the most common type of links addressed by the existing interlinking systems and this is the focus of LinkD. Along with allowing the representation of semantic equivalence in an independent and reusable way, `owl:samAs` can serve as hints to reasoner systems on how to unify data.

The data interlinking stage consists of four steps:

6.2.2.1 Preparing the Datasets

A Pre-processing step is performed to extract the value from the resources that are described using URIs, which is its last part according to Linked Data principles. Commas and underlines will be also replaced by spaces to improve the accuracy of the matching algorithms.

Example: the output of the `http://dbpedia.org/page/London_River` after pre-processing is `London River`.

6.2.2.2 Property Alignment (SimiMatch)

This stage is responsible for matching between the semantically similar properties of the source and the target (candidate for interlinking) dataset. The SimiMatch tool is adapted and utilised in LinkD to generate matching rules between the two schemas instead of creating a global schema. Algorithm 6.1 explains the adaptation of SimiMatch to align properties of two properties sets in LinkD. SimiMatch measures the semantic distance between the label of the predicates of source and the candidate datasets and compares it to a threshold. The matches rules are expressed using a map data structures. The map stores data in the form of key and value pairs where every key is unique. As a result, Algorithm 6.1 iterates, for each key, through values in order to find the pair with the highest semantic similarity (or the lowest similarity distance) score that is above the threshold. If new pair with a higher similarity score is discovered, the new value of the key in **matches** will replace the previous one. A candidate property is matched with one of the source properties.

Algorithm 6.1 SimiMatch in LinkD [Author, 2017].

```
Input:  set1, set2: PropertiesSets
        threshold
Output: matches: Map

sizeSet1 = size (set1)
sizeSet2 = size (set2)

for i=0 And i < sizeSet1 do
    temp_distance=0;
    for j=0 And j < sizeSet2 do
        distance = SemanticDistance(set1[i], set2[j]);
        if distance > threshold AND distance > temp_distance then
            matches.add(set1[i], set2[j])
        else
            i++; j++;
        end if
    end for
end for

return matches;
```

6.2.2.3 Domain Weight Allocation

It is observed and validated in Section 5.3.2 that RDF datasets referring to the same real world object or describing resources in the same domain share roughly the same properties even though the syntax may be expressed differently. More importantly, it is noticed [Kettouch *et al.*, 2015a] in many systems for interlinking domain-dependent Linked Data, that some properties are more precise in defining the identity of a dataset. Therefore, it is more practical to employ and concentrate on them in the interlinking process. Having a prediction of a limited list of the properties that will matched in a particular domain, it become feasible to set rules for allocating weights for the similarity index of the content of these properties.

Many factors come into play in defining the weight of the properties in Data interlinking with the Web of Linked Data. Three decisive criteria are identified in this chapter:

- **Number of repetition of the property and its content:** Similarly to the primary key in a database, the property or the instance with higher weight need to be unique; thus, its value should not repeat in the candidate set or in a particular domain.

- **Content length:** the result of the string similarity (both semantic and syntactic) applied in LinkD can be negatively affected by long literals.
- **Time relatedness:** this criterion allow to find unique properties that do not change over time, something that can mislead the interlinking process. To find whether a property is time related, at least two version of the published resource need to be compared.

Several approaches attempted to improve data interlinking and instance matching performance and precision using properties weights, such as: RIMOM [Zheng *et al.*, 2013], CODI [Huber *et al.*, 2011] and BOEMIE [Castano *et al.*, 2008]. These approaches are not adapted to be utilised in LinkD for many reasons, including:

- They do not consider all these criteria, or;
- The properties with distinct values are not domain-dependent, or;
- The processing of the weight is embedded in the matching process; or;
- They are not element based and depend on the structure.

The first and second reasons can make the weight allocation process incomplete. The third and last reasons can make the interlinking system not suitable to process large scale of data as it can affect considerably performance and computational time.

The weight allocation used LinkD is an extension of the weight generator proposed by Nath *et al.* [2014], which is the weight allocation solution close to meeting the criteria identified in this thesis. It is an approach that is based on penalising repeated properties by a negative probability factor `np`. This thesis extended this concept by adding two other negative factors that represents ”properties’ content length” and ”properties’ time-relatedness” in order to cover all the identified requirement, which allows LinkD to calculate the uniqueness of the properties and their abilities to define the datasets they describe. The following equation defines λ, the function to calculate the weight of a property p.

$$\lambda(p) = (1.0 - np1(p)) * (1.0 - np2(p)) * (1.0 - np3(p)) * (1.0 - np4(p))$$

The weight λ is penalised by `np1`, the ratio of the number of repetition to the number of instances the property belongs to, as described below.

$$np1 = \frac{Rep(p)}{\mid i \ni p \mid}$$

`np2` represents the repetition of the content of resource properties over the total number of instances (I) (which it does not necessarily belongs to), as defined below.

$$np1 = \frac{Rep(p)}{\mid I \mid}$$

The value `np3` penalises properties with literals that are longer than 200 characters, which is what is considered in SERIMI as representative entity label.

$$np3 = \begin{cases} 0 & length(obj(p)) \leqslant 200 \; else \\ \\ \dfrac{1}{length(obj(p))} & \end{cases}$$

The value of `np4` is 1 if the object of the property can change from version to version ($ver_{1..n}$) and/or depends on time.

$$np4 = \begin{cases} 0 & if \, obj(p_{ver1}) \neq obj(p_{ver2}) \; else \\ 1 & \end{cases}$$

6.2.2.4 Instance Matching

Having the list of matched properties between the source dataset and each of the target datasets, the system extracts their content (instance). The similarity of the instances is then measured using Jaro-Winkler algorithm. The Jaro distance d between two strings is the result of the following equation:

$$d = \begin{cases} 0 & if \, m = 0 \\ \\ \frac{1}{3}(\frac{m}{s1} + \frac{m}{s2} + \frac{m-t}{m}) & if \, m \neq 0 \end{cases}$$

Where:

- $s1$ and $s2$ are the label of the instance of the source dataset and the label of the instance of the target dataset respectively.
- m is the number of matching characters.
- t is half the number of transpositions.

Finally, the similarities of the properties and their instances are combined in a linear combination of the measures described in Tversky's contrast model as shown in the equation below:

$$Tversky(A,B) = \lambda(A \cap B) - \alpha f(AB) - \beta f(BA)$$

Where: α, β, and $\lambda \geq 0$. Three parts can be noticed in the Tversky model:

- (A ∩ B) represents the set of common properties between A and B
- (A - B) are the set of distinct properties between A and B
- (B - A) are the set of distinct properties between B and A

The coefficients α, β, and λ represents the weights of the commonalities and differences in the equation. Since the distinctness between the resources is not relevant in our case, α and β are set 0. One of the contribution of the author's proposed approach lies in setting the value of λ according to the domain of the source dataset.

6.3 Implementation and Evaluation

The flowchart in Figure 6.3 explains the implementation of LinkD. The parts A, B, C and D represents the stages (explained in Section 6.2.2) ”preparing the datasets”, ”property alignment”, ”domain weight allocation” and ”instance matching” respectively.

ID	Source	Target	Domain	Source Pairs	Target Pairs	Target Domain Pairs
D1	LinkedMDB	DBpedia	movies	10108	474M	77769
D2	LinkedMDB	DBpedia	people	3650		831558
D3	NYTimes	DBpedia	locations	2083		639450
D4	NYTimes	DBpedia	people	4588		831558
D5	NYTimes	DBpedia	organisations	1274		209471

Table 6.1: Details of the considered datasets from IM@OAEI2011 [Author, 2017].

Although, strictly speaking, LinkD is addressing rather a different problem (interlinking datasets based on specific restrictions as explained in Section 6.1) to data interlinking approaches presented at the OAEI, this section used IM@OAEI2011 benchmark to allow the numerical and direct comparison against other relatively successful data interlinking approaches. IM@OAEI2011 is chosen because it was utilised by many other relatively successful approaches.

Five datasets from IM@OAEI2011 were taken into consideration, of which four domains being: movies, people, locations and organisations, and three Linked Data sources being: DBpedia, LinkedMDB and NYTimes. In the other approaches utilising this benchmark, the number of target pairs is the same as the source pairs as their aims is to find identity links between two sets. In LinkD; however, the aim is to provide links with the Linked Data cloud; hence, the target pairs are the entire DBpedia repository (English DBpedia 3.9[45]). Although DBpedia is not the Linked Data cloud, it is the largest Linked Data repository that can be used as a target to evaluate LinkD against large-scale data. Having many Linked Data providers on the target side removes the possibility of approximate numerical comparison against other related systems. Table 6.1 gives the overview of the considered datasets (D1 to D5).

6.3.1 Evaluation of the Blocking stage

Table 6.2 shows the result of the evaluation of the blocking stage. To calculate the pair completeness (PC), the evaluation needs to have a gold standard upon which the true positive (correct) candidates can be counted. It is something that related approaches do not clarify in their evaluation and the results are displayed without giving details about the values of the components of

[45] http://wiki.dbpedia.org/services-resources/datasets/data-set-39/downloads-39

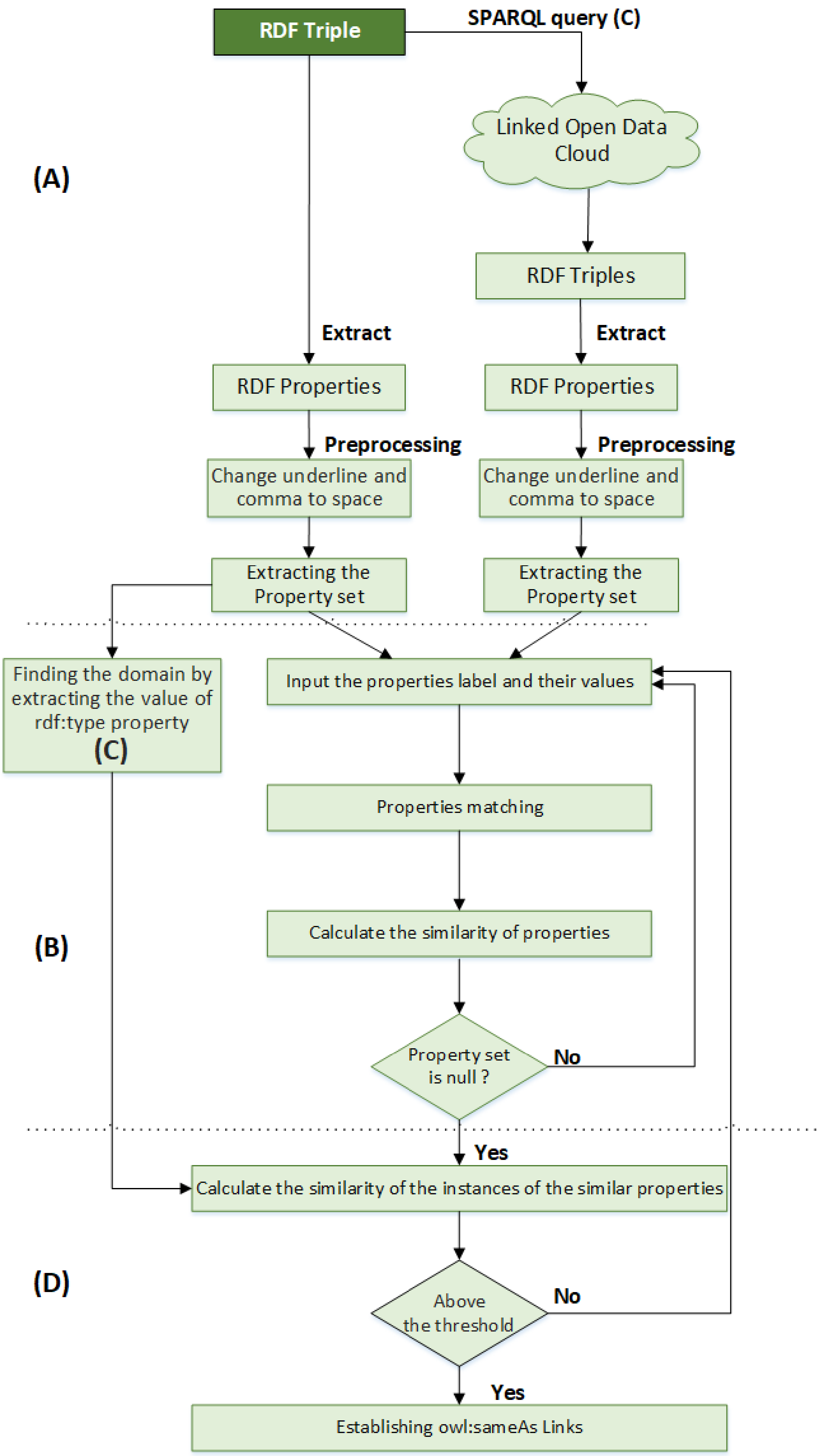

RDF Triple
SPARQL query (C)
Linked Open Data Cloud
(A)
RDF Triples
Extract
Extract
RDF Properties
RDF Properties
Preprocessing
Preprocessing
Change underline and comma to space
Change underline and comma to space
Extracting the Property set
Extracting the Property set
Finding the domain by extracting the value of rdf:type property (C)
Input the properties label and their values
Properties matching
(B)
Calculate the similarity of properties
Property set is null ?
No
Yes
Calculate the similarity of the instances of the similar properties
(D)
Above the threshold
No
Yes
Establishing owl:sameAs Links

Figure 6.3: Flowchart describing the implementation of LinkD [Author, 2017].

ID	Instance Pairs	Target Pairs (after blocking)	Correct Candidates	Candidate Pairs	RR	PC
D1	786079023	16868	9829	170501744	0.78	0.97
D2	3035186700	24123	3447	88048950	0.97	0.94
D3	1331974 350	12545	2000	42044235	0.98	0.96
D4	3815188104	18740	4496	85155120	0.98	0.98
D5	266866054	6672	1269	8500128	0.97	0.97

Table 6.2: Results of the blocking stage [Author, 2017].

ID	Source Pairs	Links Discovered	Matched Instances	Unmatched Instances	Properties Aligned	Runtime (seconds)	Rec	Pr	F1
D1	10108	10989	9586	522	1044776	51007	0.95	0.87	0.91
D2	3650	3896	3388	262	706400	2241	0.93	0.87	0.9
D3	2083	1825	1811	272	327825	1774	0.87	0.99	0.93
D4	4588	4765	4476	112	338037	2247	0.98	0.94	0.96
D5	1274	1306	1198	76	135015	1519	0.94	0.92	0.93

Table 6.3: Results of the instance matching stage [Author, 2017].

the equation.

The correct candidates in Table 6.2 are based on estimating the number of occurrences of the actual sameAs links of the datasets in the target pairs.

6.3.2 Evaluation of Instance Matching Stage

Table 6.3 reports the results of the instance matching stage. These values represents the lower band results as the benchmark utilised is created in 2011, which means new resources that may contains true positives that are not listed could have been published since then.

The weight allocation stage is run before the interlinking. It is a separate process that it is not repeated for every interlinking unless new datasets that belong to a domain that has not been previously processed are added. The result is an array for every domain considered that contains properties labels and their weights.

ID	**LinkD**			**SLINT**			**SERIMI**			**Agree.Maker**		
	Pr	**Rec**	**F1**	**Pr**	**Rec**	**F1**	**Pr**	**Rec**	**F1**	**Pr**	**Rec**	**F1**
D1	0.95	0.87	0.91	0.93	0.99	0.96	0.79	0.81	0.8	0.9	0.8	0.85
D2	0.93	0.87	0.9									
D3	0.87	0.99	0.93	0.96	0.97	0.96	0.69	0.67	0.68	0.79	0.61	0.69
D4	0.98	0.94	0.96	0.99	0.98	0.99	0.94	0.94	0.94	0.98	0.8	0.88
D5	0.94	0.92	0.93	0.98	0.95	0.96	0.89	0.87	0.88	0.84	0.67	0.74
	Target Pairs											
D1	474M			10108								
D2				3650								
D3				2083								
D4				4588								
D5				1274								

Table 6.4: Comparison of LinkD with related systems [Author, 2017].

6.3.3 Comparison with Previous Interlinking Systems

Table 6.4 provides a comparison between LinkD and popular interlinking systems. It can be clearly noticed the extent of the improvements that LinkD introduced to SERIMI. D1 and D2 are joined together in the other approaches, in LinkD, however, they are separated into two domains being movies and people (actors, writers, director, etc.).

Although Table 6.4 shows that SLINT is performing better in terms of F1 score in all the datasets of IM@OAEI2011 considered including LinkD, the scale of the targeted data is significantly larger in LinkD. This highlights that the nature of the problem addressed is not the same. It is the only way, however, to numerically compare LinkD and to show that despite the difference in terms of the scale of the data targeted, LinkD performance is relatively good.

Table 6.5 reports the time it took LinkD to process the datasets D1-D5 comparing SLINT (more information about the implementation environment can be found in Appendix III). It is not a direct comparison, the difference in terms of the number of target pairs is highlighted. For instance, with the presumption that SLINT performance strongly and directly correlates with the amount of the target datasets, its performance for D1 would be 474 Millions divided by 10108, multiplied by 67, the results is approximately 3,141,867 seconds.

ID	**LinkD (seconds) for 474M**	**SLINT (seconds)**
D1	51 007	67 (for 13758)
D2	2 241	
D3	1 774	3.55(for 2083)
D4	2 247	12.74 (for 4588)
D5	1 519	4.29 (for 1274)

Table 6.5: Performance evaluation of LinkD against SLINT [Author, 2017].

6.4 Summary

This section proposed a new interlinking approach, called LinkD, that provides identity links between a single source dataset and the Linked Data cloud, using the domain as reference in applying variable weights in the similarity measurement. The approach proposed goes through two stages to achieve this aim: the blocking (see Section 6.2.1) and data interlinking (see Section 5.3. A variety of distance measurement tools and algorithms were used to calculate the similarity between the labels describing the resources, including UMBC EBIQUITY-COR [Han *et al.*, 2013] (to measure semantic distance) and Jaro-Winkler (to measure the similarity between two sets or strings, (see Section 6.2.2.4)). Neither the structure nor the ontology of the dataset were considered on the proposed system in order to maintain its feasibility to target large-scale data. The major challenges faced are the high computational cost and the dynamic allocation of the weights according to the domain and the number of the matched properties. The evaluation of different components (see Section 6.3) showed that LinkD is able to target significantly larger datasets whilst maintaining high quality measures (precision, recall and F1 score) (see Section 6.3.3).

A new data integration approach along with its prototypes are presented in the next chapter. It aims at ”consuming” two important data models available on the Web being semi-structured and Linked Data. The new data integration approach of the next chapter is the short-term solution the author proposes in this thesis to bridge between semi-structured and Linked Data.

Chapter 7

SemiLD: Keyword Search over Semi-Structured and Linked Data

With data collection, 'The sooner the better' is always the best answer.

Marissa Mayer

7.1 Introduction

The distributed and the autonomous nature of Linked Data sources make it unlikely to sustain the use of one model in representing data in a particular domain. Each source has its specificities, conditions, and a different vision on the way to expand. The internal links (pointers to data within the local Linked Data source) can be relatively consistent and easily maintained as the data publishers are aware of the changes occurring on their data repositories. The external links, however, represent a challenging task, given they connect two vocabularies, models or views that are managed and situated in separate locations and are regularly changing. This dynamism of the relations between the integration system and sources, and the continuous expansion of Web of Data, along with data freshness requirements, suggests that a solution would need to integrate data virtually on-the-fly Kettouch *et al.* [2015b]. SemiLD combines the use of ontologies, to obtain high precision, with property matching, to achieve a high degree of automation while targeting large-scale data.

The distributed and autonomous nature of Semantic Web sources, as explained in Section 3.5, imposes more challenges and leads to heterogeneous terminologies. Multiple ontologies and vocabularies can be utilised to represent similar information in a particular domain. On the other hand, as stated in Section 5.2.1 (and confirmed in Section 5.2.2), although different dispersed RDF datasets, describing data in the same domain, may not be exactly identical, they overlap in the semantics of their properties. This statement is as valid for Linked Data sources as for semi-structured data sources. Semi-structured data are frequently described using XML or JSON technologies, where tags play the same role as properties in RDF.

This chapter proposes a mediator-based modular architecture to integrate on-the-fly heterogeneous semi-structured and Linked Data. This chapter presents SemiLD, a novel approach to integrate semi-structure and Linked Data. SimiMatch and LinkD are adapted and included as modules in SemiLD, as Figure 7.1 shows. This chapter provides two prototypes of the SemiLD. The first prototype is a highly-automated keyword search system that retrieves its input from various SPARQL endpoints and Web APIs. The second prototype is movie collection manager and is provided to highlight the adaptivity of the author's proposed approach as well as to present another working scenario and an evaluation method. The evaluation of the system illustrates the high performance, usability and efficiency of the contributed approach.

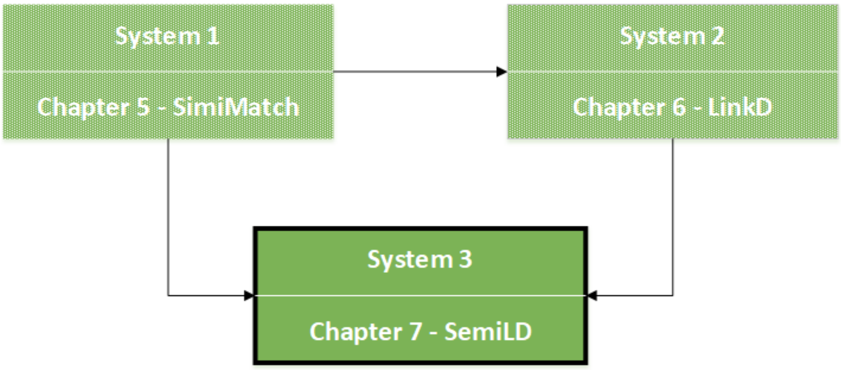


Figure 7.1: Relation of SemiLD approach with reference to other systems of this thesis [Author, 2017].

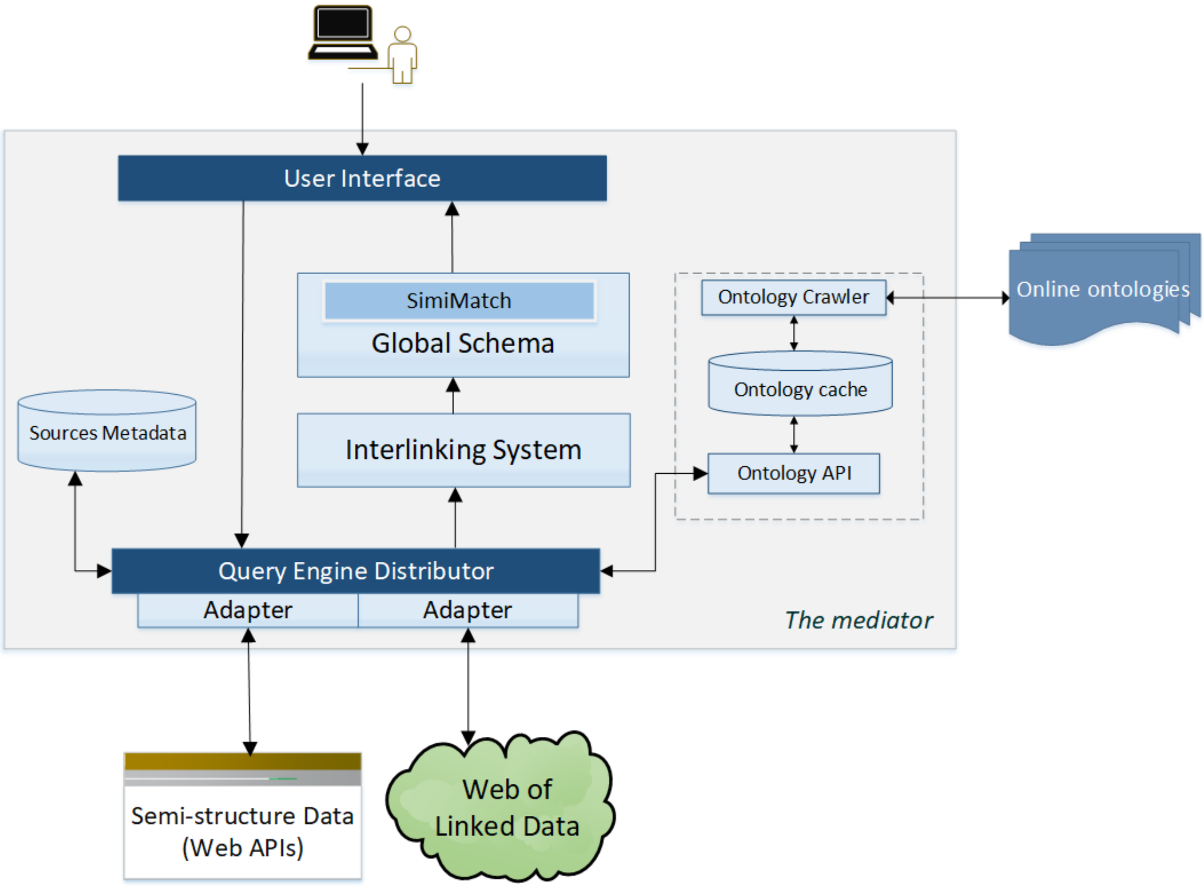


Figure 7.2: General architecture of SemiLD [Author, 2017].

7.2 The new SemiLD Approach

Figure 7.2 illustrates the author's proposed modular architecture of SemiLD solution. SemiLD uses a global schema, as part of the mediator, that has the ability to learn and expand automatically. The global schema is an XML file, constructed using SimiMatch, that contains all the possible and potential properties (tags) that the system may retrieve in running a query in a specific domain. By using the interlinking module, the system is able to match properties and transfer values from different semi-structured and Linked Data sources to the global schema, in a specific domain. The ontologies (or vocabularies) are used in the system to support the formulating of the query.

SemiLD modular architecture consists of Six (6) main components:

7.2.1 User Interface

A user-friendly interface is a crucial component in order to allow non-expert users to benefit from the service provided. Ontologies and vocabularies may contribute to the writing of the queries by adapting them to the vocabulary or the structure used. As explained in Section 3.4.1, this enabled new opportunities that classic environments did not have. Previously developers had to specify the search requirements, data model and the query language in order to be able to write the query.

To hide the complexity of SPARQL queries and API HTTP requests, the proposed interface allows keywords as an input. As argued in Section 3.5, keyword searches are considered the most usable mean of accessing information on the Internet. Their limitation, however, is centred on their lack of expressivity as little information is provided in the query. This problem can be addressed in the implementation by adding optional and domain-related text fields that can match the structure of some of the sources. This will enable adding additional information to the query. In this approach, the expressivity is addressed by offering the possibility to post-filter results according to all the properties of all the sources. For every property, it provides a list of the potential values to filter with. This is because of the global schema that reconciles between the heterogeneity of the properties of different sources and transfer their data to a uniform repository file. The usability is also addressed in the interfaces of the prototypes (see Section 7.3.1, 7.3.2 and 7.3.3) by providing some features that are adapted to the domain of the data sources searched.

The output of this module is one main keyword, which will be used to retrieve the ontologies needed in the query engine module and the results from all relevant sources. Other optional keywords and parameters may also provided to increase the expressivity of the query but they may not be supported by all the sources considered (depending on the structure and the number of parameters accepted by their Web APIs). The latter need to be defined and included in the interface. For example: Web APIs for movies require mandatorily one keyword of the title, but some of them also permit to send more information, such as: the year and the genre, as an optional input. In the interface, a text field to write the year can be provided in order to be used with a limited list of sources, but it is not utilised in the processing nor in the retrieval of ontologies.

7.2.2 Ontology

Ontologies are a flexible, extensible, and scalable mechanism to describe and structure stored information [Ramis *et al.*, 2014]. The information is encoded in ontologies in the form of concepts and properties linked via semantic relations. SemiLD uses ontologies for two tasks in the system:

- To support the query engine distributor in forming and structuring the query against Linked Data sources by allocating the appropriate vocabulary.
- To cluster the outputs according to the domain in a domain-independent application of the approach to ensure an accurate and conflict-free property matching.

The ontology module is based on what is proposed in Fatima *et al.* [2014]. It consists of many parts, rather than one central ontology repository, that crawl for the ontologies and vocabularies of the Linked Data namespaces considered, in order to support and sustain their continuous changes. The following three components are part of the ontology module:

7.2.2.1 Ontology Crawler

The Ontology Crawler is responsible for caching online ontologies and checking regularly for updates. Using the list of saved ontologies and their paths, the crawler module checks regularly not only for the existence of the file containing the ontology, but also for the date of creation compared to the online version. The crawler is key to maintaining the high performance, precision and efficiency in search mining ”by semantically discovering, formatting, and indexing information” [Dong and Hussain, 2014, p. 2]. The process is demonstrated in Algorithm 7.1.

Algorithm 7.1 Ontology Crawler [Author, 2017]

Input: ontology_list, cache_path

```
records = count(ontology_list)
list← new queue
while record + 1 < records do
    file_exists = check_if cached(listrecord,cache_path)
    if file_exists == 1 then
        creation_date= get_creation_date(listrecord)
        modify_date= check_modify_date(listrecord)
        if modify_date >creation_date then
            file = fetch_revision(listrecord)
            update_index(file)
            update_cache(file)
        end if
    else
        file = fetch_file(listrecord)
        insert_index(file)
        save_cache(file)
    end if
end while
```

7.2.2.2 Ontology Cache

It is a repository of ontologies that have been retrieved from running previous queries. This eliminates the time that the system would take to mine for ontologies online. A simple indexer is built into the ontology cache module that classifies the ontologies URIs according to the keyword and Linked Data source requested.

7.2.2.3 Ontology API

The API is an intermediate between the query engine distributor and the rest of ontology modules of the system. After receiving a keyword from the engine distributor, it considers the sources then outputs a list of tags and vocabularies needed to form the query.

7.2.3 Sources Metadata

The metadata modules provides the minimum information required for the functioning of the system. It stores the path to access the sources, such as Web API URI and structure and SPARQL endpoint link and version. In addition, the index of SimiMatch (see Section 5.2.3), specifying which global schemas has been previously created, is also saved in the metadata.

7.2.4 Query Engine Distributor

The query engine distributor is the central subsystem where all the information is gathered from the ontology module and the sources metadata store in order to perform the integration in both directions. Algorithm 7.2 illustrates the overall functioning of the query engine distributor. Having received the information needed, the system prepares the query to be sent to the sources (top-down part). In the ascending direction (bottom-up), the query engine, through adapters, parses the outputs retrieved from the sources. The results are then organised into a list of datasets in order to facilitate the next step (interlinking). For every semi-structured source,

Algorithm 7.2 Query Engine [Author, 2017]

```
Input: Keyword, Metadata, ontology_index_file
                                                    ▷ Top-down direction
for each source in Metadata.source_list() do
    Query_structure= OntologyAPI(source, keyword)
    Load (Adapter)
    Connect (metadata.sourceLocation())
    Query = PrepareQuery (Key_word, Query_structure)
    Result_file = Execute (Query)
end for each
                                                    ▷ Bottom-up direction
Results = Parse (Result_file)
rows_count = Results.length()
outputs.setSize(rows_count)
for $i$=0 to rows_count do
    outputs($i$).setcontent(Results.getrow($i$).getcontent())
end for

return outputs
```

an instance of an adapter is created to process its queries and outputs. It contains pre-defined functions to identify the source type and to send, parse and organise the information. Two roles are assigned to this module: first, it establishes the connection with the API URI or the SPARQL endpoint of the source; second, it gathers all the information to reformulate the query and sends it. Having retrieved the files containing the results from the sources and identifying both the format and the model, the adapters do the reverse process, splitting the results into a list of datasets.

7.2.5 SimiMatch

As indicated at the end of the Chapter 5, SimiMatch is designed to be utilised as part of a data integration and an interlinking system. It is the module responsible for reconciling the structural heterogeneity between the semi-structured and Linked Data. This section explained how SimiMatch is embedded and adapted as a component in SemiLD.

7.2.5.1 Semantic Distinction

The algorithms of the global schema creation and update utilise a common method, described in Algorithm 5.1 (in Section 5.2.2), to calculate the semantic distance between each item of two sets of properties. The output of this method is an index and a set of the semantically distinct properties that the two inputs contain.

7.2.5.2 Internal Properties Extraction

As explained in Section 5.2.1, the extraction of properties in Linked Data is different from the semi-structured data model. One result is sufficient in semi-structured data since all datasets share the same properties. Various vocabularies, however, are used in describing Linked Data. These vocabularies share many semantically properties when describing data in the same domain. Therefore, SimiMatch discards semantically duplicates properties in order solve this redundancy and to avoid conflicts in generating matching rules.

Algorithm 7.3 Properties Extraction in a Linked Data source [Author, 2017]

```
Input: $PR_x$: Properties retrieved of a result x
Output: $P$: Semantically distinct properties of an LD source
        $N$: Number of the First Results
x = 0
n_semanticDP = 0
n_previousSemanticDP = 0

do
    if $x$==0 then
        $P$.addProperties($PR_0$)
        n_semanticDP = size(P)
    else
        n_semanticDP = size(P)
        $P$.AddAttributes(SemanticDistinction($PR_x$, $P$))
        n_semanticDP = size(P)
    end if
    $N$++

while n_semanticDP $\neq$ n_previousSemanticDP
```

Algorithm 7.3 illustrates the extraction of semantically distinct properties in Linked Data source. N is the number of the first results required for (nearly) all semantically distinct to be retrieved which varies according to the Linked Data source considered, as Section 5.3.1 shows (where N=500 is when the number of semantically distinct properties started to stabilise). `n_semanticDP` and `n_previousSemanticDP` refers to ”the number of semantically distinct properties found” and ”the number of semantically distinct properties found until the previous result”. Thus the process of the internal extraction of properties does not terminate until these two variables are equal, which means that the number of semantically distinct properties virtually stabilised.

7.2.5.3 Global Schema Creation

The global schema is formed by extracting all the semantically distinct properties of all the sources considered. This will force a semantic overlap between the global schema and all the sources considered. It is designed to make the properties of every source semantically a subset of the properties of global schema, as demonstrated in Algorithm 7.4. As indicated in Section 5.2, the global schema is created when the system is first developed, and it is updated incre-

Algorithm 7.4 The Creation of the global schema [Author, 2017]

```
Input: $S$: DataSources
       $i$: Number of Data Sources
       $ps_i$: Properties of a Source $i$
Output: $G$: GlobalSchema

while $S$.hasNext() do
    if $S_0$ (The first Source) then
        $ps_0$ = extractProperties($S_0$)
        $G$.addAttributes($ps_0$)
    else
        $G$.AddAttributes(SemanticDistinction($ps_i$, $G$) )
    end if
    $i$++;
end while
```

mentally on a time-lapse basis to verify whether a new source has been added or the structure of existing sources is modified or extended. Whenever a new set of semantically distinct properties is added or updated, it is semantically compared with the existing distinct properties the global schema already contains. Thus, only the new properties with a unique meaning and no counterparts inside the global schema are added.

7.2.6 Interlinking Subsystem

Having the global schema created with a guarantee of a semantic overlap with all the participant sources, the interlinking system matches semantically the properties of the retrieved results with it. Data interlinking is a technique that discovers the counterparts of the same real world object that may be situated in the same or in a different data source [Nguyen *et al.*, 2012b]. In the context of this chapter, however, it is used as a complementary tool to the integration approach. Unlike LinkD objective presented in Chapter 6, which is to establish identity links between resource, the main role of this adapted LinkD version presented in this chapter is to populate the global schema created using SimiMatch. It re-matches the properties of the properties of the different sources with the global schema. Then it utilises these matches to transfer the values of the sources properties to their counterparts in the global schema.

The flowchart in Figure 7.3 describes the interlinking module proposed. It can be seen that the dataset goes through many stages before the properties matches are identified and the content is re-allocated from the sources to the global schema. These stages can be organised and grouped in three main phases as follows:

7.2.6.1 Preparing the datasets

Section (A) in the flowchart in Figure 7.3 corresponds to the query engine distributor (Section 7.2.4). It highlights that the subsystem starts when the results are retrieved from all sources. A pre-processing step (see Section B in Figure 7.3) is then performed to extract the label of the resources (in this case the predicate) that are described using URIs, which is its last part according to Linked Data principles. The label of the predicate (or the property) is tokenised in order to optimise the measurement of semantic similarity in the next stages (see Section 6.2.2.1). The properties label with only one character are excluded, as no semantics can be derived from them.

7.2.6.2 Properties Matching

The distance between the properties of the source dataset and the target dataset is measured by calculating the semantic similarity of their labels (see Section C in Figure 7.3), using the semantic text similarity system: UMBC EBIQUITY-CORE [Han *et al.*, 2013]. UMBC concentrates on the semantics of the word but not its lexical category, which makes it a typical similarity measurement mean for our system, since the available vocabularies for describing vary between nouns and verbs. It also provides a Web API whereby external systems can retrieve the similarity between two texts without the necessity of going through the re-implementation of the approach (an example is presented in Section 3.7.4.2).

The UMBC similarity tool is implemented in SemiLD to eliminate the time to connect to, send and receive information from their API every time a similarity matching is needed. This also helps in evaluating the genuine performance of the system.

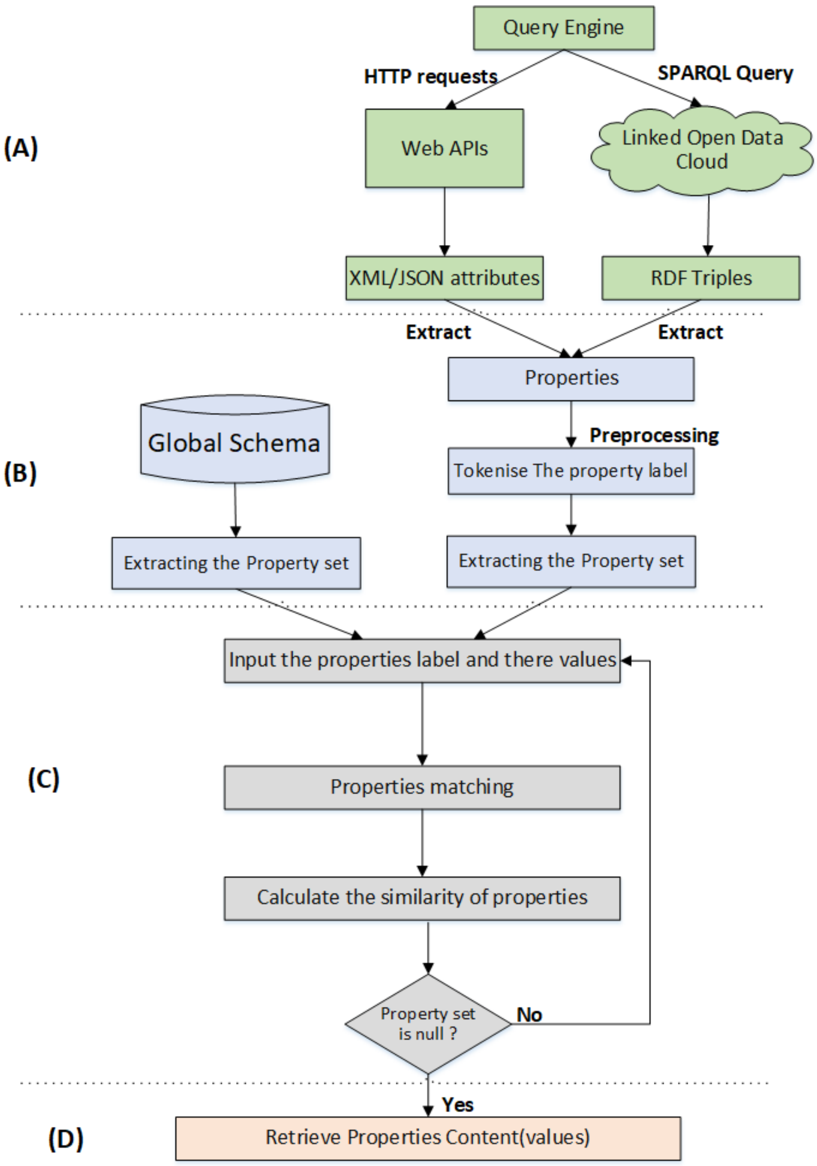


Figure 7.3: Flowchart of the interlinking module and the Query Engine Distributor [Author, 2017].

7.2.6.3 Instances integration

This is the final stage of the integration process. Having the list of the matched properties between the global schema and each of the target datasets, the system extracts their content (instance) (see Section D in Figure 7.3). The instances are then transferred to their counterpart in the global schema using the generated mapping rules.

7.3 Prototype 1 Evaluation

The first prototype is a java Web based data-centric keyword search system (more information about the implementation environment can be found in Appendix III). It is an extension of the implementation of SimiMatch. The same technologies employed in SimiMatch are used to impelement the other modules (Java, Jena, XML, JSON, UMBC EBIQUITY-CORE etc.). XML is the format used to represent the global schema due to its effectiveness and popularity in information exchange along with the simplicity and the availability of the tools manipulating this data language.

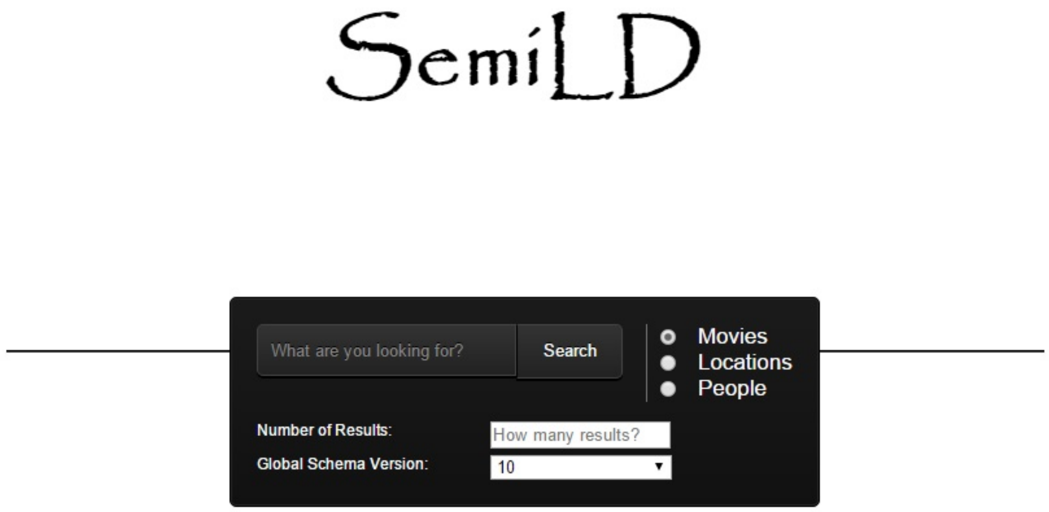
The user in this SemiLD prototype searches for information about movies, people and locations in ten heterogeneous sources, being:

- Four Linked Data sources: DBpedia (movies, locations and people), LinkedMDB (movies and people), LinkedGeoData (locations), Geonames (locations);
- Six Semi-structured sources: OMDB (movies), TMDB (movies), GoogleMapsAPI (locations), GooglePlusAPI [46] (people), IMDB[47] (people) and Last.fm (people).

Figure 7.4 shows the home page of the proposed SemiLD prototype. For the purpose of the experiment, the fields ”number of results” and ”global schema version” as well as radio buttons to select the domains have been added. The user in SemiLD can enter one or multiple keywords.

[46] https://developers.google.com/+/web/api/rest/

[47] http://www.imdb.com/xml/[parameters]



Number of Results: How many results?

Global Schema Version: 10

Figure 7.4: Home page of SemiLD prototype [Author, 2017].

7.3.1 Movies Search

This section presents details about a search session for movies. Figure 7.5 shows the loading page from the four movies datasets considered. They are designed to inform the user in real time what data source SemiLD is processing. Figure 7.6 illustrates the filtering service of SemiLD that is designed to offer more experssivity to the search without affecting the system's usability. Finally, Figures 7.7 and 7.8 depict the presentation of the results. As it can be seen, various information are provided to the user including the original data source.

7.3.2 Locations Search

The presentation of the results of locations in SemiLD differs from presenting the information on movies. The results in this domain are also projected onto a Google map using googleMap API. Figure 7.9, 7.10 and 7.11 show various presentations of location search results.

Figure 7.5: Loading pages of movies search in SemiLD [Author, 2017].

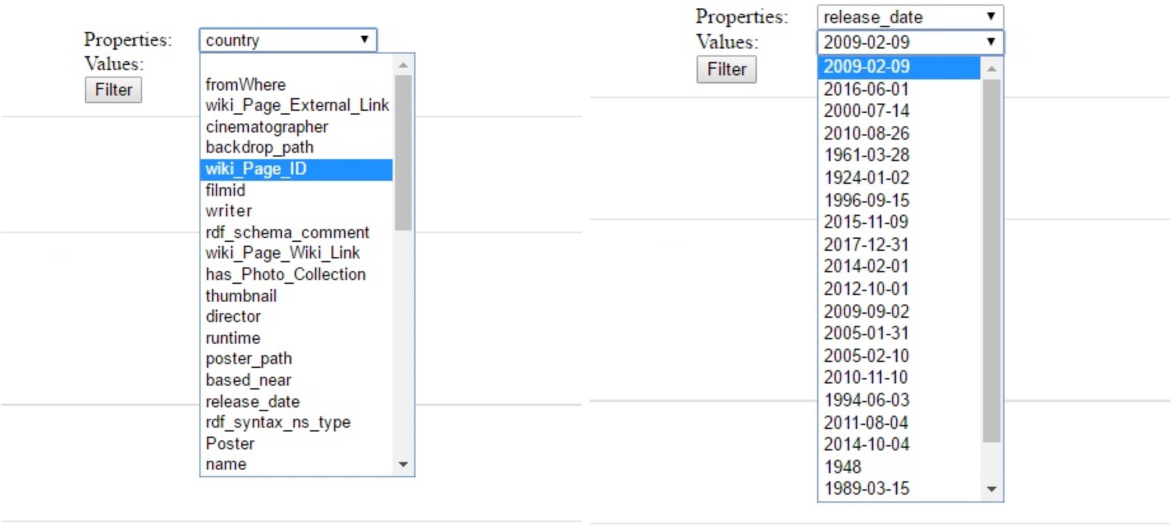
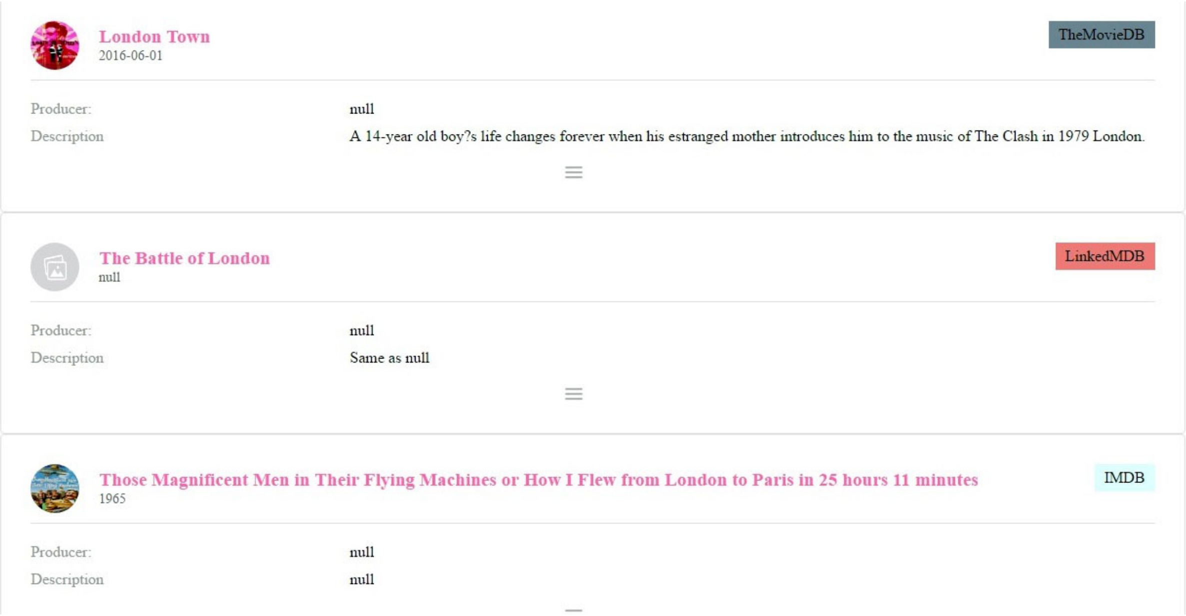


Figure 7.6: Filtering feature in SemiLD prototype [Author, 2017].



London Town
2016-06-01

TheMovieDB

Producer: null
Description A 14-year old boy?s life changes forever when his estranged mother introduces him to the music of The Clash in 1979 London.

The Battle of London
null

LinkedMDB

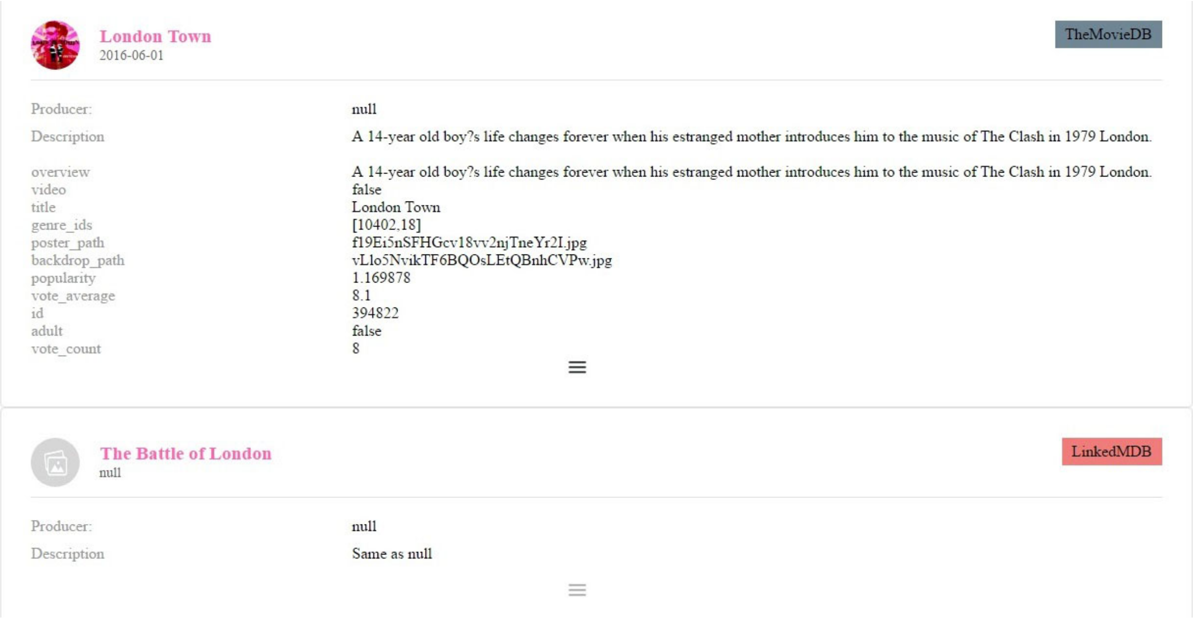
Producer: null
Description Same as null

Those Magnificent Men in Their Flying Machines or How I Flew from London to Paris in 25 hours 11 minutes
1965

IMDB

Producer: null
Description null

Figure 7.7: Movies results presentation [Author, 2017].



London Town
2016-06-01

TheMovieDB

Producer: null
Description A 14-year old boy?s life changes forever when his estranged mother introduces him to the music of The Clash in 1979 London.

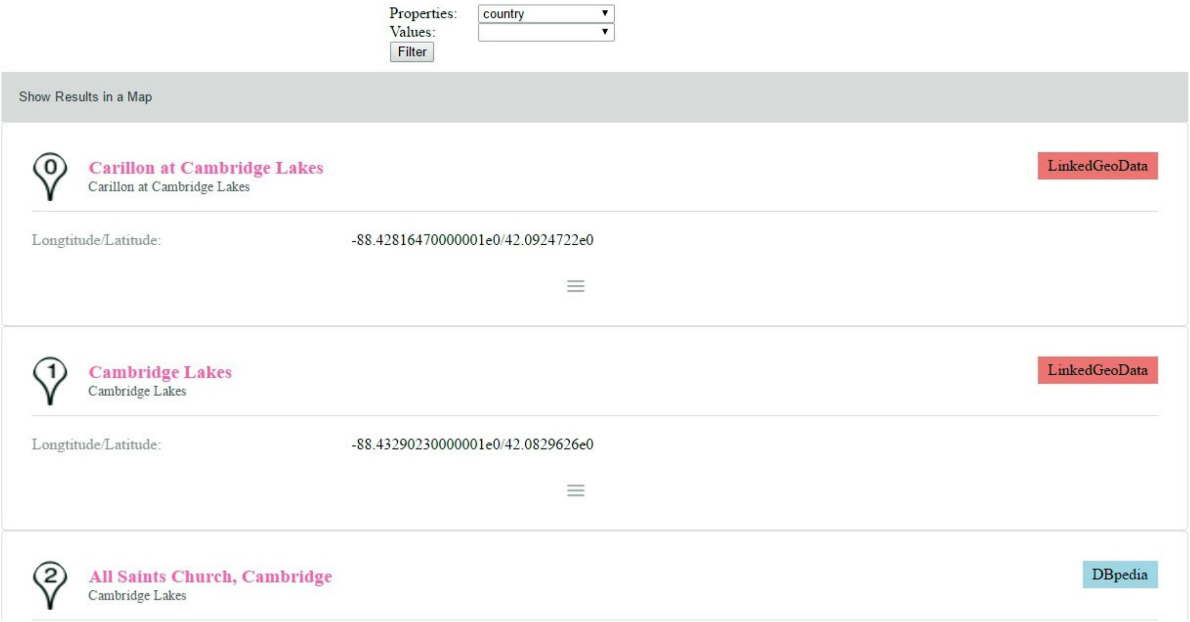
overview	A 14-year old boy?s life changes forever when his estranged mother introduces him to the music of The Clash in 1979 London.
video	false
title	London Town
genre_ids	[10402,18]
poster_path	f19Ei5nSFHGcv18vv2njTneYr2I.jpg
backdrop_path	vLlo5NvikTF6BQOsLEtQBnhCVPw.jpg
popularity	1.169878
vote_average	8.1
id	394822
adult	false
vote_count	8

The Battle of London
null

LinkedMDB

Producer: null
Description Same as null

Figure 7.8: Movies description presentation [Author, 2017].



Properties: country ▾

Values: ▾

Filter

Show Results in a Map

0 **Carillon at Cambridge Lakes** LinkedGeoData

Carillon at Cambridge Lakes

Longtitude/Latitude: -88.42816470000001e0/42.0924722e0

1 **Cambridge Lakes** LinkedGeoData

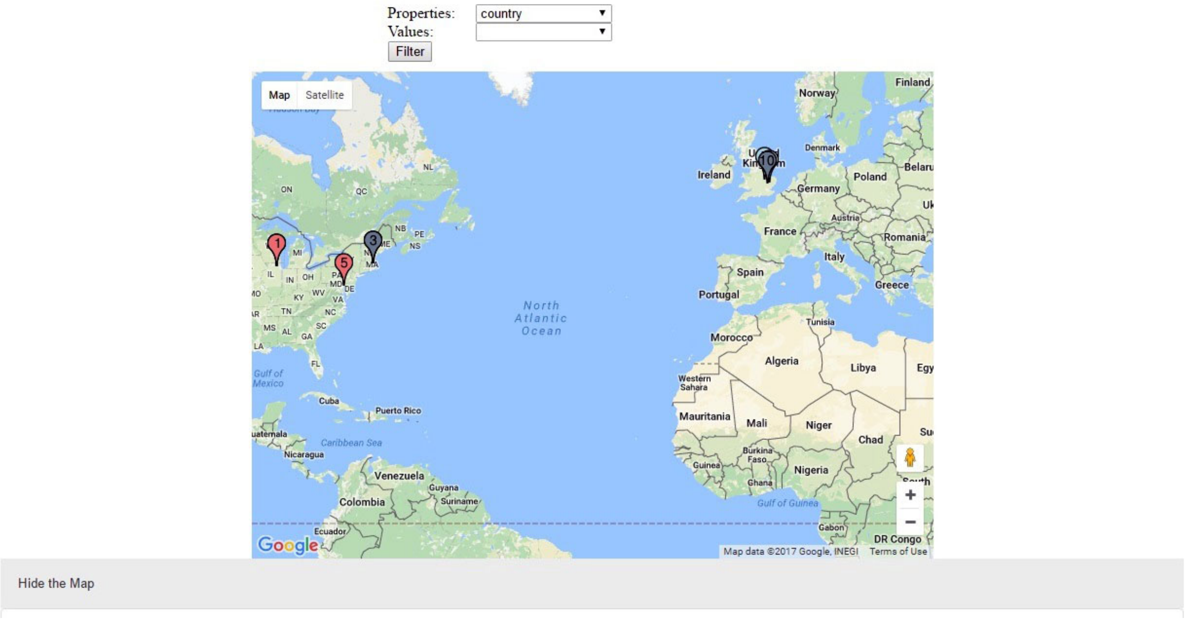
Cambridge Lakes

Longtitude/Latitude: -88.43290230000001e0/42.0829626e0

2 **All Saints Church, Cambridge** DBpedia

Cambridge Lakes

Figure 7.9: Locations results presentation [Author, 2017].



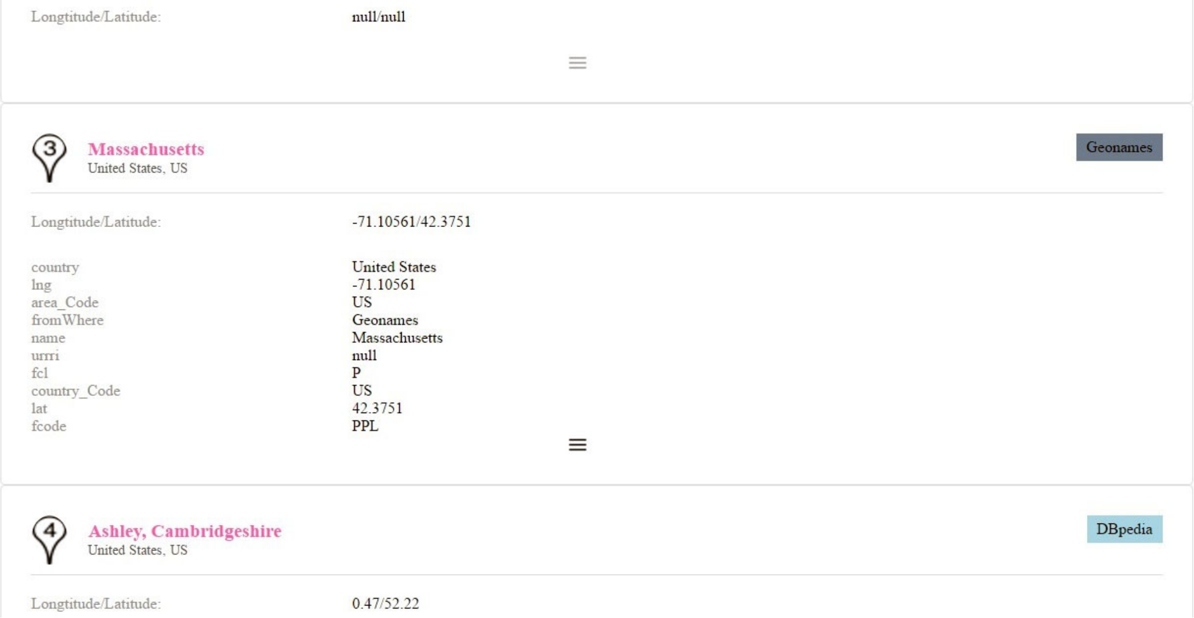
Properties: country ▾

Values: ▾

Filter

Hide the Map

Figure 7.10: Locations results in a Google map presentation [Author, 2017].



Longtitude/Latitude:	null/null

Massachusetts
United States, US

Geonames

Longtitude/Latitude:	-71.10561/42.3751
country	United States
lng	-71.10561
area_Code	US
fromWhere	Geonames
name	Massachusetts
urrri	null
fcl	P
country_Code	US
lat	42.3751
fcode	PPL

Ashley, Cambridgeshire
United States, US

DBpedia

Longtitude/Latitude:	0.47/52.22

Figure 7.11: Locations results description [Author, 2017].

7.3.3 People Search

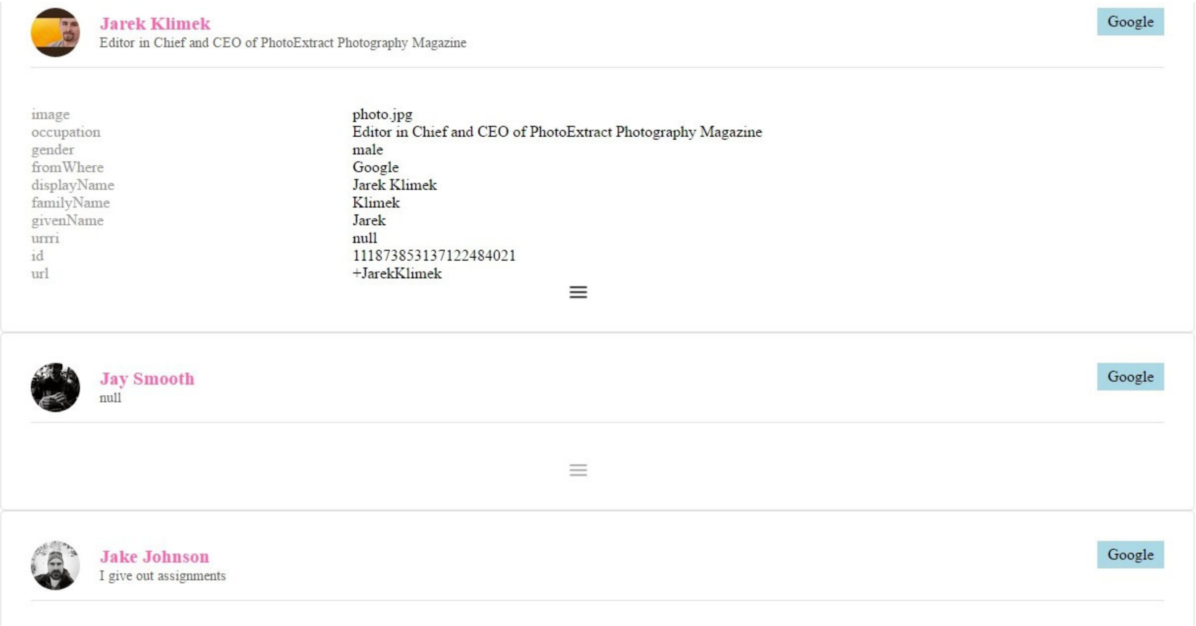
Similarly to locations search, the presentation of the results of people in SemiLD is slightly different from presenting the information of movies, as Figure 7.12 illustrates.

7.3.4 Formative Evaluation

To measure the quality of SemiLD and to allow direct comparison with related systems, this prototype is evaluated using the method proposed by Xu and Mease [2009]. The evaluation is based on calculating the completion time of three tasks that 10 users had to carry out. The number of participant is chosen to be the same as FuhSen. This was (at the time of writing this thesis) the only related systems that is evaluated, using a clear methodology that presents numerical evaluation results enabling precise comparison.

The tasks formulated to evaluate SemiLD (SemiLD's evaluation sheet can be found in Appendix II) are:

1. find "the director of a movie called `The Best` that was released in 1998".



Jarek Klimek
Editor in Chief and CEO of PhotoExtract Photography Magazine

image	photo.jpg
occupation	Editor in Chief and CEO of PhotoExtract Photography Magazine
gender	male
fromWhere	Google
displayName	Jarek Klimek
familyName	Klimek
givenName	Jarek
urrri	null
id	111873853137122484021
url	+JarekKlimek

Jay Smooth
null

Jake Johnson
I give out assignments

Figure 7.12: People results presentation [Author, 2017].

2. find the latitude and longitude of Alexandria with the country code 256.

3. (a): find a person `James Smith Garcia`, who is 30 years old and works as an English Teacher. (b): find yourself.

In the first task, participants are asked to search for the director name a movie called `The best`. This movie was selected to highlight the difference between a data-centric search and document-centric search engine.

The second task points out the advantage of all properties-values filtering offered in the SemiLD prototype since there are many cities and locations Alexandria over the world.

The last task (including the two subtasks) is to draw a comparison with FuhSen where they used something similar to evaluate their criminal investigation system.

The users were instructed to stop when they considered that they had invested sufficient effort to accomplish the task or when they find a result that they consider correct.

Five participants tried to complete the tasks using conventional search tools such as Yahoo, Google or Bing, and the other five utilised this SemiLD prototype. For the purpose of the experiment, the participants who did not have a Google+ account were asked to create one

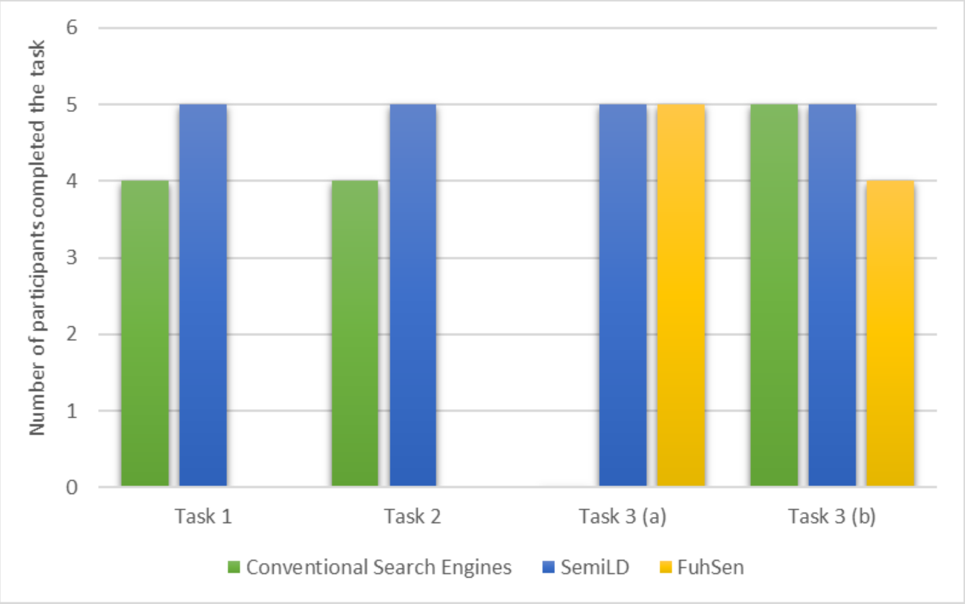


Figure 7.13: Users task completion rate [Author, 2017].

temporarily. Figure 7.13 shows the task completion success rate. It can be observed that no participant was able to accomplish Task 3 (a) using a conventional search engine. In contrast, with SemiLD prototype, all users were able to find `James Smith Garcia`. The explanation for these figures can be related to the difference in terms of data freshness between Web APIs search and document-centric search. Universal (conventional) search engines takes time to index a Web page; whereas, Web APIs interacts directly with the database or the data source hence the results are instantly available after publication. The second explanation can be the ranking algorithms and the way they prioritise a result over another, which does not always favour what the user is looking for. For the rest of the tasks, SemiLD performed similarly or slightly better than the FuhSen or the other search engines.

Figure 7.14 shows the time each participants needed to complete the tasks using SemiLD comparing to FuhSen and the other conventional search engines. The participants in the evaluation of SemiLD in Task 3 were not the same as those for FuhSen. This part of the diagram is rather a rough comparison of how the search times of the two systems compare. Task 1 and 2 do not belong to the crime investigation domain, so they are not part of FuhSen scope.

In general, it took all participants less than 2 minutes, 1.5 minutes and 1 minute to accomplish the first, second, and the third tasks respectively. As it can be seen, the participants completed Task 1 faster using SemiLD. Using conventional search engines, the results were

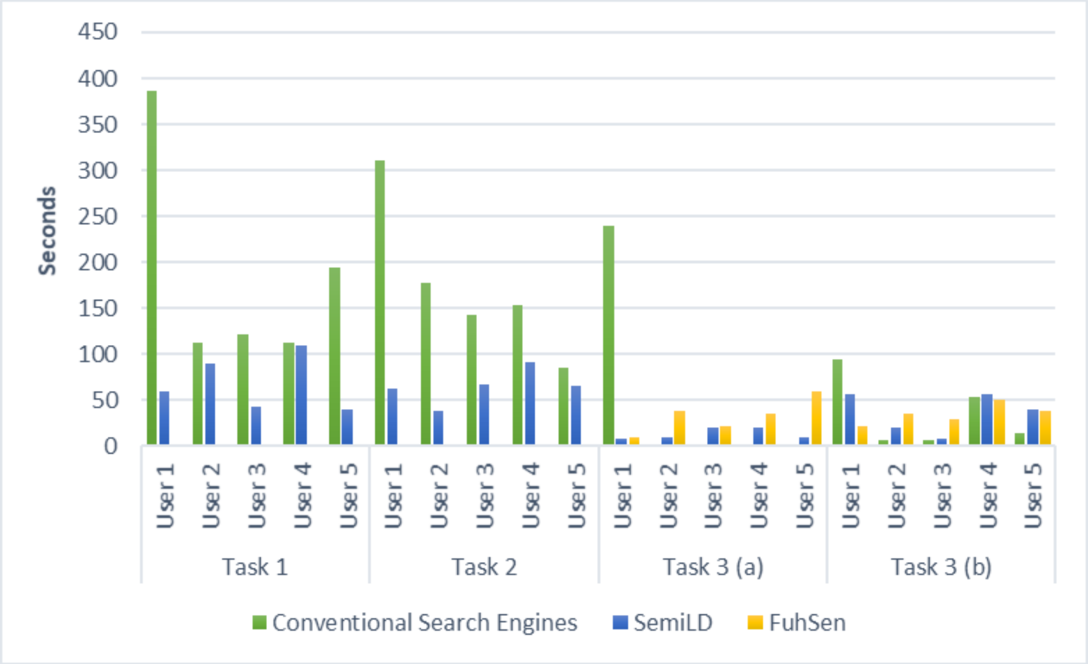


Figure 7.14: Task completion duration [Author, 2017].

showing the best movies in 1998 rather than a movie called `The best` made in 1998. In Task 2, the duration of the search is significantly less in SemiLD due to all properties-values filtering. Searching for Alexandria 256 in a universal search engine will not necessarily output Alexandria the city with the country code 256. Many interpretation might be made depending on the number occurrences of these words in the indexed documents, their popularity and other factors. Search results examples of the participants encountered were: Arius of Alexandria (256-336 AD), Alexandria with the calling code 256, books where Alexandria is mentioned in the 256th page etc. Task 3 shows that the figures of SemiLD are comparable to those of FuhSen with the slight advantage of the latter in Task 2. These figures demonstrates that even though SemiLD is designed to be generic, its performance can be considered similar to FuhSen's performance in its own domain.

7.3.5 Evaluation of Usability

This evaluation was carried out with all participants (those who used conventional search engines were asked to test SemiLD at the end and give their feedback). Similarly to FuhSen, two techniques were used during this evaluation: Think aloud protocol and a Post-Study System

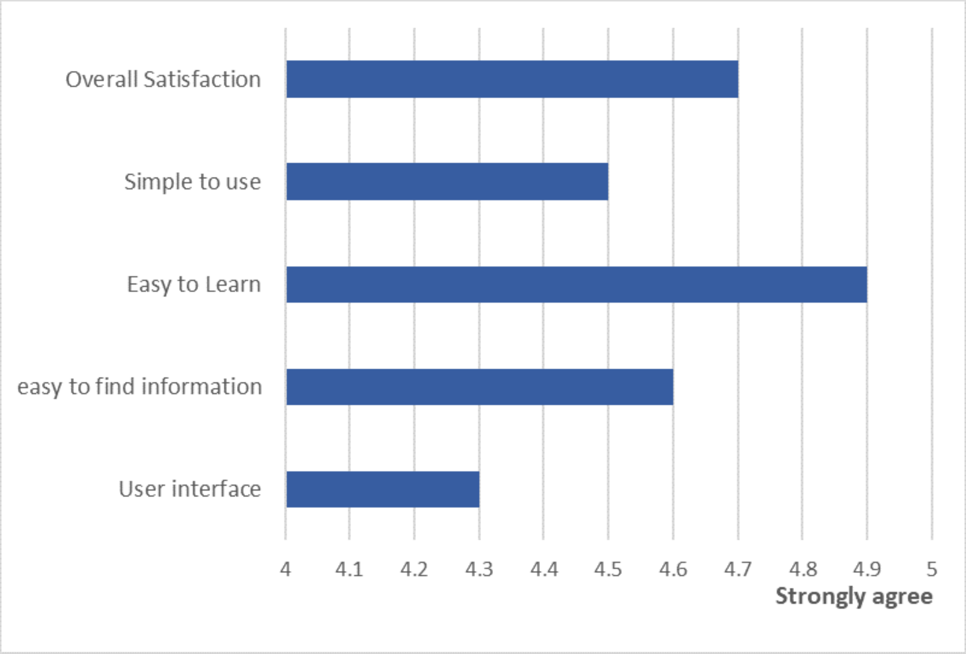


Figure 7.15: PSSUQ results: Summary of subjective user feedback on SemiLD [Author, 2017].

Usability Questionnaire (PSSUQ) [Lewis, 1995]. Figure 7.15 summarises the results. SemiLD received good scores in all aspects, which indicates the good interaction design decisions implemented in the prototype. Some of the users indicated ways to improve the usability during the experiment, for example: ordering the content of the filter select box alphabetically and removing ”number of results” and the ”global schema version” text fields that were added for the purpose of the experiment. These comments should be taken into consideration for further improvements of the user interface and experience.

7.4 Prototype 2 Evaluation

This section presents another example test scenario of the implementation of the proposed architecture. SemiLD is implemented into a keyword search in a collection manager for movies. This prototype is implemented using Java and Jena libraries, along with other several tools to parse JSON and XML files.

The user searches for information about movies in four heterogeneous sources, being:

- Two Linked Data sources: DBpedia, LinkedMDB;

- Two Semi-structured sources: OMDB, TMDB.

The system then fetches all the available information and displays them in an interactive interface. Many graphical forms and features that are adapted to the context of this prototype are provided as part of the usable experience to hide complexity.

7.4.1 The Structuring of the Global Schema

The aim of this section is to show the significant difference between the number of the properties that are syntactically un-similar and semantically different, as well as identify the N number for this implementation. The N number, discussed in Section 7.2.5, represents the number of the first results needed to extract all the semantically distinct properties. First, a query is run to count the syntactically distinct properties on the SPARQL on DBpedia endpoint, which is a multi-domain Linked Data source that uses various and heterogeneous vocabularies in describing datasets in the same domain. Then, the semantic distance is measured between these properties in order to extract the semantically distinct properties and count them. The N number is identified when the number of semantically distinct properties becomes steady.

Example 7.1 is a SPARQL query that counts (and retrieves by removing count) the syntactically distinct properties that the first 200 movie datasets contains. It is an adaptable query for all Linked Data endpoints supporting subquery feature (SPARQL 1.1), where only the vocabularies used change. In this example, it is expressed to work on the DBpedia endpoint.

Example 7.1: A SPARQL query to count syntactically distinct properties

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT COUNT(DISTINCT ?p) WHERE {
    ?s ?p ?o .
    FILTER (?s = ?film)
    {
        SELECT ?film {
        ?film rdf:type <http://schema.org/Movie> .
        }
    limit 100
    }
}
```

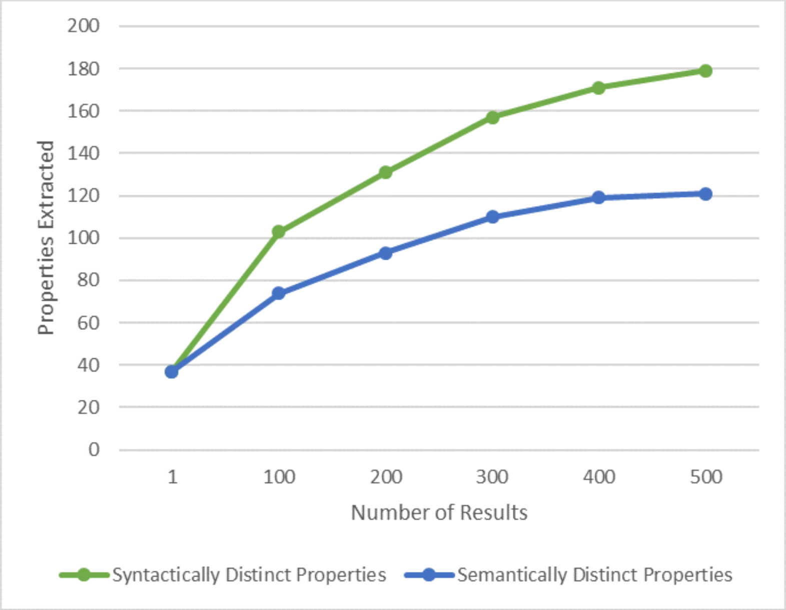
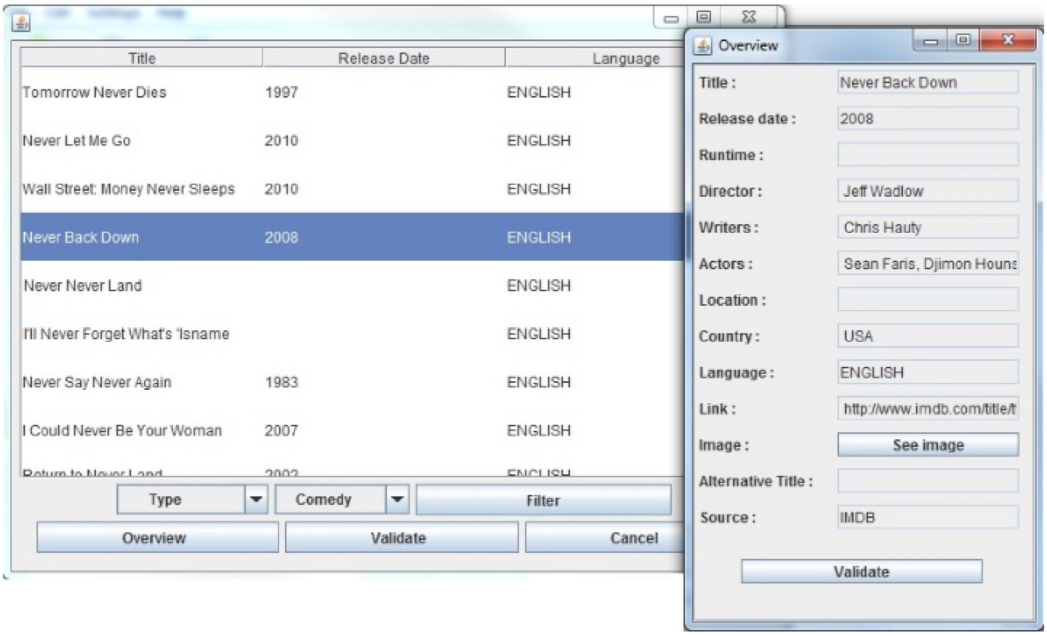


Figure 7.16: The discrepancy between the syntaxes and semantic unsimilarity [Author, 2017].

For other Linked Data endpoints that have not updated and were still running SPARQL 1.0, such as LinkedMDB (at the time this thesis is written), the system uses Jena framework to nest the results of a query within another query. The version of the SPARQL endpoint is included in the Metadata to decide automatically which of the two predefined means will be used.

After applying Algorithm 7.4 (see Section 7.2.6) to extract and count the number of the semantically distinct properties, the line chart in Figure 7.16 illustrates discrepancy between the numbers of the syntactically and semantically distinct properties according to the number of the results retrieved (similar to results in Section 5.3).

The structure of the global schema is created when the system is first developed, and it is updated in the background on a time lapse basis, similar to a ”cron job” (a scheduled process). Since the time of the creation of the global schema is not part of the response time, the N number can be set to a maximum and a ”safer” value that ensure all the properties are recalled from all sources. Thus, it does not affect the adaptivity nor the degree of the automation, as it is does not run every time the system is queried, and nor does it need to be changed when a new Linked Data source is added.



Title	Release Date	Language
Tomorrow Never Dies	1997	ENGLISH
Never Let Me Go	2010	ENGLISH
Wall Street: Money Never Sleeps	2010	ENGLISH
Never Back Down	2008	ENGLISH
Never Never Land		ENGLISH
I'll Never Forget What's 'Isname		ENGLISH
Never Say Never Again	1983	ENGLISH
I Could Never Be Your Woman	2007	ENGLISH
Return to Never Land	2002	ENGLISH

Type: Comedy

Overview

Title :	Never Back Down
Release date :	2008
Runtime :	
Director :	Jeff Wadlow
Writers :	Chris Hauty
Actors :	Sean Faris, Djimon Houns
Location :	
Country :	USA
Language :	ENGLISH
Link :	http://www.imdb.com/title/t
Image :	See image
Alternative Title :	
Source :	IMDB

Figure 7.17: Results presentation and the filtering service [Author, 2017].

7.4.2 Prototype 2 User Interface

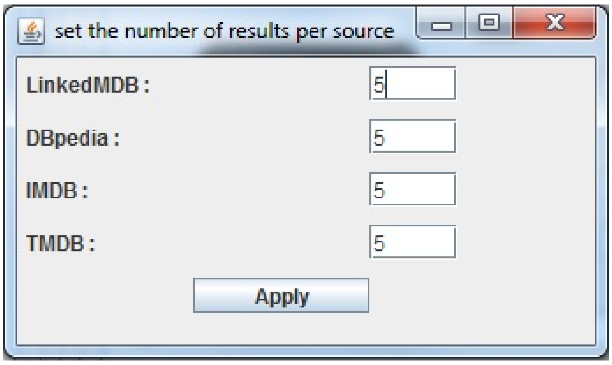
Figure 7.17 shows the dynamic and interactive feature that allows the users to filter the results according to the properties of the global schema, along with the possible values extracted from the sources. These features are essential in this keyword search to rectify their lack of expressivity without affecting the usability, by assisting the users in finding the requested result. The system also keeps track of the originated source of every result.

7.4.3 Prototype 2 Metadata

This is the only module in the system that is predefined manually. This repository indicates the links to the SPARQL endpoints or Web APIs of the sources to be queried (see Table 7.1). In addition, the metadata gives the users the possibility to choose the number of the results desired from each of the sources considered, accessed through an interface (see Figure 7.18). It is also important, as discussed in Section 7.2.3, to determine whether the version of SPARQL supported in the endpoint is SPARQL 1.1 or lower.

The Source	API URL / SPARQL Endpoint	Results	SPARQL 1.1
DBpedia	http://dbpedia.org/sparql	4	1
LinkedMDB	http://linkedmdb.org/sparql	4	0
IMDB	http://api.themoviedb.org/3/search/movie	3	N/A
OMDB	http://www.omdbapi.com/	3	N/A

Table 7.1: Example of a Metadata repository [Author, 2017].



set the number of results per source

LinkedMDB : 5

DBpedia : 5

IMDB : 5

TMDB : 5

Figure 7.18: The interface to change the number of the results per source [Author, 2017].

7.4.4 Evaluation of the System

As part of the evaluation, the system is queried and tested using a common keyword `best`. Example 7.2 is a SPARQL query to search a keyword in a Linked Data source.

Example 7.2: A SPARQL query to search using a keyword

```
PREFIX mdb: <PATH/data.linkedmdb.org/resource/movie/>
PREFIX rdfs: <PATH/w3.org/2000/01/rdf-schema#>
PREFIX dc: <PATH/purl.org/dc/terms/>
SELECT * WHERE {
    ?title dc:title ?keyword.
    filter(REGEX(?keyword, ’’best’’,’’i’’))
}
LIMIT 4
```

SPARQL endpoints offer various formats in expressing the results. To point out the heterogeneity, RDF is the format outputted from Linked Data sources, and JSON and XML are the formats expressing the results of the semi-structured sources.

S1, S2, S3, S4 in Tables 7.2 and 7.3 refer to DBPedia, LinkedMDB, TMDB, OMDB sources respectively

Keyword	Syntactically Distinct Properties				Semantically Distinct Properties				Global Schema
	S1	S2	S3	S4	S1	S2	S3	S4	
100	103	36	20	13	74	36	20	13	112
200	131	41	20	13	93	41	20	13	136
300	157	43	20	13	110	43	20	13	153
400	171	47	20	13	119	45	20	13	164
500	179	47	20	13	121	45	20	13	164

Table 7.2: The Number of the Semantically Distinct Properties Extracted per Source [Author, 2017].

Keyword	Results Requested per source	Results available					Properties retrieved				Properties matched with Global Schema				Overall Precision
		S1	S2	S3	S4	overall	S1	S2	S3	S4	S1	S2	S3	S4	
in	50	50	50	50	50	200	56	36	20	13	56	36	20	13	1
in	200	200	200	200	200	800	94	41	20	13	85	41	20	13	0.97
best	50	50	5	50	50	152	59	21	20	13	57	21	20	13	0.99
best	200	124	5	200	200	529	83	21	20	13	71	21	20	13	0.96
London	200	85	4	200	200	489	79	22	20	13	72	22	20	13	0.97
Steve	200	15	0	200	200	415	57	-	20	13	56	-	20	13	0.99

Table 7.3: Matching Precision of SemiLD prototype 2 [Author, 2017].

Table 7.2 evaluates the process of the creation of the global schema according to the number of first results considered. It shows that DBpedia has a considerable semantic overlap between the properties of the vocabularies describing its datasets. More importantly, there is a noticeable overlap with the properties of the four sources considered. For example, for N= 500, the system extracted 199 semantically distinct properties from all sources; however, 35 were deleted as they have the same semantics as some of 164 properties previously retrieved. It can be seen that DBpedia and LinkedMDB are more general than the rest of semi-structured sources. In this case, they contain all the properties available in OMDB and TMDB.

Table 7.3 suggests a way to estimate the precision of the system. The keywords are ordered from the most to the least general. The general keywords generate the maximum number of results; whereas the specific words do not occur in many movies so less results exist. The precision is calculated by dividing the properties matched (the number of mapping rules generates) on the semantically distinct properties retrieved. The global schema generated from processing the 500 first results is the one utilised in this Table. Table 7.4 is an example of a portion of the results retrieved. The predicates runtime and director are included to show the differences

Title	Runtime	Director	Year	Source
My Best Friend's Wedding	105	P. J. Hogan	1997	LinkedMDB
The Best Man	102	Franklin Schaffner	1964	LinkedMDB
My Best Friend's Birthday	-	Quentin Tarantino	1987	LinkedMDB
The Best of Insomniac	-	Nick McKinney	2003	LinkedMDB
O Despertar da Besta	-	Jose Mojica Marins	1983	DBpedia
Best Wishes for Tomorrow	110	Takashi_Koizumi	2008	DBpedia
Best Player	98	Damon Santostefano	2011	DBpedia
The Best Exotic Marigold Hotel	-	-	2011	OMDB
The Best Offer	-	-	2013	OMDB
Best	-	Mary McGuckian	2000	TMDB
Best of the Best	-	Andrew Lau Wai-Keung	1996	TMDB
Best of the Best	-	Herman Yau	1992	TMDB
...				

Table 7.4: Example of some of the results retrieved by running a search using the keyword ”best” [Author, 2017].

	Approach	Query expressiveness	Adaptivity	Up-to-date Data	Generic	Semantic
Google	Federated	Keywords, NL	Yes	Yes	Yes	No
PowerAqua	Centralised	NL	No	Yes	Yes	Yes
SWIM	Centralised	SPARQL/RQL	No	No	Yes	Yes
LSM	Centralised	SPARQL/CQELS	No	No	No	Yes
MOMIS	Federated	n/a	No	n/a	Yes	Yes
FuhSen	Federated	Keywords	No	Yes	No	Yes
SemiLD	Centralised	Keywords	Yes	Yes	Yes	Yes

Table 7.5: Theoretical comparison of SemiLD against related systems [Author, 2017].

between the sources and the results retrieved in term of the available information.

In Figure 7.19, the diagram presents a comparison between the performances of SemiLD against FuhSen. Although the scenario in FuhSen upon which the line chart is generated is different, the performance included is based on 10 wrappers, which is the optimal that FuhSen can achieve. Table 7.5 presents a theoretical comparison of SemiLD, the system proposed, against the related works. As discussed in Section 3.5, Adaptivity, in this context, refers to the ability to add new sources automatically so as to increase the scale of the amount of data retrieved, and not addressing a subset of the available sources in the targeted data structure(s). In contrast to FuhSen, SemiLD does not utilise any pre-defined vocabulary or language. The global schema is the intermediate that accommodates the resultant datasets. It is constructed automatically according to the sources considered. More importantly, the approach generates mapping rules between the global schema and the sources, that guarantees the integration of the

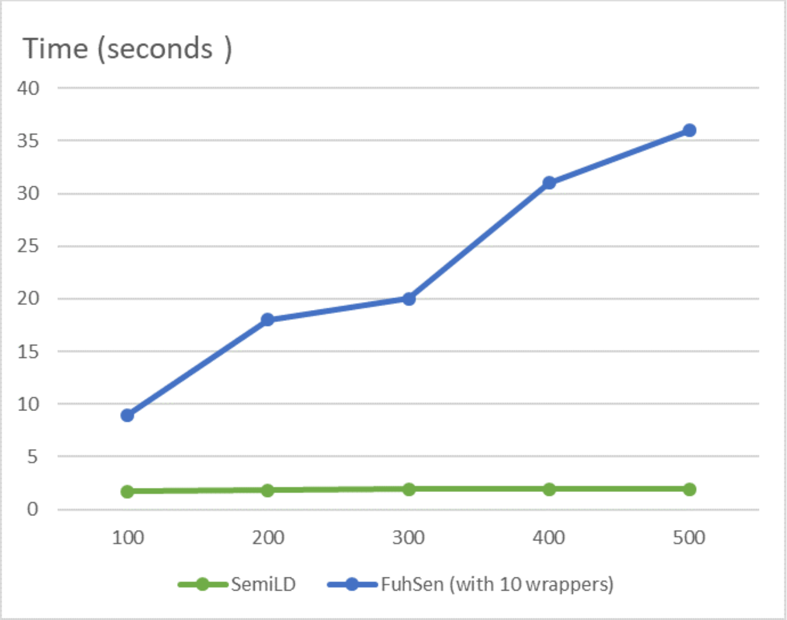


Figure 7.19: Comparison between the performance of SemiLD and FuhSen [Author, 2017].

data. Furthermore, SemiLD has only two classes of adapters, one for semi-structured source and the other for Linked Data sources, which are used to access rather than extract. They are not modelled to conform to the structure of the sources. Instead, for every approach an instance of adapter is created, which receives all the information needed from a metadata file. The latter is the only module that contains preloaded information of the minimum needed about the sources for the system to function.

As far as the present author is aware, FuhSen is the only integration tool that includes a clear evaluation of its performance, though not the precision. None of the few available systems integrating semi-structured with Linked Data sources, presents numerical data that allows a precise comparison. It is challenging ”to make data of different types of benchmarks comparable with each other” due to the lack of a common description or a parameter that can be measured [Pfaff and Krcmar, 2014, p. 1]. Moreover, little has been done to address the current issue even though there are many sources that are still actively outputting semi-structured data with a considerable relevance.

Finally, the concept of privacy is addressed in this approach by excluding relational databases and other tools that may give the users access to non-public data. The system can be only used to search through different Web APIs and SPARQL endpoints, which represent an alternative gateway to available public data.

7.5 Summary

Researchers in the Semantic Web community have been designing tools and architectures to integrate heterogeneous data originated from distributed sources in the last decade. Technologies, such as RDF, resulted from the increased adoption of the Linked Data paradigm, have enabled new data spaces and concept descriptors to define an increasing complex and heterogeneous Web of Data. Other types of data that existed previously, such as semi-structured, still hold a significant value in many areas. To bridge between the two data spaces, this chapter proposed a mediator-based approach that has offered a homogeneous and transparent access to their sources.

The idea was to create a more general global schema in order to force an overlap with the participating sources. It was composed by retrieving all the semantically distinct properties of both Linked Data sources and semi-structured sources (see Section 7.2.5). Then, using the interlinking module (see Section 7.2.6), the matching rules were generated automatically. The data originated from the heterogeneous sources were parsed and re-organised in the global schema (see Section 7.2.5.3) to be finally displayed in an interactive interface for the user (see Sections 7.3.1,7.3.2,7.3.3 and 7.4.2). The contributed approach consists of 5 components, being: a user interface (see Section 7.2.1), an ontology (see Section 7.2.2), sources metadata (see Section 7.2.3), a query engine distributor (see Section 7.2.4) and an interlinking subsystem (see Section 7.2.5). The implementation of this approach (see Sections 7.3 and 7.4) was a keyword search engine, embedded in a Movie Collection Manager for movies, that takes into consideration all the challenges and the criteria stated in this thesis. The results (see Sections 7.3.4, 7.3.5 and 7.4.4) confirmed the performance, precision and usability of the proposed approach.

Chapter 8

Conclusions and Future Work

> *You have your way. I have my way. As for the right way, the correct way, and the only way, it does not exist [in research].*
>
> Friedrich Nietzsche

8.1 Introduction

The main motivation behind this thesis is to provide a bridge between semi-structured and Linked Data, which can be achieved by offering transparent and homogeneous access to both data models, or by contributing to the migration of semi-structured data to the Web of Linked Data. Schema matching is another area that was researched in this work to reconcile the structure of the two data models. This chapter summarises the contribution of this thesis and continues with potential future directions from this line of investigation.

8.2 Summary of Contributions to Knowledge

To fill the gap that exists between semi-structured and Linked Data and to bridge their data sources, this thesis broke down this aim in Chapter 1 into research questions (RQs, see Section

1.3) and their associated objectives. This section refers back to them explaining how this thesis successfully addressed them.

As argued in Section 1.5, the author in this thesis made both theoretical and practical contributions. The author in this work presented the following theoretical contributions:

- **Extending the literature** by offering a review of existing systems according to identified challenges (see Chapter 4). These challenges were extracted after studying the input data, being semi-structured and Linked Data, and analysing various technologies and identifying different challenges associated with their usage (see Chapter 3). This helped the author to have a clear insight about the challenges associated with addressing the specific technologies and operation targeted by the research questions of this thesis.
- Highlighting the challenge of **automatically accommodating the changes of Linked Data sources** in a schema matching approach (see Chapter 5). This challenge was addressed by designing an element-based schema matching approach (see Section 5.2) that has the ability to update its global schema (see Section 5.2.2) automatically on a timelapse basis. The global schema is the output of the schema matching approach, which is a set of the semantically distinct properties of all the considered data sources. The evaluation of this approach (see Section 5.3) showed the correlation between the different parameters that affect precision and recall of the designed approach, which allowed the author to find the optimised settings. Additionally, the evaluation brought to light a view showing the discrepancy between the number of semantically and syntactically distinct properties in Linked Data sources (see Figure 5.6). This high quality measure of the results validates the feasibility of keeping the global schema up-to-date with the changes of the Linked Data sources, which responds to the Research Question 3 (RQ3).
- **Designing a data interlinking approach** that verifies, in the first place, the existence of the URI of the resource being published in the Web of Data in order to establish links with it (see Chapter 7). Unlike other approaches, the interlinking approach does not start with a source set and a target set of resources. The only input is one set of resources (source set) that are not associated with any ontology, vocabulary or a predefined structure. The input is the result of an RDF fixed transformation of a semi-structured dataset

(see Section 2.5.4). Therefore, the designed interlinking approach did not use ontologies or a knowledge base in finding the identity links. Instead, it utilised the domain to allocate dynamic weights that define more the identity of the resources being interlinked (see Section 6.2.2.4). The interlinking approach was based on SERIMI [Araujo *et al.*, 2011], since it is one of the relatively successful approaches that do not use the structure during the interlinking process. The weight allocation is an extension and improvement of what has been proposed in Nath *et al.* [2014]. Other criteria were added in the weight allocation such as the time relatedness of the property. The evaluation (see Section 6.3) showed the effectiveness of the approach in targeting large scale of data that is comparable to related approaches targeting significantly less amount data, which addresses and responds to RQ1.

- **Providing a transparent and data centric access to semi-structured and Linked Data**. The querying, retrieval of results, the integration as well as the adaptation of the global schema when adding new sources (due to embedding SimiMatch) are automatic. The contributed data integration approach (see Section 7.2) comprises of both the designed schema matching and data interlinking approaches that were adapted and embedded as modules. Unlike other state of the art integration approach targeting semi-structured and Linked Data, such as Fuhsen, the data integration approach presented in this thesis is generic and not limited to a specific domain, as highlighted in its evaluation sections (see Sections 7.3 and 7.4). Since the quality of the schema matching and interlinking of the data sources has already been evaluated as part of Chapters 5 and 6, the evaluation in this chapter concentrated on the usability and the expressivity of the user interface of the proposed prototypes (see Sections 7.3.4, 7.3.5 and 7.4.2). The results (see Sections 7.3 and 7.4) respond to Research Question 2 (RQ2) and validates that is feasible for data integration system targeting semi-structured and Linked Data to have a transparent, usable and expressive interface.

The following are the practical contributions that the author made in this thesis:

- Offering three working prototypes: SimiMatch, LinkD and SemiLD (see Sections 5.2, 6.3 and 7.2 respectively).

- Adding the filtering service that searches using every property and every value of each property upon previously heterogeneous data sources in their structure, access method and protocol, model ,etc (see Figure 7.6). It validates the ability of SemiLD to offer a data centric results;

- Adding feature-centric and domain related components. For example, in prototype 1 (see Section 7.3) the results of ”locations” are displayed in a map and the icons are numbered to make the connection between the map and the results; whereas in ”people”, the picture of the person is displayed in the icon and profession in the description.;

- Combining and using both semantic technologies (ontologies to formulate the queries) and similarity measurements techniques (UMBC EBIQUITY-COR [Han *et al.*, 2013] semantic similarity tool) (see Section 7.2);

8.3 Scope and Limitations

The research presented in this thesis is concerned with bridging between semi-structured and Linked Data. This objective involves many tasks in both the considered operations (the integration and the interlinking).

Integrating semi-structured and Linked Data can involve other specificities depending on the context of the implementation. It is beyond the scope of this thesis to address these context-related tasks. For instance, in this work, the context of the implementation is a search system. The thesis, however, does not research or contribute to result optimisation or ranking techniques and approaches.

Similarly, the interlinking operation does not involve digging deep into converting semi-structured to RDF, but rather concentrates on providing an interlinking that supports the characteristics of an RDF resulting from the conversion. The scope of this research is not to publish semi-structured in the Web of Linked Data, but rather to contribute to this overall task by providing identity links between the two data paradigms.

The approaches concentrate on the labels of the properties, which can be encoded and do not signify any meaning. This is why one limitation of the approaches proposed is that the sources need to be parsing and mining friendly. They are suitable for this work as the data considered are originated from Web APIs and SPARQL endpoints, which are frequently designed to be output data that is parse-able or utilise a well-expressed vocabulary. For these approaches to be evolved and improved in the future, a disambiguation stage or module would be needed.

8.4 Future Work and Challenges

This section presents new areas of research and propose some potential projects that can build upon the work presented in this thesis.

8.4.1 Semantic Data Management in Smart Cities

The Smart City approach presented in this section is one example where the approaches proposed in this approach can be combined and applied toghether. Providing a semantic data management approach to Smart Cities allows more depth and rigour when analysing and reasoning upon the available data.

One major problem Smart Cities information and data management systems are facing currently is the heterogeneity, not only of the stream data, but also of the external data sources, such as the Web of Linked Data, the use of which is inevitable in decision making on the scale of a city.

The birth of Smart City and Linked Data initiatives has led to new challenges, but also opportunities, in retrieving and managing data. This project aims at finding and exploiting ways of how they can profit from each other. Publishing Smart City data as Linked Data can expand its available information in Linked Data cloud by creating new entities and establishing new links. Similarly, the more knowledge about a city the systems can recall the more effective are the decisions made. The proposed middleware reconciles stream data, originated from both social and physical sensors related to Smart Cities, with Linked Data in order to offer a transparent access. In this project the authors Kettouch *et al.* [2017b] highlighted and addressed the data

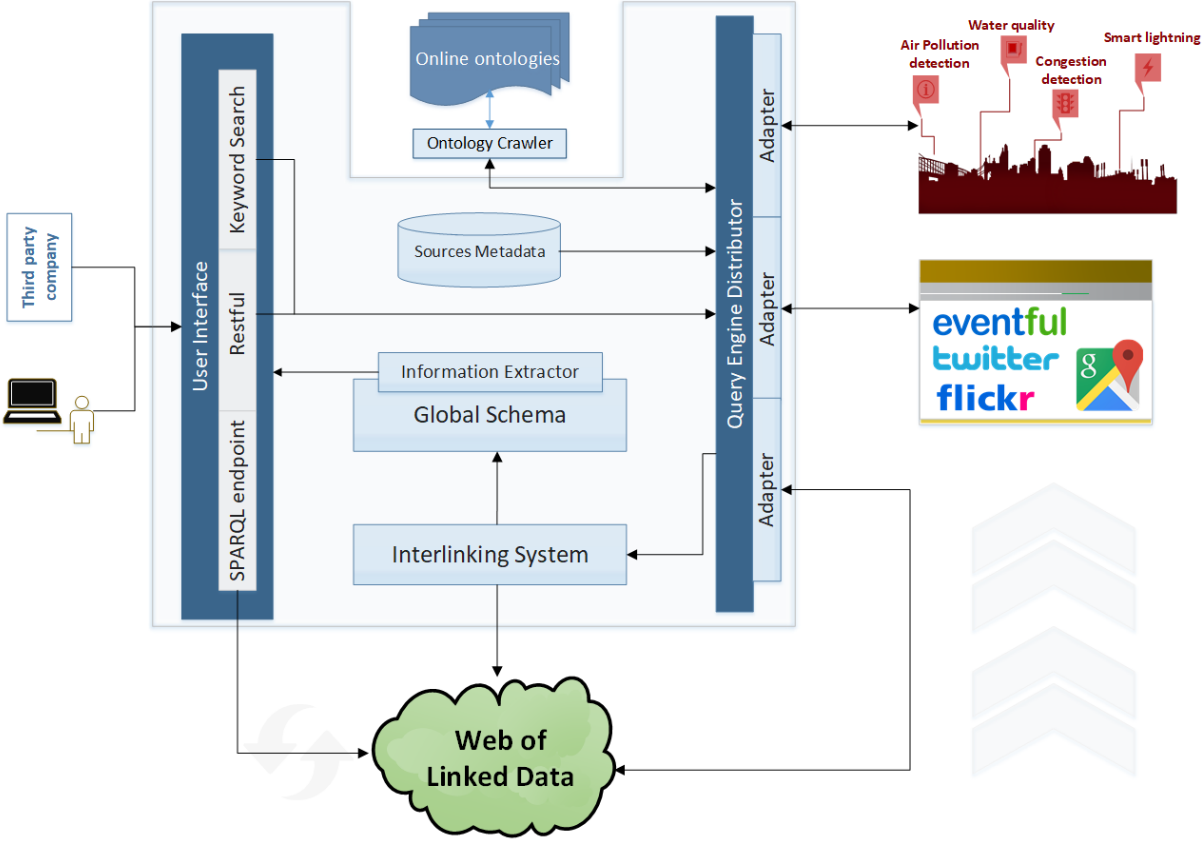


Figure 8.1: The proposed framework for semantic data management of Smart Cities [Kettouch *et al.*, 2017b].

freshness requirements necessary to take into account when working with stream and Linked Data and described the functionality the proposed middleware.

The diagram in Figure 8.1 shows the general architecture of the ongoing project Smart City and its proposed framework. It illustrates the overall role of the new middleware and its modules with respect to the users, who could be managers, planners or simple citizens, and third party companies. Third party companies are distinguished as they can contribute to the Linked Data cloud by processing data originated from the sensors of the Smart City and publishing the outputs. The flowchart in Figure 8.2 describes a potential workflow of the approach.

The challenge of this approach is the creation of a prototype. Hence, further work will be needed to explore ways to simulate and test the entire framework (the modules were only tested independently) and it will also need an accurate method to evaluate it.

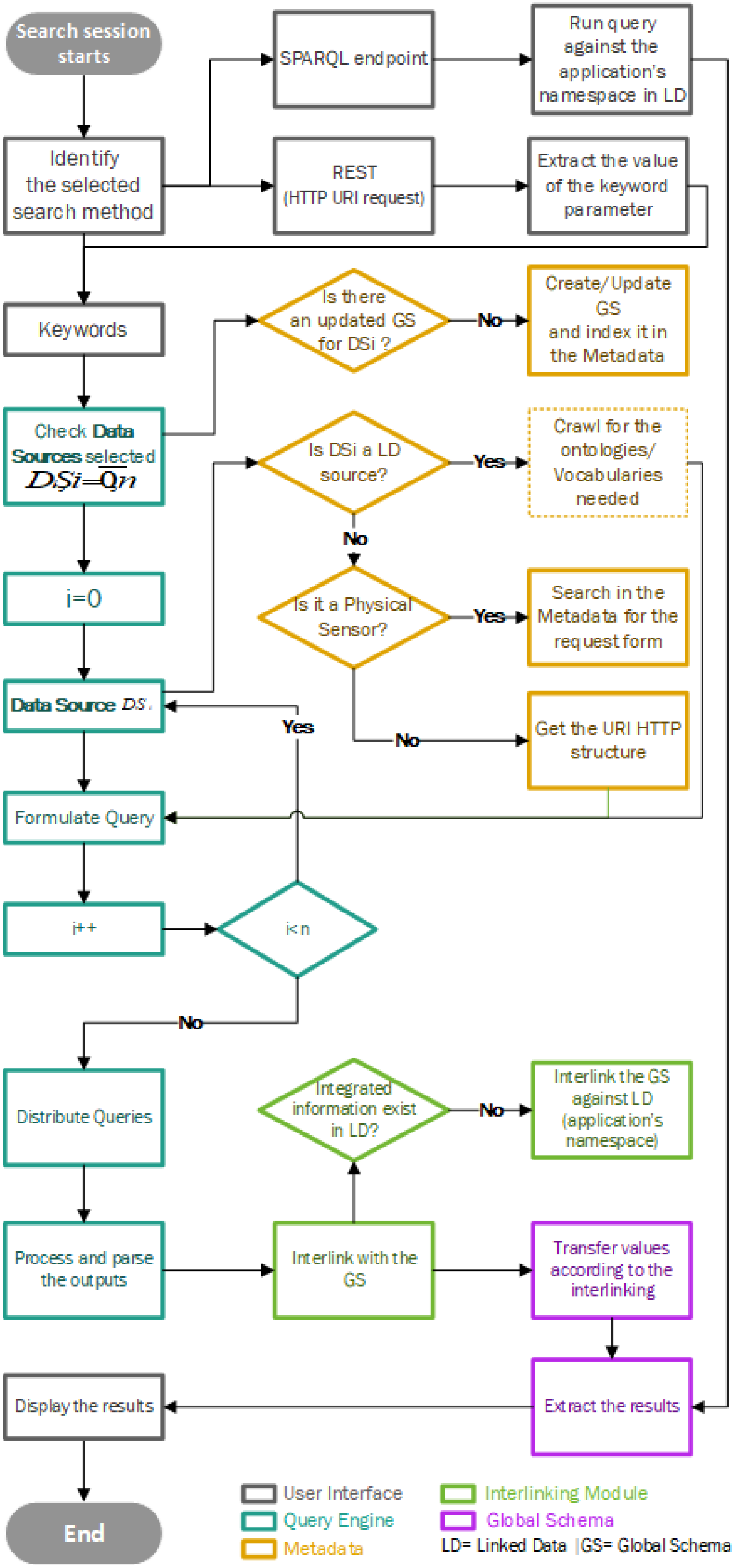

Search session starts
Identify the selected search method
SPARQL endpoint
Run query against the application's namespace in LD
REST (HTTP URI request)
Extract the value of the keyword parameter
Keywords
Is there an updated GS for DSi ?
No
Create/Update GS and index it in the Metadata
Check Data Sources selected DSi=Qn
Is DSi a LD source?
Yes
Crawl for the ontologies/ Vocabularies needed
No
Is it a Physical Sensor?
Yes
Search in the Metadata for the request form
i=0
Data Source DSi
Yes
No
Get the URI HTTP structure
Formulate Query
i++
i<n
No
Distribute Queries
Integrated information exist in LD?
No
Interlink the GS against LD (application's namespace)
Process and parse the outputs
Interlink with the GS
Transfer values according to the interlinking
Display the results
Extract the results
End
User Interface
Query Engine
Metadata
Interlinking Module
Global Schema
LD= Linked Data |GS= Global Schema

Figure 8.2: The flowchart describing the work flow of the proposed framework [Kettouch *et al.*, 2017b].

8.4.2 Other Future Work

There is much future work and improvement that can be introduced to the solutions and the work that is presented in this thesis. This section provides some examples of how the contributed architectures and tools can be extended upon or enhanced in the future.

First, the schema matching approach can benefit from a disambiguation module, which would increase the effectiveness of the approach and presents the properties in a more understandable form. Furthermore, a study investigating the average rate of changes in Linked Data sources would make the settings of SimiMatch by defining the time lapse needed to effectively keep the global schema updated.

The data integration approach, as pointed out previously (such as in Section 7.2) can be adapted to different contexts due to the importance of this task and its wide and diverse use. Accordingly, future work can include applying the novel data integration approach as a component or a layer in different systems and study its impact. Automating certain processes in the extraction of metadata from Linked Data sources can improve SemiLD.

LinkD's further work could be embedded it in a complete publishing tool and set up a dedicated namespace in order to be able to host the converted and published semi-structured data and the identified links.

8.5 Concluding Remarks

Bringing together relevant data from different sources and updating it regularly enables a richer analysis and more accurate decision. More importantly, however, designing generic approaches that are sound theoretically and functions practically to bridge between data models and paradigms not only contributes to the application where it is implemented, but also to the progress the Semantic Web community making for a more homogeneous, automatic and consistent Web.

The author is convinced that this work advances the current understanding and state of-the-art of semantic search. The approaches and the tools proposed along with the other similar approaches working on either data integration or interlinking can substantially affect many areas and improve the accessibility and the usage of semantic data, and that this research brings to light many new ideas that cen be extended to make further contributions.

References

Abawajy, J. 2015. Comprehensive analysis of big data variety landscape. *International journal of parallel, emergent and distributed systems* 30(1), pp. 5–14. 4, 79

Abele, A. and McCrae, J. 2017. *The Linking Open Data cloud diagram*, [Online]. Available at: <`http://lod-cloud.net/`>. [Accessed on: 25 June 2017]. x, 28, 29

Abelló, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazón López, J. N., Naumann, F., Pedersen, T. B., Rizzi, S., Trujillo Mondéjar, J. C., Vassiliadis, P. *et al.* 2013. Fusion cubes: towards self-service business intelligence . 8

Ahammad, T., Al Mamun, M. S. and Tabassum, M. 2016. Towards the application of big data: A new way to make data driven healthcare decision. *International Journal of Computer Applications* 134(14). 4, 5

Alam, M., Buzmakov, A., Codocedo, V. and Napoli, A. 2015. Mining definitions from rdf annotations using formal concept analysis. In: *IJCAI*. pp. 823–829. xiii, 5, 6

Alba, A., Bhagwan, V. and Grandison, T. 2008. Accessing the deep web: when good ideas go bad. In: *Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications*. ACM, pp. 815–818. 2

Albertoni, R., De Martino, M. and Podesta, P. 2014. Environmental thesauri under the lens of reusability. In: *International Conference on Electronic Government and the Information Systems Perspective*. Springer, pp. 222–236. 30

An, J., Kim, Y., Lee, M. and Lee, Y. 2013. Ontology property-based adaptive crawler for linked data (opac). In: *Network of the Future (NOF), 2013 Fourth International Conference on the*. IEEE, pp. 1–6. 10

Araujo, S., Hidders, J., de Vries, A. P. and Schwabe, D. 2011. Serimi: resource description similarity, rdf instance matching and interlinking. In: *Proceedings of the 6th International Conference on Ontology Matching-Volume 814*. CEUR-WS. org, pp. 246–247. 44, 62, 68, 77, 103, 104, 105, 148

Ashish, N. and Mehrotra, S. 2010. Community driven data integration for emergency. In: *Proceedings of the 7th International ISCRAM Conference–Seattle*. vol. 1. 50

Atzeni, P. and Torlone, R. 1997. Schema translation between heterogeneous data models in a lattice framework. In: *Database Applications Semantics*, Springer, pp. 345–364. 40

Aumueller, D., Do, H.-H., Massmann, S. and Rahm, E. 2005. Schema and ontology matching with coma++. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. Acm, pp. 906–908. 66

Batet, M., Sánchez, D., Valls, A. and Gibert, K. 2013. Semantic similarity estimation from multiple ontologies. *Applied intelligence* 38(1), pp. 29–44. 61

Beek, W., Rietveld, L., Schlobach, S. and van Harmelen, F. 2016. Lod laundromat: Why the semantic web needs centralization (even if we don't like it). *IEEE Internet Computing* 20(2), pp. 78–81. 33

Bellahsene, Z., Bonifati, A., Duchateau, F. and Velegrakis, Y. 2011. On evaluating schema matching and mapping. In: *Schema matching and mapping*, Springer, pp. 253–291. 40, 66, 67, 76

Berners-Lee, T. 2006a. *Artificial Intelligence and the Semantic Web*, [Online]. Available at: <http://www.w3.org/2006/Talks/0718-aaai-tbl/Overview.html#(14)>. [Accessed on: 15 July 2017]. x, 20, 21

Berners-Lee, T. 2006b. *Linked Data*, [Online]. Available at: <http://www.w3.org/DesignIssues/LinkedData.html>. [Accessed on: 04 January 2017]. 3, 28

Berners-Lee, T., Fielding, R. and Masinter, L. 1998. Rfc 2396: uniform resource identifiers (uri). *IETF RFC* . 26, 30

Berners-Lee, T. and Fischetti, M. 1999. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco, 1st ed. 2

Berners-Lee, T., Hendler, J. and Lassila, O. 2001. The semantic web. *Scientific American* 284, pp. 34–43. 19, 20

Bernstein, P. A., Madhavan, J. and Rahm, E. 2011. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment* 4(11), pp. 695–701. 43, 95

Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V. and Petrelli, D. 2008. Hybrid search: Effectively combining keywords and semantic searches. In: *European Semantic Web Conference*. Springer, pp. 554–568. 54

Bischof, S., Karapantelakis, A., Nechifor, C.-S., Sheth, A. P., Mileo, A. and Barnaghi, P. 2014. Semantic modelling of smart city data . 44

Bizer, C., Heath, T. and Berners-Lee, T. 2009. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts* pp. 205–227. 2, 3, 55

Böhm, C., de Melo, G., Naumann, F. and Weikum, G. 2012. Linda: distributed web-of-data-scale entity matching. In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, pp. 2104–2108. 62

Bohring, H., Auer, S. *et al.* 2005. Mapping xml to owl ontologies. *Leipziger Informatik-Tage* 72, pp. 147–156. 36

Branting, L. K. 2003. A comparative evaluation of name-matching algorithms. In: *Proceedings of the 9th international conference on Artificial intelligence and law*. ACM, pp. 224–232. 69

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E. and Yergeau, F. 1997. Extensible markup language (xml). *World Wide Web Journal* 2(4), pp. 27–66. 35

Breitling, F. 2009. A standard transformation from xml to rdf via xslt. *Astronomische Nachrichten* 330(7), pp. 755–760. 36

Brennan, R., Feeney, K., Walsh, B., Thomas, H. and O'Sullivan, D. 2011. Explicit federal relationship management to support semantic integration. In: *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. IEEE, pp. 1148–1155. 52

Brickley, D. and Guha, R. 2004. RDF Vocabulary Description Language 1.0: RDF Schema. Tech. rep., Available at: <http://www.w3.org/TR/rdf-schema/>. 22, 23

Buneman, P., Fan, W., Siméon, J. and Weinstein, S. 2001. Constraints for semistructured data and xml. *ACM Sigmod Record* 30(1), pp. 47–54. 33

Burghardt, D., Neun, M. and Weibel, R. 2005. Generalization services on the web-classification and an initial prototype implementation. *Cartography and geographic information science* 32(4), pp. 257–268. 34

Burton, C. 2012. *Is API Growth in a Stall?*, [Online]. Available at: <https://www.kuppingercole.com/blog/burton/is-api-growth-in-a-stall>. [Accessed on: 15 April 2016]. x, 5

Calı, A., Calvanese, D., De Giacomo, G. and Lenzerini, M. 2004. Data integration under integrity constraints. *Information Systems* 29(2), pp. 147–163. 44

Cambiaghi, A., Ferrario, M. and Masseroli, M. 2016. Analysis of metabolomic data: tools, current strategies and future challenges for omics data integration. *Briefings in bioinformatics* 18(3), pp. 498–510. 6

Cao, B., Liu, J., Tang, M., Zheng, Z. and Wang, G. 2013. Mashup service recommendation based on user interest and social network. In: *Web Services (ICWS), 2013 IEEE 20th International Conference on*. IEEE, pp. 99–106. 34

Castano, S., Ferrara, A., Montanelli, S. and Lorusso, D. 2008. Instance matching for ontology population. In: *SEBD*. pp. 121–132. 108

Cheng, G., Ge, W. and Qu, Y. 2008. Falcons: searching and browsing entities on the semantic web. In: *Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 1101–1102. 33

Chou, H. H. 2005. *BioDig: Architecture for integrating heterogeneous biological data repositories using ontologies*. Massachusetts Institute of Technology. 49

Christen, P. 2012. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media. 59

Christodoulou, K. 2015. *On Techniques For Pay-as-you-go Data Integration Of Linked Data*. PhD. The University of Manchester. 24, 30

Chung, S. M. and Jesurajaiah, S. B. 2005. Schemaless xml document management in object-oriented databases. In: *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*. IEEE, vol. 1, pp. 261–266. 34

Collarana, D., Lange, C. and Auer, S. 2016. Fuhsen: a platform for federated, rdf-based hybrid search. In: *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 171–174. 53, 74

Crnkovic, G. 2010. Constructive research and info-computational knowledge generation. *Model-Based Reasoning in Science and Technology* 314, pp. 359–380. 11

Cruz, I. F., Stroe, C., Caci, M., Caimi, F., Palmonari, M., Antonelli, F. P. and Keles, U. C. 2010. Using agreementmaker to align ontologies for oaei 2010. In: *Proceedings of the 5th International Conference on Ontology Matching-Volume 689*. CEUR-WS. org, pp. 118–125. 44

Cruz, I. F., Xiao, H. and Hsu, F. 2004. Peer-to-peer semantic integration of xml and rdf data sources. In: *AP2PC*. Springer, vol. 3601, pp. 108–119. 36

Dayananda, P. and Shettar, R. 2011. Survey on information retrieval in semi structured data. *International Journal of Computer Applications* 32(8), pp. 1–5. 4

Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M. and Horrocks, I. 2000. The semantic web: The roles of xml and rdf. *IEEE Internet computing* 4(5), pp. 63–73. 23, 35

Demidova, E., Risse, T., Tran, G. B. and Gossen, G. 2015. Entity-centric preservation for linked open data: Use cases, requirements and models. In: *SDA@ TPDL*. pp. 61–75. 56

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V. and Sachs, J. 2004. Swoogle: a search and metadata engine for the semantic web. In: *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, pp. 652–659. 33

Do, H.-H., Melnik, S. and Rahm, E. 2002. Comparison of schema matching evaluations. In: *Net. ObjectDays: International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World*. Springer, pp. 221–237. x, 41, 42, 65, 95

Do, H.-H. and Rahm, E. 2002. Coma: a system for flexible combination of schema matching approaches. In: *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, pp. 610–621. 40, 65, 66

Doan, A., Halevy, A. and Ives, Z. 2012. *Principles of data integration*. Elsevier. 6

Dong, H. and Hussain, F. K. 2014. Self-adaptive semantic focused crawler for mining services information discovery. *IEEE Transactions On Industrial Informatics* 10(2), pp. 1616–1626. 43, 61, 120

Efthymiou, K., Sipsas, K., Mourtzis, D. and Chryssolouris, G. 2015. On knowledge reuse for manufacturing systems design and planning: A semantic technology approach. *CIRP Journal of Manufacturing Science and Technology* 8, pp. 1–11. 24

Elmagarmid, A. K., Ipeirotis, P. G. and Verykios, V. S. 2007. Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering* 19(1), pp. 1–16. 59

Erling, O. and Mikhailov, I. 2010. Virtuoso: Rdf support in a native rdbms. In: *Semantic Web Information Management*, Springer, pp. 501–519. 32

Euzenat, J. 2015. Ontology alignments and keys for data interlinking. 56

Euzenat, J., Shvaiko, P. *et al.* 2007. *Ontology matching*, vol. 18. Springer. 59

Fatima, A., Luca, C. and Wilson, G. 2014. New framework for semantic search engine. In: *Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on*. IEEE, pp. 446–451. 120

Fensel, D. 2003. *Ontologies: a silver bullet for knowledge management and electronic commerce*. Springer Science & Business Media. 20

Ferrara, A., Lorusso, D., Montanelli, S. and Varese, G. 2008. Towards a benchmark for instance matching. In: *Proceedings of the 3rd International Conference on Ontology Matching-Volume 431*. CEUR-WS. org, pp. 37–48. 73

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T. 1999. Hypertext transfer protocol–http/1.1. Tech. rep. 30

Fielding, R. T. and Taylor, R. N. 2000. *Architectural styles and the design of network-based software architectures*. PhD. University of California. 34

Force, A. M. 2014. *Mapforce–graphical data mapping, conversion, and integration tool*, [Online]. Available at: <https://www.altova.com/mapforce.html>. [Accessed on: 25 January 2016]. 67

Franco-Bedoya, O. 2015. Open source software ecosystems: Towards a modelling framework. In: *OSS*. pp. 171–179. 22

Freitas, A., Curry, E., Oliveira, J. G. and O'Riain, S. 2012. Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends. *IEEE Internet Computing* 16(1), pp. 24–33. 51

Freudenberg, M., Kontokostas, D., Zeman, V., Meehan, A., Gangemi, A., Belinski, R., Idehen, K., Verborgh, R. and Ismayilov, A. 2017. *DBpedia version 2016-10*, [Online]. Available at: <http://wiki.dbpedia.org/datasets/dbpedia-version-2016-10>. [Accessed on: 02 April 2017]. xiii, 6

Friedrich, S. and Wingerath, W. 2010. *Search-space reduction techniques for duplicate detection in probabilistic data*. Bachelor. Universität Hamburg. 55

Gandon, F. 2014. *URL-URI-IRI*, [Online]. Available at: <http://www-sop.inria.fr/members/Fabien.Gandon/docs/URL-URI-IRI.png>. [Accessed on: 21 June 2015]. x, 27

Getoor, L. and Diehl, C. P. 2005. Link mining: a survey. *Acm Sigkdd Explorations Newsletter* 7(2), pp. 3–12. 100

Giunchiglia, F., Yatskevich, M., Avesani, P. and Shvaiko, P. 2009. A large dataset for the evaluation of ontology matching. *The Knowledge Engineering Review* 24(2), pp. 137–157. 40

Göçebe, P., Dikenelli, O. and Kose, U. 2015. Bringing agility into linked data development: An industrial use case in logistics domain. In: *LDOW@ WWW*. 32

Grobe, M. 2009. Rdf, jena, sparql and the'semantic web'. In: *Proceedings of the 37th annual ACM SIGUCCS fall conference: communication and collaboration*. ACM, pp. 131–138. 32

Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge acquisition* 5(2), pp. 199–220. 20

Guinard, D., Trifa, V. and Wilde, E. 2010. A resource oriented architecture for the web of things. In: *Internet of Things (IOT), 2010*. IEEE, pp. 1–8. 35

Gunaratna, K., Lalithsena, S. and Sheth, A. 2014. Alignment and dataset identification of linked data in semantic web. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4(2), pp. 139–151. 59

Gupta, H. and Mumick, I. S. 2005. Selection of views to materialize in a data warehouse. *IEEE Transactions on Knowledge and Data Engineering* 17(1), pp. 24–43. 45

Han, L., Kashyap, A. L., Finin, T., Mayfield, J. and Weese, J. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. In: ** SEM@ NAACL-HLT*. pp. 44–52. 84, 115, 126, 149

Hartig, O., Bizer, C. and Freytag, J.-C. 2009. Executing sparql queries over the web of linked data. *The Semantic Web-ISWC 2009* pp. 293–309. 32

Hassanzadeh, O. 2013. *Record linkage for web data*. PhD. University of Toronto. 100

Hausenblas, M. 2011. Utilising linked open data in applications. In: *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*. ACM, p. 7. 27

He, B. and Chang, K. C.-C. 2003. Statistical schema matching across web query interfaces. In: *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, pp. 217–228. 76

Heath, T. and Bizer, C. 2011. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology* 1(1), pp. 1–136. 32, 33, 100

Hietanen, E., Lehto, L. and Latvala, P. 2016. Providing geographic datasets as linked data in sdi. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* pp. 583–586. 8

Homoceanu, S., Kalo, J.-C. and Balke, W.-T. 2014. Putting instance matching to the test: Is instance matching ready for reliable data linking? In: *International Symposium on Methodologies for Intelligent Systems*. Springer, pp. 274–284. 62

Hu, W., Yang, R. and Qu, Y. 2014. Automatically generating data linkages using class-based discriminative properties. *Data & Knowledge Engineering* 91, pp. 34–51. 55

Huber, J., Sztyler, T., Noessner, J. and Meilicke, C. 2011. Codi: combinatorial optimization for data integration-results for oaei 2011. In: *Proceedings of the 6th International Conference on Ontology Matching-Volume 814*. CEUR-WS. org, pp. 134–141. 108

Jimenez-Ruiz, E. 2017. *Ontology Alignment Evaluation Initiative*, [Online]. Available at: <http://oaei.ontologymatching.org/>. [Accessed on: 08 May 2017]. 67, 101

Johnson, T. 2013. Indexing linked bibliographic data with json-ld, bibjson and elasticsearch. *Code4lib Journal* 19, pp. 1–11. 36

Kalja, A., Haav, H.-M. and Robal, T. 2014. *Databases and Information Systems VIII: Selected Papers from the Eleventh International Baltic Conference, DB&IS 2014*, vol. 270. IOS Press. 44

Kasanen, E., Lukka, K. and Siitonen, A. 1993. The constructive approach in management accounting research. *Journal of management accounting research* 5, p. 243. 11, 12

Katsis, Y. and Papakonstantinou, Y. 2009. *View-based Data Integration*, Boston, MA: Springer US, pp. 3332–3339. Available at: <https://doi.org/10.1007/978-0-387-39940-9_1072>. x, 47

Kaufmann, E. and Bernstein, A. 2010. Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Web Semantics: Science, Services and Agents on the World Wide Web* 8(4), pp. 377–393. 51

Kettouch, M., Luca, C. and Hobbs, M. 2017a. Schema matching for semi-structured and linked data. In: *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*. IEEE, pp. 270–271. 79, 83

Kettouch, M., Luca, C., Hobbs, M. and Dascalu, S. in press. Using semantic similarity for schema matching of semi-structured and linked data. In: *Internet Technologies and Applications (ITA), 2017*. IEEE. 79

Kettouch, M., Luca, C., Khorief, O., Wu, R. and Dascalu, S. 2017b. Semantic data management in smart cities. In: *Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), 2017 International Conference on*. IEEE, pp. 1126–1131. xii, 8, 76, 150, 151, 152

Kettouch, M. S., Luca, C. and Hobbs, M. 2015a. An interlinking approach based on domain recognition for linked data. In: *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. IEEE, pp. 488–491. 79, 107

Kettouch, M. S., Luca, C., Hobbs, M. and Fatima, A. 2015b. Data integration approach for semi-structured and structured data (linked data). In: *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. IEEE, pp. 820–825. 116

Kettouch, M. S., Luca, C. and Khorief, O. 2016. A framework for integrating and publishing linked data in smart cities. In: *Proceedings of The International Conference on Communications, Computer Science and Information Technology*. 44, 76

Khodaei, A. and Shahabi, C. 2012. Social-textual search and ranking. *CrowdSearch* 37(5), pp. 3–8. 3

Khrouf, H. and Troncy, R. 2016. Eventmedia: A lod dataset of events illustrated with media. *Semantic Web* 7(2), pp. 193–199. 62, 103

Kienast, R. and Baumgartner, C. 2011. Semantic data integration on biomedical data using semantic web technologies. In: *Bioinformatics-Trends and Methodologies*, InTech. 8

Koffina, I., Serfiotis, G., Christophides, V. and Tannen, V. 2006. Mediating rdf/s queries to relational and xml sources. *International Journal on Semantic Web and Information Systems (IJSWIS)* 2(4), pp. 68–91. 75

Krachina, O. and Raskin, V. 2006. Ontology-based inference methods. 20

Kroeze, J. H. 2010. Ontology goes postmodern in ict. In: *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*. ACM, pp. 153–159. 20

Le, B. T., Dieng-Kuntz, R. and Gandon, F. 2004. On ontology matching problems. *ICEIS (4)* pp. 236–243. 65

Le-Phuoc, D., Nguyen-Mau, H. Q., Parreira, J. X. and Hauswirth, M. 2012. A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web* 16, pp. 42–51. 8, 75

Lehmann, J. and Völker, J. 2014. *Perspectives on Ontology Learning*, vol. 18. IOS Press. 28

Lenzerini, M. 2003. Information integration. In: *International Joint Conference on Artificial Intelligence (IJCAI)*. 46

Lesnikova, T. 2016. *RDF Data Interlinking: evaluation of Cross-lingual Methods*. PhD. Université Grenoble Alpes. 55

Lewis, J. R. 1995. Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction* 7(1), pp. 57–78. 137

Li, Y., Li, J., Zhang, D. and Tang, J. 2006a. Result of ontology alignment with rimom at oaei'06. In: *Proceedings of the 1st International Conference on Ontology Matching-Volume 225*. CEUR-WS. org, pp. 181–190. 70, 71

Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D. and Crockett, K. 2006b. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering* 18(8), pp. 1138–1150. 84

Liu, W. 2015. Truth discovery to resolve object conflicts in linked data. *arXiv preprint arXiv:1509.00104* . 37

Lopez, V., Fernández, M., Motta, E. and Stieler, N. 2012. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web* 3(3), pp. 249–265. 53, 74

Lopez, V., Unger, C., Cimiano, P. and Motta, E. 2013. Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web* 21, pp. 3–13. 53

Lopez, V., Uren, V., Motta, E. and Pasin, M. 2007. Aqualog: An ontology-driven question answering system for organizational semantic intranets. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2), pp. 72–105. 74

Lukka, K. 2003. The constructive research approach. *Case study research in logistics. Publications of the Turku School of Economics and Business Administration, Series B* 1(2003), pp. 83–101. 11

Ma, B., Yang, Y., Zhao, F., Dong, R. and Zhou, X. 2015. Semantic similarity computation based on multi-features fusion. *International Journal of Hybrid Information Technology* 8(5), pp. 31–40. 61

Macura, M. 2014. Integration of data from heterogeneous sources using etl technology. *Computer Science* 15(2)), pp. 109–132. 51

Madhavan, J., Bernstein, P. A. and Rahm, E. 2001. Generic schema matching with cupid. In: *vldb*. vol. 1, pp. 49–58. 40, 65

Maleshkova, M., Pedrinaci, C. and Domingue, J. 2010. Investigating web apis on the world wide web. In: *Web Services (ECOWS), 2010 IEEE 8th European Conference on*. IEEE, pp. 107–114. 34

Manakanatas, D. and Plexousakis, D. 2006. A tool for semi-automated semantic schema mapping: Design and implementation. In: *DISWEB*. 65

McMullen, T. and Hawick, K. 2013. Improving platform independent graphical performance by compressing information transfer using json. In: *Proceedings of the International Conference on Semantic Web and Web Services (SWWS)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p. 12. 34

Méndez, E. and Greenberg, J. 2012. Linked data for open vocabularies and hive's global framework. *El profesional de la información* 21(3), pp. 236–244. 28

Minsky, M. 1975. A framework for representing knowledge. In: Winston, P., ed., *The Psychology of Computer Vision*, McGraw-Hill, New York, pp. 211–277. 22

Mitchell, E. T. 2013. Building blocks of linked open data in libraries. *Library technology reports* 49(5), pp. 11–25. 28

Morbidoni, C., Le Phuoc, D., Polleres, A., Samwald, M. and Tummarello, G. 2008. Previewing semantic web pipes. *The Semantic Web: Research and Applications* pp. 843–848. 54

Mukherjea, S., Hirata, K. and Hara, Y. 1997. Towards a multimedia world-wide web information retrieval engine. *Computer networks and ISDN systems* 29(8-13), pp. 1181–1191. 3

Nardi, D., Brachman, R. J. *et al.* 2003. An introduction to description logics. *Description logic handbook* 1, p. 40. 22

Nath, R. P. D., Seddiqui, H. and Aono, M. 2014. A novel automatic property weight generator for semantic data integration. In: *Computer and Information Technology (ICCIT), 2013 16th International Conference on*. IEEE, pp. 408–413. 108, 148

Navarro, G. 2001. A guided tour to approximate string matching. *ACM computing surveys (CSUR)* 33(1), pp. 31–88. 71

Nemirovski, G., Nolle, A., Sicilia, Á., Ballarini, I. and Corado, V. 2013. Data integration driven ontology design, case study smart city. In: *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*. ACM, p. 43. 44

Nentwig, M., Hartung, M., Ngonga Ngomo, A.-C. and Rahm, E. 2017. A survey of current link discovery frameworks. *Semantic Web* 8(3), pp. 419–436. 69

Ngomo, A.-C. N. and Auer, S. 2011. Limes-a time-efficient approach for large-scale link discovery on the web of data. In: *IJCAI*. pp. 2312–2317. 56, 72

Nguyen, K. and Ichise, R. 2013. Slint+ results for oaei 2013 instance matching. In: *Proceedings of the 8th International Conference on Ontology Matching-Volume 1111*. CEUR-WS. org, pp. 177–183. 62, 70

Nguyen, K. and Ichise, R. 2015. Scslint: Time and memory efficient interlinking framework for linked data. In: *International Semantic Web Conference (Posters & Demos)*. 70, 72

Nguyen, K., Ichise, R. and Le, B. 2012a. Interlinking linked data sources using a domain-independent system. In: *Joint International Semantic Technology Conference*. Springer, pp. 113–128. 68

Nguyen, K., Ichise, R. and Le, B. 2012b. Slint: a schema-independent linked data interlinking system. In: *Proceedings of the 7th International Conference on Ontology Matching-Volume 946*. CEUR-WS. org, pp. 1–12. 56, 59, 62, 125

Nikolov, A., Uren, V., Motta, E. and De Roeck, A. N. 2008a. Integration of semantically annotated data by the knofuss architecture. In: *EKAW*. Springer, pp. 265–274. 60

Nikolov, A., Uren, V. S., Motta, E. and De Roeck, A. N. 2008b. Refining instance coreferencing results using belief propagation. In: *ASWC*. Springer, pp. 405–419. 61

Niu, X., Wang, H., Wu, G., Qi, G. and Yu, Y. 2011. Evaluating the stability and credibility of ontology matching methods. *The Semantic Web: Research and Applications* pp. 275–289. 70

Pfaff, M. and Krcmar, H. 2014. Semantic integration of semi-structured distributed data in the domain of it benchmarking. In: *16th International Conference on Enterprise Information Systems (ICEIS), Lisbon, Portugal*. 68, 95, 144

Poe, V., Brobst, S. and Klauer, P. 1997. *Building a data warehouse for decision support*. Prentice-Hall, Inc. 45

Poggi, A. 2006. Structured and semi-structured data integration. *Dipartimento di Informatica e Sistemistica, Università di Roma La Sapienza* . 55

Popov, I. 2013. *End-user data-centric interactions over linked data*. PhD. University of Southampton, Available at: <`https://eprints.soton.ac.uk/361729/`>. 4, 20

ProgrammableWeb. 2013. *ProgrammableWeb Research Center*, [Online]. Available at: <`https://www.programmableweb.com/api-research`>. [Accessed on: 12 December 2016]. x, 5

Rahm, E. 2011. Towards large-scale schema and ontology matching. In: *Schema matching and mapping*, Springer, pp. 3–27. 66

Rajabi, E. *et al.* 2015. *Interlinking educational data to web of data*. PhD. Universidad de Alcala. 72

Ramis, B., Gonzalez, L., Iarovyi, S., Lobov, A., Lastra, J. L. M., Vyatkin, V. and Dai, W. 2014. Knowledge-based web service integration for industrial automation. In: *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. IEEE, pp. 733–739. 120

Rathinasamy, K. 2011. *Comparison of Schema and Data Integration tools for the Asset Management Domain*. PhD. University of South Australia. 67

Ray, E. T. 2003. *Learning XML: creating self-describing data*. " O'Reilly Media, Inc.". 34

Richardson, L. and Ruby, S. 2008. *RESTful web services*. " O'Reilly Media, Inc.". 34

Rietveld, L. 2016. *Publishing and Consuming Linked Data: Optimizing for the Unknown*. Studies on the semantic web. 31

Robal, T. and Kalja, A. 2009. Creating interactive learning objects with web services. In: *EAEEIE Annual Conference, 2009*. IEEE, pp. 1–6. 35

Rong, S., Niu, X., Xiang, E. W., Wang, H., Yang, Q. and Yu, Y. 2012. A machine learning approach for instance matching based on similarity metrics. In: *International Semantic Web Conference*. Springer, pp. 460–475. 71

Roy, S. D. and Zeng, W. 2015. Revelations from social multimedia data. In: *Social Multimedia Signals*, Springer, pp. 135–142. 27

Rula, A. and Palmonari, M. 2013. Time-related quality dimensions in linked data. 23

Scharffe, F. and Euzenat, J. 2011. Melinda: an interlinking framework for the web of data. *CoRR* abs/1107.4502. Available at: <http://arxiv.org/abs/1107.4502>. x, 56

Scharffe, F., Ferrara, A. and Nikolov, A. 2013. Data linking for the semantic web. *Semantic Web: Ontology and Knowledge Base Enabled Tools, Services, and Applications* 169, p. 326. 59

Schlaefer, N., Ko, J., Betteridge, J., Pathak, M. A., Nyberg, E. and Sautter, G. 2007. Semantic extensions of the ephyra qa system for trec 2007. In: *TREC*. vol. 1, p. 2. 53

Schmachtenberg, M., Bizer, C. and Paulheim, H. 2014. *State of the LOD Cloud 2014*, [Online]. Available at: <http://lod-cloud.net/state/state_2014/>. [Accessed on: 02 May 2016]. xiii, 6

Seligman, L., Mork, P., Halevy, A., Smith, K., Carey, M. J., Chen, K., Wolf, C., Madhavan, J., Kannan, A. and Burdick, D. 2010. Openii: an open source information integration toolkit. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, pp. 1057–1060. 66

Shadbolt, N., Berners-Lee, T. and Hall, W. 2006. The semantic web revisited. *IEEE intelligent systems* 21(3), pp. 96–101. 19

Shekarpour, S., Marx, E., Ngomo, A.-C. N. and Auer, S. 2015. Sina: Semantic interpretation of user queries for question answering on interlinked data. *Web Semantics: Science, Services and Agents on the World Wide Web* 30, pp. 39–51. 53

Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M. and Cruz, I. 2010. Ontology matching om-2010. *Unknown Journal* 689. 70

Singh, K., Gulati, D. and Gulati, D. 2011. Technological march from web 1.0 to web 3.0: A comparative study. *Library Herald* 49(2), pp. 146–157. 19

Songtao, L. and Junliang, C. 2005. Semantic web enabled vhe for 3/sup rd/generation telecommunications. In: *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on*. IEEE, pp. 539–544. 19

Sumaray, A. and Makki, S. K. 2012. A comparison of data serialization formats for optimal efficiency on a mobile platform. In: *Proceedings of the 6th international conference on ubiquitous information management and communication*. ACM, p. 48. 35

Svoboda, M. and Mlỳnková, I. 2011. Efficient querying of distributed linked data. In: *Proceedings of the 2011 Joint EDBT/ICDT Ph. D. Workshop*. ACM, pp. 45–50. 10

Symeonidou, D. 2014. *Automatic key discovery for Data Linking*. PhD. University of Paris 11. 72

Szeredi, P., Lukácsy, G., Benkő, T. and Nagy, Z. 2014. *The Semantic Web Explained: The Technology and Mathematics behind Web 3.0*. Cambridge University Press. 2, 3, 5

Taheri, A. and Shamsfard, M. 2012. Consolidation of linked data resources upon heterogeneous schemas. In: *Proceedings of the Sixth International Conference on Advances in Semantic Processing (SEMAPRO 2012), Spain*. 7

Trifa, V., Guinard, D., Davidovski, V., Kamilaris, A. and Delchev, I. 2010. Web messaging for open and scalable distributed sensing applications. *Web Engineering* pp. 129–143. 34

Tuomi, I. 1999. Data is more than knowledge: Implications of the reversed knowledge hierarchy for knowledge management and organizational memory. In: *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*. IEEE, pp. 12–pp. 27

Ullman, J. D. 1990. *Principles of Database and Knowledge-Base Systems: Volume II: The New Technologies*. New York, NY, USA: W. H. Freeman & Co. 22

Umbrich, J., Hogan, A., Polleres, A. and Decker, S. 2012. Improving the recall of live linked data querying through reasoning. *RR* 7497, pp. 188–204. 55

Usbeck, R., Ngomo, A.-C. N., Bühmann, L. and Unger, C. 2015. Hawk–hybrid question answering using linked data. In: *European Semantic Web Conference*. Springer, pp. 353–368. 54

Van der Aalst, W. M. and Kumar, A. 2003. Xml–based schema definition for support of interorganizational workflow. *Information Systems Research* 14(1), pp. 23–46. 35

Van Deursen, D., Poppe, C., Martens, G., Mannens, E. and Van de Walle, R. 2008. Xml to rdf conversion: a generic approach. In: *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on*. IEEE, pp. 138–144. 36

Verborgh, R., Hartig, O., De Meester, B., Haesendonck, G., De Vocht, L., Vander Sande, M., Cyganiak, R., Colpaert, P., Mannens, E. and Van de Walle, R. 2014a. Querying datasets on the web with high availability. In: *International Semantic Web Conference*. Springer, pp. 180–196. 33

Verborgh, R., Vander Sande, M., Colpaert, P., Coppens, S., Mannens, E. and Van de Walle, R. 2014b. Web-scale querying through linked data fragments. In: *LDOW*. 32, 33

Vincini, M., Beneventano, D. and Bergamaschi, S. 2013. Semantic integration of heterogeneous data sources in the momis data transformation system. *J. UCS* 19(13), pp. 1986–2012. 75

Volz, J., Bizer, C., Gaedke, M. and Kobilarov, G. 2009. Silk-a link discovery framework for the web of data. *LDOW* 538. 71, 101

W3C. 2004. *OWL Web Ontology Language Reference*, [Online]. Available at: <https://www.w3.org/TR/owl-ref/>. [Accessed on: 20 March 2016]. 22

W3C SPARQL Working Group. 2008. *SPARQL Query Language for RDF*, [Online]. Available at: <https://www.w3.org/TR/rdf-sparql-query/>. [Accessed on: 25 August 2016]. 24

W3C SPARQL Working Group. 2013. *SPARQL 1.1 Overview*, [Online]. Available at: <http://www.w3.org/TR/sparql11-query/>. [Accessed on: 25 August 2016]. 24

Wang, C., Lu, J. and Zhang, G. 2006. Integration of ontology data through learning instance matching. In: *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*. IEEE, pp. 536–539. 20, 59

Wang, Z., Li, J., Wang, Z. and Tang, J. 2012. Cross-lingual knowledge linking across wiki knowledge bases. In: *Proceedings of the 21st international conference on World Wide Web*. ACM, pp. 459–468. 71

Wang, Z., Zhang, X., Hou, L., Zhao, Y., Li, J., Qi, Y. and Tang, J. 2010. Rimom results for oaei 2010. In: *Proceedings of the 5th International Conference on Ontology Matching-Volume 689*. CEUR-WS. org, pp. 195–202. 44

Wu, R., Hossain, M., Painumkal, J., Kettouch, M. S., Luca, C., Dascalu, S. and Harris, F. in press. Web-service framework for environmental models. In: *Internet Technologies and Applications (ITA), 2017*. IEEE. 34

Wu, X., Xia, J. C., West, G., Arnold, L. and Veenendaal, B. 2012. Managing schema evolution in a federated spatial database system. In: *Discovery of Geospatial Resources: Methodologies, Technologies, and Emergent Applications*, IGI Global, pp. 56–77. 49

Xu, Y. and Mease, D. 2009. Evaluating web search using task completion time. In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 676–677. 133

Ya-qin, F. and Wen-yong, F. 2010. Xml in web data mining application. In: *Information Engineering (ICIE), 2010 WASE International Conference on*. IEEE, vol. 4, pp. 53–56. 34

Yaghouti, N., Kahani, M. and Behkamal, B. 2015. A metric-driven approach for interlinking assessment of rdf graphs. In: *Computer Science and Software Engineering (CSSE), 2015 International Symposium on*. IEEE, pp. 1–8. 26

Yeganeh, S. H., Hassanzadeh, O. and Miller, R. J. 2011. Linking semistructured data on the web. *Interface* . 101

Zhang, M., Yuan, J., Gong, J. and Yue, P. 2013. An interlinking approach for linked geospatial data. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1(2), pp. 283–287. 62, 103

Zhang, Y., Jin, H., Pan, L. and Li, J.-Z. 2016. Rimom results for oaei 2016. In: *OM@ ISWC*. pp. 210–216. 70

Zhao, J. 2010. Publishing chinese medicine knowledge as linked data on the web. *Chinese medicine* 5(1), p. 27. 38

Zheng, Q., Shao, C., Li, J., Wang, Z. and Hu, L. 2013. Rimom2013 results for oaei 2013. In: *Proceedings of the 8th International Conference on Ontology Matching-Volume 1111*. CEUR-WS. org, pp. 161–168. 62, 71, 108

Ziegler, P. and Dittrich, K. R. 2007. Data integration-problems, approaches, and perspectives. *Conceptual Modelling in Information Systems Engineering* pp. 39–58. 52

Appendix I - Full Evaluation Results of SimiMatch

The global schema version and the number of results stops at 200 in the table below as further data does not significantly change the F1 score or the trend already reported.

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
10	M	0.7	10	482	1	0.87	0.93
10	M	0.7	50	482	0.98	0.88	0.93
10	M	0.7	100	482	0.98	0.9	0.94
10	M	0.7	200	482	0.98	0.9	0.94
10	M	0.7	300	482	0.98	0.92	0.95
10	M	0.7	400	482	0.98	0.92	0.95
10	M	0.7	500	482	0.98	0.92	0.95
10	M	0.7	600	482	0.98	0.92	0.95
10	M	0.7	700	482	0.98	0.92	0.95
10	M	0.75	10	482	1	0.87	0.93
10	M	0.75	50	482	0.98	0.88	0.93
10	M	0.75	100	482	0.98	0.9	0.94
10	M	0.75	200	482	0.98	0.9	0.94
10	M	0.75	300	482	0.98	0.92	0.95
10	M	0.75	400	482	0.98	0.92	0.95
10	M	0.75	500	482	0.98	0.92	0.95
10	M	0.75	600	482	0.98	0.92	0.95
10	M	0.75	700	482	0.98	0.92	0.95
10	M	0.8	10	482	1	0.86	0.92
10	M	0.8	50	482	1	0.87	0.93
10	M	0.8	100	482	0.99	0.91	0.95
10	M	0.8	200	482	0.99	0.91	0.95
10	M	0.8	300	482	0.99	0.91	0.95
10	M	0.8	400	482	0.99	0.91	0.95
10	M	0.8	500	482	0.99	0.91	0.95
10	M	0.8	600	482	0.99	0.91	0.95
10	M	0.8	700	482	0.99	0.91	0.95

Continued on next page

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
10	M	0.85	10	482	1	0.86	0.92
10	M	0.85	50	482	1	0.87	0.93
10	M	0.85	100	482	0.99	0.91	0.95
10	M	0.85	200	482	0.99	0.91	0.95
10	M	0.85	300	482	0.99	0.91	0.95
10	M	0.85	400	482	0.99	0.91	0.95
10	M	0.85	500	482	0.99	0.91	0.95
10	M	0.85	600	482	0.99	0.91	0.95
10	M	0.85	700	482	0.99	0.91	0.95
10	M	0.9	10	482	1	0.86	0.92
10	M	0.9	50	482	1	0.87	0.93
10	M	0.9	100	482	0.99	0.91	0.95
10	M	0.9	200	482	0.99	0.91	0.95
10	M	0.9	300	482	0.99	0.91	0.95
10	M	0.9	400	482	0.99	0.91	0.95
10	M	0.9	500	482	0.99	0.91	0.95
10	M	0.9	600	482	0.99	0.91	0.95
10	M	0.9	700	482	0.99	0.91	0.95
10	M	0.95	10	482	1	0.86	0.92
10	M	0.95	50	482	1	0.87	0.93
10	M	0.95	100	482	0.99	0.91	0.95
10	M	0.95	200	482	0.99	0.91	0.95
10	M	0.95	300	482	0.99	0.91	0.95
10	M	0.95	400	482	0.99	0.91	0.95
10	M	0.95	500	482	0.99	0.91	0.95
10	M	0.95	600	482	0.99	0.91	0.95
10	M	0.95	700	482	0.99	0.91	0.95
50	M	0.7	10	2411	1	0.83	0.91
50	M	0.7	50	2411	0.98	0.87	0.92
50	M	0.7	100	2411	0.98	0.91	0.94
50	M	0.7	200	2411	0.98	0.91	0.94
50	M	0.7	300	2411	0.98	0.91	0.94
50	M	0.7	400	2411	0.98	0.91	0.94
50	M	0.7	500	2411	0.98	0.91	0.94
50	M	0.7	600	2411	0.98	0.91	0.94
50	M	0.7	700	2411	0.98	0.91	0.94
50	M	0.75	10	2411	1	0.82	0.9
50	M	0.75	50	2411	0.98	0.87	0.92
50	M	0.75	100	2411	0.98	0.91	0.94
50	M	0.75	200	2411	0.98	0.91	0.94
50	M	0.75	300	2411	0.98	0.91	0.94
50	M	0.75	400	2411	0.98	0.91	0.94
50	M	0.75	500	2411	0.98	0.91	0.94

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
50	M	0.75	600	2411	0.98	0.91	0.94
50	M	0.75	700	2411	0.98	0.91	0.94
50	M	0.8	10	2411	1	0.82	0.9
50	M	0.8	50	2411	1	0.87	0.93
50	M	0.8	100	2411	0.99	0.9	0.94
50	M	0.8	200	2411	0.99	0.9	0.94
50	M	0.8	300	2411	0.99	0.9	0.94
50	M	0.8	400	2411	0.99	0.9	0.94
50	M	0.8	500	2411	0.99	0.9	0.94
50	M	0.8	600	2411	0.99	0.9	0.94
50	M	0.8	700	2411	0.99	0.9	0.94
50	M	0.85	10	2411	1	0.82	0.9
50	M	0.85	50	2411	1	0.87	0.93
50	M	0.85	100	2411	0.99	0.9	0.94
50	M	0.85	200	2411	0.99	0.9	0.94
50	M	0.85	300	2411	0.99	0.9	0.94
50	M	0.85	400	2411	0.99	0.9	0.94
50	M	0.85	500	2411	0.99	0.9	0.94
50	M	0.85	600	2411	0.99	0.9	0.94
50	M	0.85	700	2411	0.99	0.9	0.94
50	M	0.9	10	2411	1	0.82	0.9
50	M	0.9	50	2411	1	0.87	0.93
50	M	0.9	100	2411	0.99	0.9	0.94
50	M	0.9	200	2411	0.99	0.9	0.94
50	M	0.9	300	2411	0.99	0.9	0.94
50	M	0.9	400	2411	0.99	0.9	0.94
50	M	0.9	500	2411	0.99	0.9	0.94
50	M	0.9	600	2411	0.99	0.9	0.94
50	M	0.9	700	2411	0.99	0.9	0.94
50	M	0.95	10	2411	1	0.82	0.9
50	M	0.95	50	2411	1	0.87	0.93
50	M	0.95	100	2411	0.99	0.9	0.94
50	M	0.95	200	2411	0.99	0.9	0.94
50	M	0.95	300	2411	0.99	0.9	0.94
50	M	0.95	400	2411	0.99	0.9	0.94
50	M	0.95	500	2411	0.99	0.9	0.94
50	M	0.95	600	2411	0.99	0.9	0.94
50	M	0.95	700	2411	0.99	0.9	0.94
100	M	0.7	10	5124	1	0.83	0.91
100	M	0.7	50	5124	0.98	0.87	0.92
100	M	0.7	100	5124	0.98	0.91	0.94
100	M	0.7	200	5124	0.98	0.91	0.94
100	M	0.7	300	5124	0.98	0.91	0.94

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
100	M	0.7	400	5124	0.98	0.91	0.94
100	M	0.7	500	5124	0.98	0.91	0.94
100	M	0.7	600	5124	0.98	0.91	0.94
100	M	0.7	700	5124	0.98	0.91	0.94
100	M	0.75	10	5124	1	0.83	0.91
100	M	0.75	50	5124	0.98	0.87	0.92
100	M	0.75	100	5124	0.98	0.89	0.93
100	M	0.75	200	5124	0.98	0.89	0.93
100	M	0.75	300	5124	0.98	0.89	0.93
100	M	0.75	400	5124	0.98	0.89	0.93
100	M	0.75	500	5124	0.98	0.89	0.93
100	M	0.75	600	5124	0.98	0.89	0.93
100	M	0.75	700	5124	0.98	0.89	0.93
100	M	0.8	10	5124	1	0.82	0.9
100	M	0.8	50	5124	0.99	0.87	0.93
100	M	0.8	100	5124	0.99	0.89	0.94
100	M	0.8	200	5124	0.99	0.89	0.94
100	M	0.8	300	5124	0.99	0.89	0.94
100	M	0.8	400	5124	0.99	0.89	0.94
100	M	0.8	500	5124	0.99	0.89	0.94
100	M	0.8	600	5124	0.99	0.89	0.94
100	M	0.8	700	5124	0.99	0.89	0.94
100	M	0.85	10	5124	1	0.82	0.9
100	M	0.85	50	5124	0.99	0.87	0.93
100	M	0.85	100	5124	0.99	0.89	0.94
100	M	0.85	200	5124	0.99	0.89	0.94
100	M	0.85	300	5124	0.99	0.89	0.94
100	M	0.85	400	5124	0.99	0.89	0.94
100	M	0.85	500	5124	0.99	0.89	0.94
100	M	0.85	600	5124	0.99	0.89	0.94
100	M	0.85	700	5124	0.99	0.89	0.94
100	M	0.9	10	5124	1	0.82	0.9
100	M	0.9	50	5124	0.99	0.87	0.93
100	M	0.9	100	5124	0.99	0.89	0.94
100	M	0.9	200	5124	0.99	0.89	0.94
100	M	0.9	300	5124	0.99	0.89	0.94
100	M	0.9	400	5124	0.99	0.89	0.94
100	M	0.9	500	5124	0.99	0.89	0.94
100	M	0.9	600	5124	0.99	0.89	0.94
100	M	0.9	700	5124	0.99	0.89	0.94
100	M	0.95	10	5124	1	0.81	0.9
100	M	0.95	50	5124	0.99	0.86	0.92
100	M	0.95	100	5124	0.99	0.89	0.94

Continued on next page

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
100	M	0.95	200	5124	0.99	0.89	0.94
100	M	0.95	300	5124	0.99	0.89	0.94
100	M	0.95	400	5124	0.99	0.89	0.94
100	M	0.95	500	5124	0.99	0.89	0.94
100	M	0.95	600	5124	0.99	0.89	0.94
100	M	0.95	700	5124	0.99	0.89	0.94
200	M	0.7	10	10382	1	0.84	0.91
200	M	0.7	50	10382	0.98	0.89	0.93
200	M	0.7	100	10382	0.98	0.92	0.95
200	M	0.7	200	10382	0.98	0.92	0.95
200	M	0.7	300	10382	0.98	0.92	0.95
200	M	0.7	400	10382	0.98	0.92	0.95
200	M	0.7	500	10382	0.98	0.92	0.95
200	M	0.7	600	10382	0.98	0.92	0.95
200	M	0.7	700	10382	0.98	0.92	0.95
200	M	0.75	10	10382	1	0.84	0.91
200	M	0.75	50	10382	0.98	0.89	0.93
200	M	0.75	100	10382	0.98	0.92	0.95
200	M	0.75	200	10382	0.98	0.92	0.95
200	M	0.75	300	10382	0.98	0.92	0.95
200	M	0.75	400	10382	0.98	0.92	0.95
200	M	0.75	500	10382	0.98	0.92	0.95
200	M	0.75	600	10382	0.98	0.92	0.95
200	M	0.75	700	10382	0.98	0.92	0.95
200	M	0.8	10	10382	1	0.84	0.91
200	M	0.8	50	10382	0.99	0.88	0.93
200	M	0.8	100	10382	0.99	0.91	0.95
200	M	0.8	200	10382	0.99	0.91	0.95
200	M	0.8	300	10382	0.99	0.91	0.95
200	M	0.8	400	10382	0.99	0.91	0.95
200	M	0.8	500	10382	0.99	0.91	0.95
200	M	0.8	600	10382	0.99	0.91	0.95
200	M	0.8	700	10382	0.99	0.91	0.95
200	M	0.85	10	10382	1	0.84	0.91
200	M	0.85	50	10382	0.99	0.88	0.93
200	M	0.85	100	10382	0.99	0.91	0.95
200	M	0.85	200	10382	0.99	0.91	0.95
200	M	0.85	300	10382	0.99	0.91	0.95
200	M	0.85	400	10382	0.99	0.91	0.95
200	M	0.85	500	10382	0.99	0.91	0.95
200	M	0.85	600	10382	0.99	0.91	0.95
200	M	0.85	700	10382	0.99	0.91	0.95
200	M	0.9	10	10382	1	0.84	0.91

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
200	M	0.9	50	10382	0.99	0.88	0.93
200	M	0.9	100	10382	0.99	0.91	0.95
200	M	0.9	200	10382	0.99	0.91	0.95
200	M	0.9	300	10382	0.99	0.91	0.95
200	M	0.9	400	10382	0.99	0.91	0.95
200	M	0.9	500	10382	0.99	0.91	0.95
200	M	0.9	600	10382	0.99	0.91	0.95
200	M	0.9	700	10382	0.99	0.91	0.95
200	M	0.95	10	10382	1	0.82	0.9
200	M	0.95	50	10382	1	0.87	0.93
200	M	0.95	100	10382	0.99	0.89	0.94
200	M	0.95	200	10382	0.99	0.89	0.94
200	M	0.95	300	10382	0.99	0.89	0.94
200	M	0.95	400	10382	0.99	0.89	0.94
200	M	0.95	500	10382	0.99	0.89	0.94
200	M	0.95	600	10382	0.99	0.89	0.94
200	M	0.95	700	10382	0.99	0.89	0.94
10	G	0.7	10	424	0.99	0.75	0.85
10	G	0.7	50	424	0.98	0.8	0.88
10	G	0.7	100	424	0.96	0.86	0.91
10	G	0.7	200	424	0.96	0.89	0.92
10	G	0.7	300	424	0.96	0.89	0.92
10	G	0.7	400	424	0.96	0.89	0.92
10	G	0.7	500	424	0.96	0.89	0.92
10	G	0.7	600	424	0.96	0.89	0.92
10	G	0.7	700	424	0.96	0.89	0.92
10	G	0.75	10	424	0.99	0.75	0.85
10	G	0.75	50	424	0.99	0.79	0.88
10	G	0.75	100	424	0.99	0.85	0.91
10	G	0.75	200	424	0.99	0.88	0.93
10	G	0.75	300	424	0.99	0.88	0.93
10	G	0.75	400	424	0.99	0.88	0.93
10	G	0.75	500	424	0.99	0.88	0.93
10	G	0.75	600	424	0.99	0.88	0.93
10	G	0.75	700	424	0.99	0.88	0.93
10	G	0.8	10	424	1	0.75	0.86
10	G	0.8	50	424	0.99	0.78	0.87
10	G	0.8	100	424	0.99	0.84	0.91
10	G	0.8	200	424	0.99	0.88	0.93
10	G	0.8	300	424	0.99	0.88	0.93
10	G	0.8	400	424	0.99	0.88	0.93
10	G	0.8	500	424	0.99	0.88	0.93

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
10	G	0.8	600	424	0.99	0.88	0.93
10	G	0.8	700	424	0.99	0.88	0.93
10	G	0.85	10	424	1	0.74	0.85
10	G	0.85	50	424	0.99	0.77	0.87
10	G	0.85	100	424	0.99	0.83	0.9
10	G	0.85	200	424	0.99	0.85	0.91
10	G	0.85	300	424	0.99	0.85	0.91
10	G	0.85	400	424	0.99	0.85	0.91
10	G	0.85	500	424	0.99	0.85	0.91
10	G	0.85	600	424	0.99	0.85	0.91
10	G	0.85	700	424	0.99	0.85	0.91
10	G	0.9	10	424	1	0.74	0.85
10	G	0.9	50	424	0.99	0.77	0.87
10	G	0.9	100	424	0.99	0.83	0.9
10	G	0.9	200	424	0.99	0.85	0.91
10	G	0.9	300	424	0.99	0.85	0.91
10	G	0.9	400	424	0.99	0.85	0.91
10	G	0.9	500	424	0.99	0.85	0.91
10	G	0.9	600	424	0.99	0.85	0.91
10	G	0.9	700	424	0.99	0.85	0.91
10	G	0.95	10	424	1	0.73	0.84
10	G	0.95	50	424	1	0.76	0.86
10	G	0.95	100	424	0.99	0.82	0.9
10	G	0.95	200	424	0.99	0.85	0.91
10	G	0.95	300	424	0.99	0.85	0.91
10	G	0.95	400	424	0.99	0.85	0.91
10	G	0.95	500	424	0.99	0.85	0.91
10	G	0.95	600	424	0.99	0.85	0.91
10	G	0.95	700	424	0.99	0.85	0.91
50	G	0.7	10	2725	0.99	0.72	0.83
50	G	0.7	50	2725	0.98	0.76	0.86
50	G	0.7	100	2725	0.96	0.81	0.88
50	G	0.7	200	2725	0.96	0.85	0.9
50	G	0.7	300	2725	0.96	0.85	0.9
50	G	0.7	400	2725	0.96	0.85	0.9
50	G	0.7	500	2725	0.96	0.85	0.9
50	G	0.7	600	2725	0.96	0.85	0.9
50	G	0.7	700	2725	0.96	0.85	0.9
50	G	0.75	10	2725	0.99	0.72	0.83
50	G	0.75	50	2725	0.99	0.76	0.86
50	G	0.75	100	2725	0.99	0.81	0.89
50	G	0.75	200	2725	0.99	0.85	0.91
50	G	0.75	300	2725	0.99	0.85	0.91

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
50	G	0.75	400	2725	0.99	0.85	0.91
50	G	0.75	500	2725	0.99	0.85	0.91
50	G	0.75	600	2725	0.99	0.85	0.91
50	G	0.75	700	2725	0.99	0.85	0.91
50	G	0.8	10	2725	1	0.71	0.83
50	G	0.8	50	2725	0.98	0.75	0.85
50	G	0.8	100	2725	0.98	0.8	0.88
50	G	0.8	200	2725	0.98	0.84	0.9
50	G	0.8	300	2725	0.98	0.84	0.9
50	G	0.8	400	2725	0.98	0.84	0.9
50	G	0.8	500	2725	0.98	0.84	0.9
50	G	0.8	600	2725	0.98	0.84	0.9
50	G	0.8	700	2725	0.98	0.84	0.9
50	G	0.85	10	2725	1	0.71	0.83
50	G	0.85	50	2725	0.98	0.75	0.85
50	G	0.85	100	2725	0.98	0.8	0.88
50	G	0.85	200	2725	0.98	0.84	0.9
50	G	0.85	300	2725	0.98	0.83	0.9
50	G	0.85	400	2725	0.98	0.83	0.9
50	G	0.85	500	2725	0.98	0.83	0.9
50	G	0.85	600	2725	0.98	0.83	0.9
50	G	0.85	700	2725	0.98	0.83	0.9
50	G	0.9	10	2725	1	0.7	0.82
50	G	0.9	50	2725	0.99	0.73	0.84
50	G	0.9	100	2725	0.99	0.78	0.87
50	G	0.9	200	2725	0.99	0.83	0.9
50	G	0.9	300	2725	0.99	0.82	0.9
50	G	0.9	400	2725	0.99	0.82	0.9
50	G	0.9	500	2725	0.99	0.82	0.9
50	G	0.9	600	2725	0.99	0.82	0.9
50	G	0.9	700	2725	0.99	0.82	0.9
50	G	0.95	10	2725	1	0.7	0.82
50	G	0.95	50	2725	0.99	0.73	0.84
50	G	0.95	100	2725	0.99	0.78	0.87
50	G	0.95	200	2725	0.99	0.83	0.9
50	G	0.95	300	2725	0.99	0.82	0.9
50	G	0.95	400	2725	0.99	0.82	0.9
50	G	0.95	500	2725	0.99	0.82	0.9
50	G	0.95	600	2725	0.99	0.82	0.9
50	G	0.95	700	2725	0.99	0.82	0.9
100	G	0.7	10	5573	0.99	0.72	0.83
100	G	0.7	50	5573	0.97	0.75	0.85
100	G	0.7	100	5573	0.95	0.81	0.87

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
100	G	0.7	200	5573	0.95	0.85	0.9
100	G	0.7	300	5573	0.95	0.85	0.9
100	G	0.7	400	5573	0.95	0.85	0.9
100	G	0.7	500	5573	0.95	0.85	0.9
100	G	0.7	600	5573	0.95	0.85	0.9
100	G	0.7	700	5573	0.95	0.85	0.9
100	G	0.75	10	5573	0.99	0.72	0.83
100	G	0.75	50	5573	0.98	0.75	0.85
100	G	0.75	100	5573	0.98	0.8	0.88
100	G	0.75	200	5573	0.98	0.85	0.91
100	G	0.75	300	5573	0.98	0.85	0.91
100	G	0.75	400	5573	0.98	0.85	0.91
100	G	0.75	500	5573	0.98	0.85	0.91
100	G	0.75	600	5573	0.98	0.85	0.91
100	G	0.75	700	5573	0.97	0.85	0.91
100	G	0.8	10	5573	1	0.71	0.83
100	G	0.8	50	5573	0.98	0.75	0.85
100	G	0.8	100	5573	0.98	0.79	0.87
100	G	0.8	200	5573	0.98	0.84	0.9
100	G	0.8	300	5573	0.98	0.84	0.9
100	G	0.8	400	5573	0.98	0.84	0.9
100	G	0.8	500	5573	0.98	0.84	0.9
100	G	0.8	600	5573	0.98	0.84	0.9
100	G	0.8	700	5573	0.98	0.84	0.9
100	G	0.85	10	5573	1	0.71	0.83
100	G	0.85	50	5573	0.9	0.75	0.82
100	G	0.85	100	5573	0.98	0.79	0.87
100	G	0.85	200	5573	0.98	0.83	0.9
100	G	0.85	300	5573	0.98	0.83	0.9
100	G	0.85	400	5573	0.98	0.83	0.9
100	G	0.85	500	5573	0.98	0.83	0.9
100	G	0.85	600	5573	0.98	0.83	0.9
100	G	0.85	700	5573	0.98	0.83	0.9
100	G	0.9	10	5573	1	0.7	0.82
100	G	0.9	50	5573	0.99	0.72	0.83
100	G	0.9	100	5573	0.99	0.78	0.87
100	G	0.9	200	5573	0.99	0.82	0.9
100	G	0.9	300	5573	0.99	0.81	0.89
100	G	0.9	400	5573	0.99	0.81	0.89
100	G	0.9	500	5573	0.99	0.81	0.89
100	G	0.9	600	5573	0.99	0.81	0.89
100	G	0.9	700	5573	0.99	0.81	0.89
100	G	0.95	10	5573	1	0.7	0.82

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
100	G	0.95	50	5573	0.99	0.72	0.83
100	G	0.95	100	5573	0.99	0.77	0.87
100	G	0.95	200	5573	0.99	0.81	0.89
100	G	0.95	300	5573	0.99	0.81	0.89
100	G	0.95	400	5573	0.99	0.81	0.89
100	G	0.95	500	5573	0.99	0.81	0.89
100	G	0.95	600	5573	0.99	0.81	0.89
100	G	0.95	700	5573	0.99	0.81	0.89
200	G	0.7	10		0.99	0.72	0.83
200	G	0.7	50		0.96	0.75	0.84
200	G	0.7	100		0.95	0.81	0.87
200	G	0.7	200		0.95	0.85	0.9
200	G	0.7	300		0.95	0.85	0.9
200	G	0.7	400		0.95	0.85	0.9
200	G	0.7	500		0.95	0.85	0.9
200	G	0.7	600		0.95	0.85	0.9
200	G	0.7	700		0.95	0.85	0.9
200	G	0.75	10		0.98	0.71	0.82
200	G	0.75	50		0.97	0.74	0.84
200	G	0.75	100		0.97	0.8	0.88
200	G	0.75	200		0.97	0.85	0.91
200	G	0.75	300		0.97	0.85	0.91
200	G	0.75	400		0.97	0.85	0.91
200	G	0.75	500		0.97	0.85	0.91
200	G	0.75	600		0.97	0.85	0.91
200	G	0.75	700		0.97	0.85	0.91
200	G	0.8	10		0.99	0.7	0.82
200	G	0.8	50		0.98	0.74	0.84
200	G	0.8	100		0.98	0.78	0.87
200	G	0.8	200		0.98	0.83	0.9
200	G	0.8	300		0.98	0.83	0.9
200	G	0.8	400		0.98	0.83	0.9
200	G	0.8	500		0.98	0.83	0.9
200	G	0.8	600		0.98	0.83	0.9
200	G	0.8	700		0.98	0.83	0.9
200	G	0.85	10		1	0.7	0.82
200	G	0.85	50		0.98	0.72	0.83
200	G	0.85	100		0.97	0.76	0.85
200	G	0.85	200		0.97	0.81	0.88
200	G	0.85	300		0.97	0.81	0.88
200	G	0.85	400		0.97	0.81	0.88
200	G	0.85	500		0.97	0.81	0.88
200	G	0.85	600		0.97	0.81	0.88

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
200	G	0.85	700		0.97	0.81	0.88
200	G	0.9	10		1	0.7	0.82
200	G	0.9	50		0.99	0.72	0.83
200	G	0.9	100		0.98	0.75	0.85
200	G	0.9	200		0.98	0.8	0.88
200	G	0.9	300		0.98	0.8	0.88
200	G	0.9	400		0.98	0.8	0.88
200	G	0.9	500		0.98	0.8	0.88
200	G	0.9	600		0.98	0.8	0.88
200	G	0.9	700		0.98	0.8	0.88
200	G	0.95	10		1	0.7	0.82
200	G	0.95	50		0.99	0.71	0.83
200	G	0.95	100		0.99	0.76	0.86
200	G	0.95	200		0.99	0.8	0.88
200	G	0.95	300		0.99	0.8	0.88
200	G	0.95	400		0.99	0.8	0.88
200	G	0.95	500		0.99	0.8	0.88
200	G	0.95	600		0.99	0.8	0.88
200	G	0.95	700		0.99	0.8	0.88
10	P	0.7	10	396	0.85	0.87	0.86
10	P	0.7	50	396	0.82	0.91	0.86
10	P	0.7	100	396	0.82	0.95	0.88
10	P	0.7	200	396	0.82	0.95	0.88
10	P	0.7	300	396	0.82	0.95	0.88
10	P	0.7	400	396	0.82	0.95	0.88
10	P	0.7	500	396	0.82	0.95	0.88
10	P	0.7	600	396	0.82	0.95	0.88
10	P	0.7	700	396	0.82	0.95	0.88
10	P	0.75	10	396	0.85	0.87	0.86
10	P	0.75	50	396	0.84	0.91	0.87
10	P	0.75	100	396	0.84	0.95	0.89
10	P	0.75	200	396	0.84	0.95	0.89
10	P	0.75	300	396	0.84	0.95	0.89
10	P	0.75	400	396	0.84	0.95	0.89
10	P	0.75	500	396	0.84	0.95	0.89
10	P	0.75	600	396	0.84	0.95	0.89
10	P	0.75	700	396	0.84	0.95	0.89
10	P	0.8	10	396	0.93	0.84	0.88
10	P	0.8	50	396	0.93	0.87	0.9
10	P	0.8	100	396	0.92	0.9	0.91
10	P	0.8	200	396	0.91	0.9	0.9
10	P	0.8	300	396	0.91	0.9	0.9

Continued on next page

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
10	P	0.8	400	396	0.91	0.9	0.9
10	P	0.8	500	396	0.91	0.9	0.9
10	P	0.8	600	396	0.91	0.9	0.9
10	P	0.8	700	396	0.91	0.9	0.9
10	P	0.85	10	396	0.95	0.84	0.89
10	P	0.85	50	396	0.94	0.87	0.9
10	P	0.85	100	396	0.93	0.9	0.91
10	P	0.85	200	396	0.93	0.9	0.91
10	P	0.85	300	396	0.93	0.9	0.91
10	P	0.85	400	396	0.93	0.9	0.91
10	P	0.85	500	396	0.93	0.9	0.91
10	P	0.85	600	396	0.93	0.9	0.91
10	P	0.85	700	396	0.93	0.9	0.91
10	P	0.9	10	396	1	0.84	0.91
10	P	0.9	50	396	0.98	0.87	0.92
10	P	0.9	100	396	0.95	0.9	0.92
10	P	0.9	200	396	0.94	0.9	0.92
10	P	0.9	300	396	0.94	0.9	0.92
10	P	0.9	400	396	0.94	0.9	0.92
10	P	0.9	500	396	0.94	0.9	0.92
10	P	0.9	600	396	0.94	0.9	0.92
10	P	0.9	700	396	0.94	0.9	0.92
10	P	0.95	10	396	1	0.84	0.91
10	P	0.95	50	396	0.98	0.86	0.92
10	P	0.95	100	396	0.98	0.9	0.94
10	P	0.95	200	396	0.97	0.9	0.93
10	P	0.95	300	396	0.97	0.9	0.93
10	P	0.95	400	396	0.97	0.9	0.93
10	P	0.95	500	396	0.97	0.9	0.93
10	P	0.95	600	396	0.97	0.9	0.93
10	P	0.95	700	396	0.97	0.9	0.93
50	P	0.7	10	1930	0.83	0.88	0.85
50	P	0.7	50	1930	0.8	0.91	0.85
50	P	0.7	100	1930	0.8	0.96	0.87
50	P	0.7	200	1930	0.8	0.96	0.87
50	P	0.7	300	1930	0.8	0.96	0.87
50	P	0.7	400	1930	0.8	0.96	0.87
50	P	0.7	500	1930	0.8	0.96	0.87
50	P	0.7	600	1930	0.8	0.96	0.87
50	P	0.7	700	1930	0.8	0.96	0.87
50	P	0.75	10	1930	0.83	0.88	0.85
50	P	0.75	50	1930	0.81	0.91	0.86
50	P	0.75	100	1930	0.81	0.96	0.88

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
50	P	0.75	200	1930	0.81	0.96	0.88
50	P	0.75	300	1930	0.81	0.96	0.88
50	P	0.75	400	1930	0.81	0.96	0.88
50	P	0.75	500	1930	0.81	0.96	0.88
50	P	0.75	600	1930	0.81	0.96	0.88
50	P	0.75	700	1930	0.81	0.96	0.88
50	P	0.8	10	1930	0.91	0.86	0.88
50	P	0.8	50	1930	0.9	0.89	0.89
50	P	0.8	100	1930	0.89	0.94	0.91
50	P	0.8	200	1930	0.89	0.94	0.91
50	P	0.8	300	1930	0.89	0.94	0.91
50	P	0.8	400	1930	0.89	0.94	0.91
50	P	0.8	500	1930	0.89	0.94	0.91
50	P	0.8	600	1930	0.89	0.94	0.91
50	P	0.8	700	1930	0.89	0.94	0.91
50	P	0.85	10	1930	0.91	0.84	0.87
50	P	0.85	50	1930	0.91	0.88	0.89
50	P	0.85	100	1930	0.89	0.94	0.91
50	P	0.85	200	1930	0.89	0.94	0.91
50	P	0.85	300	1930	0.89	0.94	0.91
50	P	0.85	400	1930	0.89	0.94	0.91
50	P	0.85	500	1930	0.89	0.94	0.91
50	P	0.85	600	1930	0.89	0.94	0.91
50	P	0.85	700	1930	0.89	0.94	0.91
50	P	0.9	10	1930	1	0.84	0.91
50	P	0.9	50	1930	0.97	0.88	0.92
50	P	0.9	100	1930	0.94	0.93	0.93
50	P	0.9	200	1930	0.94	0.93	0.93
50	P	0.9	300	1930	0.94	0.93	0.93
50	P	0.9	400	1930	0.94	0.93	0.93
50	P	0.9	500	1930	0.94	0.93	0.93
50	P	0.9	600	1930	0.94	0.93	0.93
50	P	0.9	700	1930	0.94	0.93	0.93
50	P	0.95	10	1930	1	0.84	0.91
50	P	0.95	50	1930	0.98	0.87	0.92
50	P	0.95	100	1930	0.97	0.92	0.94
50	P	0.95	200	1930	0.97	0.92	0.94
50	P	0.95	300	1930	0.97	0.92	0.94
50	P	0.95	400	1930	0.97	0.92	0.94
50	P	0.95	500	1930	0.97	0.92	0.94
50	P	0.95	600	1930	0.97	0.92	0.94
50	P	0.95	700	1930	0.97	0.92	0.94
100	P	0.7	10	3813	0.82	0.9	0.86

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
100	P	0.7	50	3813	0.81	0.94	0.87
100	P	0.7	100	3813	0.79	0.96	0.87
100	P	0.7	200	3813	0.79	0.96	0.87
100	P	0.7	300	3813	0.79	0.96	0.87
100	P	0.7	400	3813	0.79	0.96	0.87
100	P	0.7	500	3813	0.79	0.96	0.87
100	P	0.7	600	3813	0.79	0.96	0.87
100	P	0.7	700	3813	0.79	0.96	0.87
100	P	0.75	10	3813	0.83	0.9	0.86
100	P	0.75	50	3813	0.82	0.94	0.88
100	P	0.75	100	3813	0.8	0.96	0.87
100	P	0.75	200	3813	0.8	0.96	0.87
100	P	0.75	300	3813	0.8	0.96	0.87
100	P	0.75	400	3813	0.8	0.96	0.87
100	P	0.75	500	3813	0.8	0.96	0.87
100	P	0.75	600	3813	0.8	0.96	0.87
100	P	0.75	700	3813	0.8	0.96	0.87
100	P	0.8	10	3813	0.9	0.88	0.89
100	P	0.8	50	3813	0.9	0.91	0.9
100	P	0.8	100	3813	0.88	0.93	0.9
100	P	0.8	200	3813	0.88	0.93	0.9
100	P	0.8	300	3813	0.88	0.93	0.9
100	P	0.8	400	3813	0.88	0.93	0.9
100	P	0.8	500	3813	0.88	0.93	0.9
100	P	0.8	600	3813	0.88	0.93	0.9
100	P	0.8	700	3813	0.88	0.93	0.9
100	P	0.85	10	3813	0.91	0.86	0.88
100	P	0.85	50	3813	0.9	0.9	0.9
100	P	0.85	100	3813	0.88	0.93	0.9
100	P	0.85	200	3813	0.89	0.93	0.91
100	P	0.85	300	3813	0.89	0.93	0.91
100	P	0.85	400	3813	0.89	0.93	0.91
100	P	0.85	500	3813	0.89	0.93	0.91
100	P	0.85	600	3813	0.89	0.93	0.91
100	P	0.85	700	3813	0.89	0.93	0.91
100	P	0.9	10	3813	1	0.86	0.92
100	P	0.9	50	3813	0.96	0.9	0.93
100	P	0.9	100	3813	0.94	0.93	0.93
100	P	0.9	200	3813	0.94	0.93	0.93
100	P	0.9	300	3813	0.94	0.93	0.93
100	P	0.9	400	3813	0.94	0.93	0.93
100	P	0.9	500	3813	0.94	0.93	0.93
100	P	0.9	600	3813	0.94	0.93	0.93

Continued on next page

Appendix I

Full Evaluation Results of SimiMatch

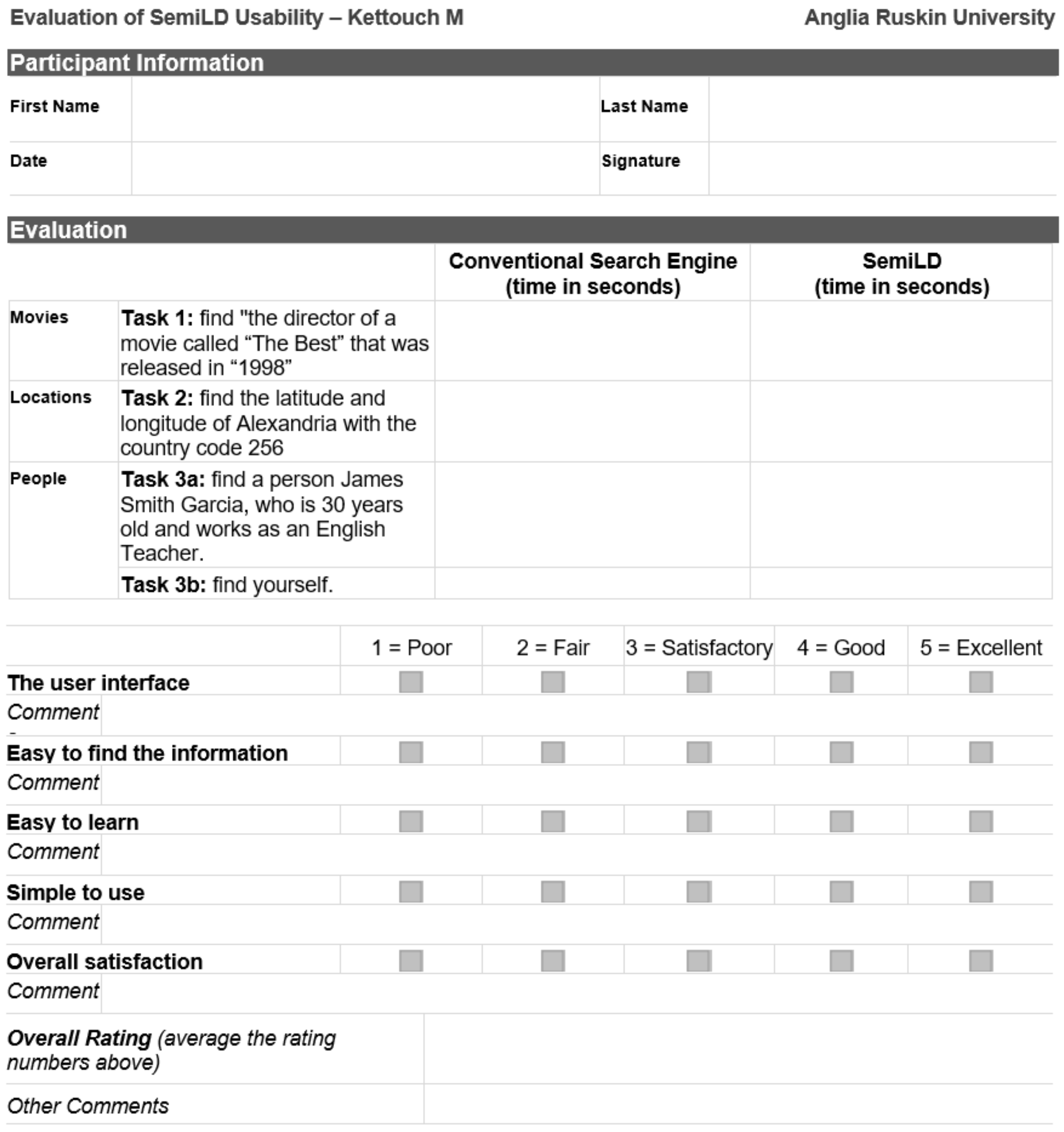
N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
100	P	0.9	700	3813	0.94	0.93	0.93
100	P	0.95	10	3813	1	0.86	0.92
100	P	0.95	50	3813	0.97	0.89	0.93
100	P	0.95	100	3813	0.96	0.93	0.94
100	P	0.95	200	3813	0.96	0.93	0.94
100	P	0.95	300	3813	0.96	0.93	0.94
100	P	0.95	400	3813	0.96	0.93	0.94
100	P	0.95	500	3813	0.96	0.93	0.94
100	P	0.95	600	3813	0.96	0.93	0.94
100	P	0.95	700	3813	0.96	0.93	0.94
200	P	0.7	10	7578	0.8	0.89	0.84
200	P	0.7	50	7578	0.8	0.92	0.86
200	P	0.7	100	7578	0.79	0.96	0.87
200	P	0.7	200	7578	0.79	0.96	0.87
200	P	0.7	300	7578	0.79	0.96	0.87
200	P	0.7	400	7578	0.79	0.96	0.87
200	P	0.7	500	7578	0.79	0.96	0.87
200	P	0.7	600	7578	0.79	0.96	0.87
200	P	0.7	700	7578	0.79	0.96	0.87
200	P	0.75	10	7578	0.81	0.89	0.85
200	P	0.75	50	7578	0.8	0.92	0.86
200	P	0.75	100	7578	0.8	0.96	0.87
200	P	0.75	200	7578	0.8	0.96	0.87
200	P	0.75	300	7578	0.8	0.96	0.87
200	P	0.75	400	7578	0.8	0.96	0.87
200	P	0.75	500	7578	0.8	0.96	0.87
200	P	0.75	600	7578	0.8	0.96	0.87
200	P	0.75	700	7578	0.8	0.96	0.87
200	P	0.8	10	7578	0.88	0.87	0.87
200	P	0.8	50	7578	0.88	0.9	0.89
200	P	0.8	100	7578	0.87	0.93	0.9
200	P	0.8	200	7578	0.87	0.93	0.9
200	P	0.8	300	7578	0.87	0.93	0.9
200	P	0.8	400	7578	0.87	0.93	0.9
200	P	0.8	500	7578	0.87	0.93	0.9
200	P	0.8	600	7578	0.87	0.93	0.9
200	P	0.8	700	7578	0.87	0.93	0.9
200	P	0.85	10	7578	0.88	0.87	0.87
200	P	0.85	50	7578	0.87	0.9	0.88
200	P	0.85	100	7578	0.87	0.93	0.9
200	P	0.85	200	7578	0.87	0.93	0.9
200	P	0.85	300	7578	0.87	0.93	0.9
200	P	0.85	400	7578	0.87	0.93	0.9

Continued on next page

Full Evaluation Results of SimiMatch

N	Domaine	Threshold	GS_version	N_properties	Precision	Recall	F1
200	P	0.85	500	7578	0.87	0.93	0.9
200	P	0.85	600	7578	0.87	0.93	0.9
200	P	0.85	700	7578	0.87	0.93	0.9
200	P	0.9	10	7578	0.99	0.87	0.93
200	P	0.9	50	7578	0.94	0.89	0.91
200	P	0.9	100	7578	0.92	0.93	0.92
200	P	0.9	200	7578	0.92	0.93	0.92
200	P	0.9	300	7578	0.92	0.93	0.92
200	P	0.9	400	7578	0.92	0.93	0.92
200	P	0.9	500	7578	0.92	0.93	0.92
200	P	0.9	600	7578	0.92	0.93	0.92
200	P	0.9	700	7578	0.92	0.93	0.92
200	P	0.95	10	7578	1	0.86	0.92
200	P	0.95	50	7578	0.97	0.89	0.93
200	P	0.95	100	7578	0.96	0.92	0.94
200	P	0.95	200	7578	0.96	0.92	0.94
200	P	0.95	300	7578	0.96	0.92	0.94
200	P	0.95	400	7578	0.96	0.92	0.94
200	P	0.95	500	7578	0.96	0.92	0.94
200	P	0.95	600	7578	0.96	0.92	0.94
200	P	0.95	700	7578	0.96	0.92	0.94

Appendix II - SemiLD’s Evaluation Sheet



Evaluation of SemiLD Usability – Kettouch M

Anglia Ruskin University

Participant Information

First Name		**Last Name**	
Date		**Signature**	

Evaluation

		Conventional Search Engine (time in seconds)	SemiLD (time in seconds)
Movies	**Task 1:** find "the director of a movie called “The Best” that was released in “1998”		
Locations	**Task 2:** find the latitude and longitude of Alexandria with the country code 256		
People	**Task 3a:** find a person James Smith Garcia, who is 30 years old and works as an English Teacher.		
	Task 3b: find yourself.		

	1 = Poor	2 = Fair	3 = Satisfactory	4 = Good	5 = Excellent
The user interface	□	□	□	□	□
Comment					
Easy to find the information	□	□	□	□	□
Comment					
Easy to learn	□	□	□	□	□
Comment					
Simple to use	□	□	□	□	□
Comment					
Overall satisfaction	□	□	□	□	□
Comment					
Overall Rating *(average the rating numbers above)*					
Other Comments					

SemiLD’s Evaluation Sheet

Appendix III - Implementations' Details

System Specification / Software / Datasets	Version / Download Link
Operating System	Windows 10 Education 64-bit
CPU	2.53 GHz i5 CPU
RAM	6.00 GB
Java	JDK 1.8.0
Apache Jena	2.7.4
RDF HDT	1.0
GSON (JSON library)	2.62
Tomcat Server	7.0
Dbpedia	http://gaia.infor.uva.es/hdt/DBPedia-3.9-en.hdt.gz
LinkedMDB	http://gaia.infor.uva.es/hdt/linkedmdb.hdt.gz
LinkedGeoData	http://gaia.infor.uva.es/hdt/linkedgeodata
Geonames	http://gaia.infor.uva.es/hdt/geonames-11-11-2012.hdt.gz

Implementations' details

The source code of the implementations presented in this thesis can be found in: https://github.com/medke